TECHNISCHE
UNIVERSITÄT
WIEN

VIENNA
UNIVERSITY OF
TECHNOLOGY

DIPLOMARBEIT

Spatiotemporal Analysis of Cultured Neural Networks

Ausgeführt am

Atominstitut der Österreichischen Universitäten

der Technischen Universität Wien

unter der Anleitung von

Prof. Harald Markum

Prof. Ruedi Stoop (ETH/Uni Zürich)

durch

Andreas Zöttl

Bräuhausgasse 42-44/9

1050 Wien

Wien, im Mai 2009

Abstract

Researchers at EMPA, St. Gallen, Switzerland cultivated an in vitro neural network on one side of a multielectrode array and grew myofibrils on the other side to measure the activity between them. The neural action potential signals from 60 spatially separated channels are analyzed by using correlation functions $C_{ij}(\tau)$ between channels i and j. Integrating the correlation functions up to a certain value results in the correlation matrix a_{ij} which is a measure for the strength of the correlations. Due to the asymmetry of a_{ij} an information flow can be extracted from the network which results in a transfer matrix A_{ij} defining a Markov process. A set of correlation measures is established by taking a_{ij} to the power of β. By introducing a corresponding partition function $Z(\beta)$ phase-transition-like behavior near a critical β_c is studied. At early days of the experiment the network of neurons is strongly synchronized in contrast to later times. The transitions are stronger for the asynchronous case where the peaks of the second derivative of the partition function are much higher than for the synchronous case.

Zusammenfassung

Forscher der EMPA in St. Gallen (Schweiz) kultivierten ein neuronales Netz in vitro auf einer Seite eines Multielectrode Arrays mit 60 Kanälen. Auf der anderen Seite wurden Myofibrillen angebracht, und anschließend wurde die Aktivität zwischen dem neuronalen Netz und den Myofibrillen gemessen. Die Aktionspotentiale der Neuronen im Netzwerk wurden mittels Korrelationsfunktionen $C_{ij}(\tau)$ zwischen Kanälen i und j analysiert. Integration der Korrelationsfunktionen bis zu einem bestimmten Wert liefert eine Korrelationsmatrix a_{ij}, die ein Maß für die Stärke der Korrelationen bedeutet. Die Asymmetrie von a_{ij} beschreibt einen Informationsfluss im Netzwerk, welcher durch eine Transfer-Matrix A_{ij}, die einen Markov-Prozess definiert, dargestellt wird. Mittels a_{ij}^{β} wird eine Schar von Korrelationsmaßen eingeführt. Die entsprechende Zustandssumme $Z(\beta)$ erlaubt die Untersuchung von phasenübergangähnlichem Verhalten. An frühen Tagen des Experiments ist das Netzwerk der Neuronen stark synchronisiert, im Gegensatz zu den späteren Tagen. Die Übergänge im asynchronen Fall, die durch große Amplituden der zweiten Ableitung der Zustandssumme charakterisiert sind, erweisen sich als stärker als im synchronisierten Fall.

Acknowledgements:
I want to thank Prof. Ruedi Stoop from the Institute of Neuroinformatics of the ETH/UNI Zürich for the mentoring and supervision during my diploma work in Zürich and for the enjoyable atmosphere in his group. Additionally I want to thank A. E. Oshiorenoya, S. Weigel and A. Bruinink from the EMPA in St. Gallen for providing interesting experimental data. Finally I want to thank Prof. Harald Markum from the Atomic Institute of the Austrian Universities for useful comments and remarks concerning the documentation of the thesis.

Contents

Chapter 1

Introduction

The interfacing of *in vitro neural networks* with *multielectrode array* (MEA) chips has been recently studied in various experiments [1–6]. MEA recordings help to understand the properties of neural networks, such as information processing, learning and plasticity. Understanding these properties will also lead to an improved development of neural prosthesis and brain-computer interfaces [7].

Neural prosthesis can substitute functions which would normally be provided by a human organ. The cochlear implant, for example, substitutes a damaged human cochlea and stimulates the auditory nerves directly. While the developmental progress of cochlear implants is quite advanced, the evolution of implants replacing motor activities is still in its incipiency. Applications of such prosthesis include implants for people with disabled motor activities such as epilepsy (see for example [8]) as well as neurally controlled limb prosthesis.

In order to achieve this goal it is useful to understand how the motor unit works within a MEA environment and how muscles respond to certain activities of networks of motor neurons. Different cultures of neural networks were analyzed with MEA experiments measuring different bursting regimes. A recent experiment at EMPA, St. Gallen, Switzerland [6], achieved to cultivate not only an ensemble of motor neurons on the MEA but additionally grew myofibrils (which are the components of the muscle fibers) on the other side of the chip. The activity not only within a cultured neural network but also activity between the neurons and the myofibrils was measured.

An advantage of recording cultured neural networks in vitro is the improved controlled environment in comparison to *in vivo* neural networks. Due to the use of drugs and chemical analysis it is thus possible to study different regimes of activity within neural networks and between them and associated destinations, such as myofibrils.

This thesis analyzes data from experiments described in [6]. Chap. 2 summarizes the experimental set-up and the results of these experiments. Example data is analyzed with the *correlation integral* and *recurrence plots* in Chap. 3 along with a tool to estimate the number of neuron spikes per burst. In Chap. 4 correlations between different MEA channels are investigated and asymmetric crosscorrelations are interpreted as an information flow in the network. In addition, a correlation weighting parameter β is introduced, which defines a partition function $Z(\beta)$, representing the behavior of the network at a certain β. The explicit form of $Z(\beta)$ is used to identify transitions in the network at a critical β_c. A transition denotes a substantially change in the network flow within a small variation of β.

Chapter 2

Neuromuscular Junction Experiments

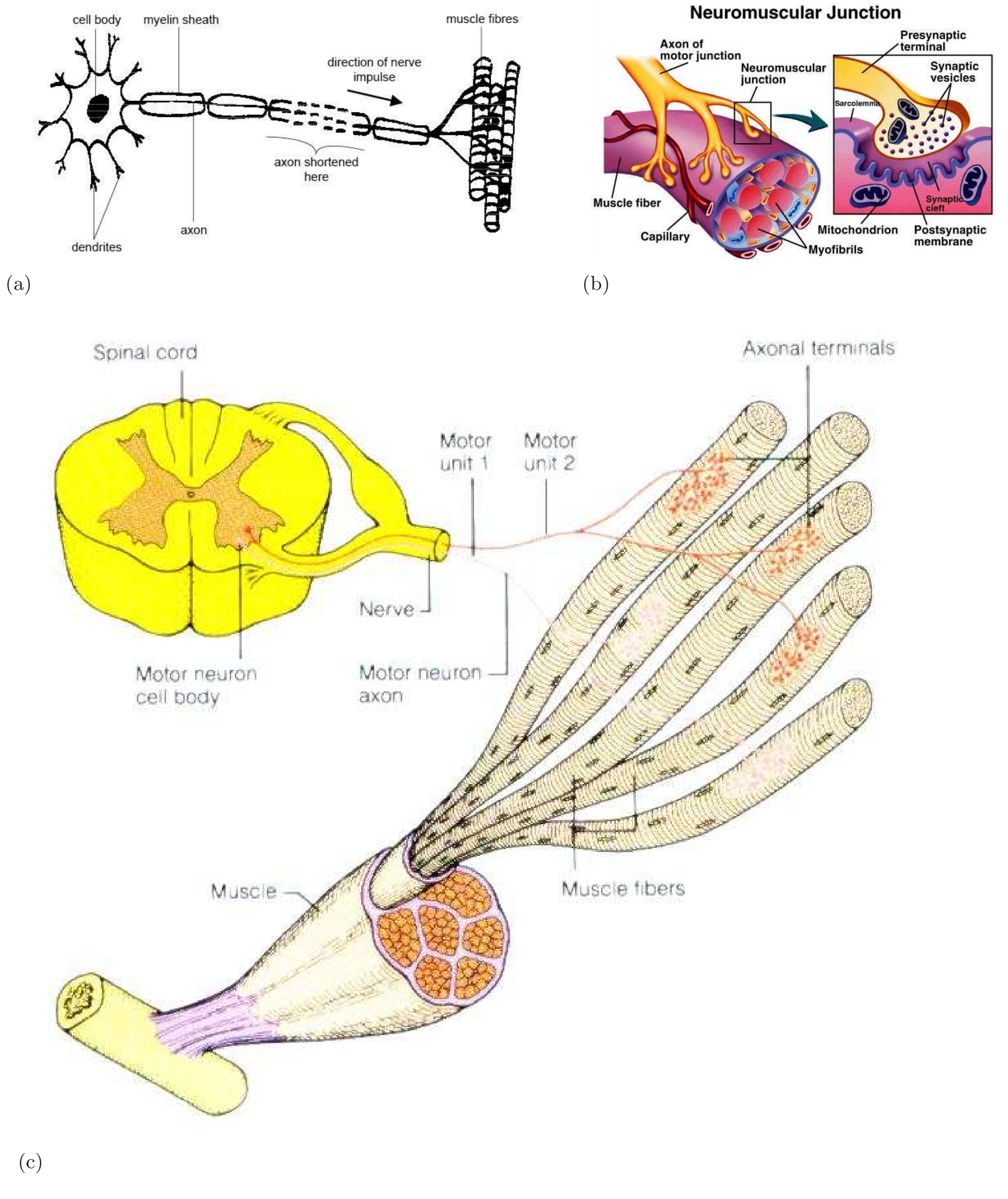
All experimental data presented in this thesis was taken from [6]. The goal of the experiments was to analyze an interface between a network of cultured neurons with a multielectrode array. In order to achieve this, A. E. Oshiorenoya et al. [6] cultivated a network of embryonic chicken motor neurons as well as embryonic chicken myofibrils on opposite sides of a multielectrode array chip in order to measure neural activity. This arrangement allowed an experimental simulation of the neuromuscular junction. One objective of the experiment was to clarify how much information is transmitted to axons depending on the activity of a neural network.

2.1 Neuromuscular Junction

Before the experiment is described in detail, the *motor unit* found in mammals is explained. A motor neuron delivers information via axons from the brain to the motor unit. A typical mammalian motor neuron (Fig. 2.1(a)) controls hundreds of *muscle fibers*. Together with this *muscle fiber block* the motor neuron builds up the motor unit. In total a typical muscle contains hundreds of thousands or even more than a million muscle fibers consisting of myofibrils. The motor neuron cell bodies are located in the spinal cord (Fig. 2.1(c)) or in the brain. Every motor neuron axon branches into hundreds of arms. The axon terminal (*presynaptic terminal*) couples to the postsynaptic membrane of the muscle fiber. Together they build up the *neuromuscular junction* or *motor end-plate* (Fig. 2.1(b)). Every muscle fiber is connected to one motor neuron only.

2.2 Experimental Setup and Results

A multielectrode array is an arrangement of typically 60 (or more) electrodes in which neural activities are recorded extracellularly by the electrodes. A typical topology of a multielectrode array is shown in Fig. 2.2. The electrodes are arranged in eight rows, where every line consists of eight electrodes, except the first and last row only consists of six electrodes. All channels are named after their row- and column-number in the form *channel ij*, where i is the column number and j is the row number. At the four edges of the grid no electrodes are arranged.

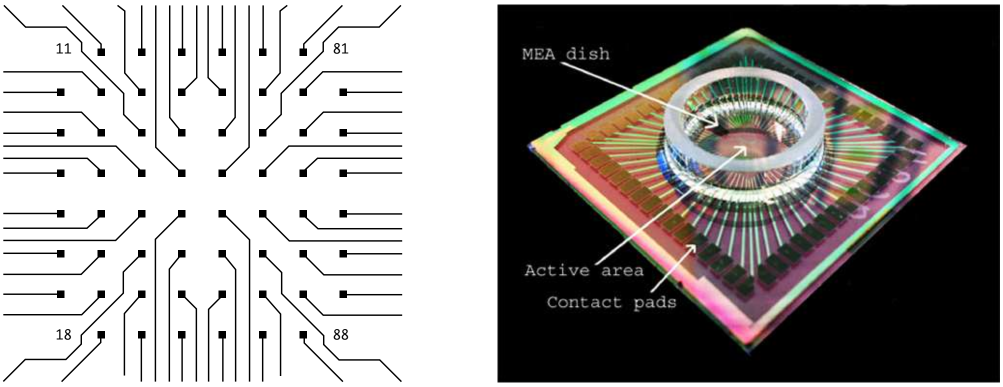


(a)

(b)

(c)

Figure 2.1: The mammalian motor unit; (a) typical motor neuron which delivers nerve impulses via the axons to the muscle fibers (taken from [9]); (b) neuromuscular junction between a motor neuron axon and a muscle fiber (taken from [10]); (c) a motor neuron located in the spinal cord is connected with a large number of muscle fibers, together they build up a motor unit (taken from [11]).



(a) (b)

Figure 2.2: (a) Topology of a typical 8 × 8 grid; (b) experimental set-up of a MEA. The electrode output runs via contact pads to a multichannel recording system (taken from [13]).

The channels 11, 81, 18 and 88 do not exist. An advantage of multielectrode arrays is the spatial information one can gain from simultaneous electrode recordings.

One side of the multielectrode array was prepared with a rat collagen matrix which provides a buffer between the chip and the myofibrils arranged on the matrix. On the other side of the MEA the cultured network of motor neurons was placed on an area prepared with laminin. The whole arrangement was surrounded in a chemical solution and axons grew from the cultured neural network through the grid to the myofibrils and built up an *in vitro* re-aggregated neuromuscular junction. Two different set-ups provided recording from the somas (cell bodies) only (Fig. 2.3 right), and from the dendrites only (Fig. 2.3 left). After a few *days in vitro* an interaction between the neural network (*information source*) and the myofibrils (*information destination*) via the neurites (*information channel*) was observed. The motor neurons were stimulated with chemicals which caused different regimes of neural network behavior and different responses to the myofibrils. The motor neurons show spontaneous synchronized network bursting activity [14], as well as asynchronous activity and regular/irregular spiking between synchronous and asynchronous bursting.

2.3 Data Used for Analysis

Data was recorded from different set-ups at various days in vitro. All signals from all channels were primarily recorded with a sample rate of 25 kHz. The signals are of the form of neural spike trains containing regular spike sequences, bursts and noise.

When a neural spike train is recorded, it can be useful to simplify the information of the neural action potential. One approach is to neglect the explicit form of the action potential and consider only the times *when* a neuron is firing. There are several methods to identify these spike times, such as *thresholding*, where the times are taken when a certain potential is reached. It is clear that one cannot avoid to identify noise as a spike in this approach. The

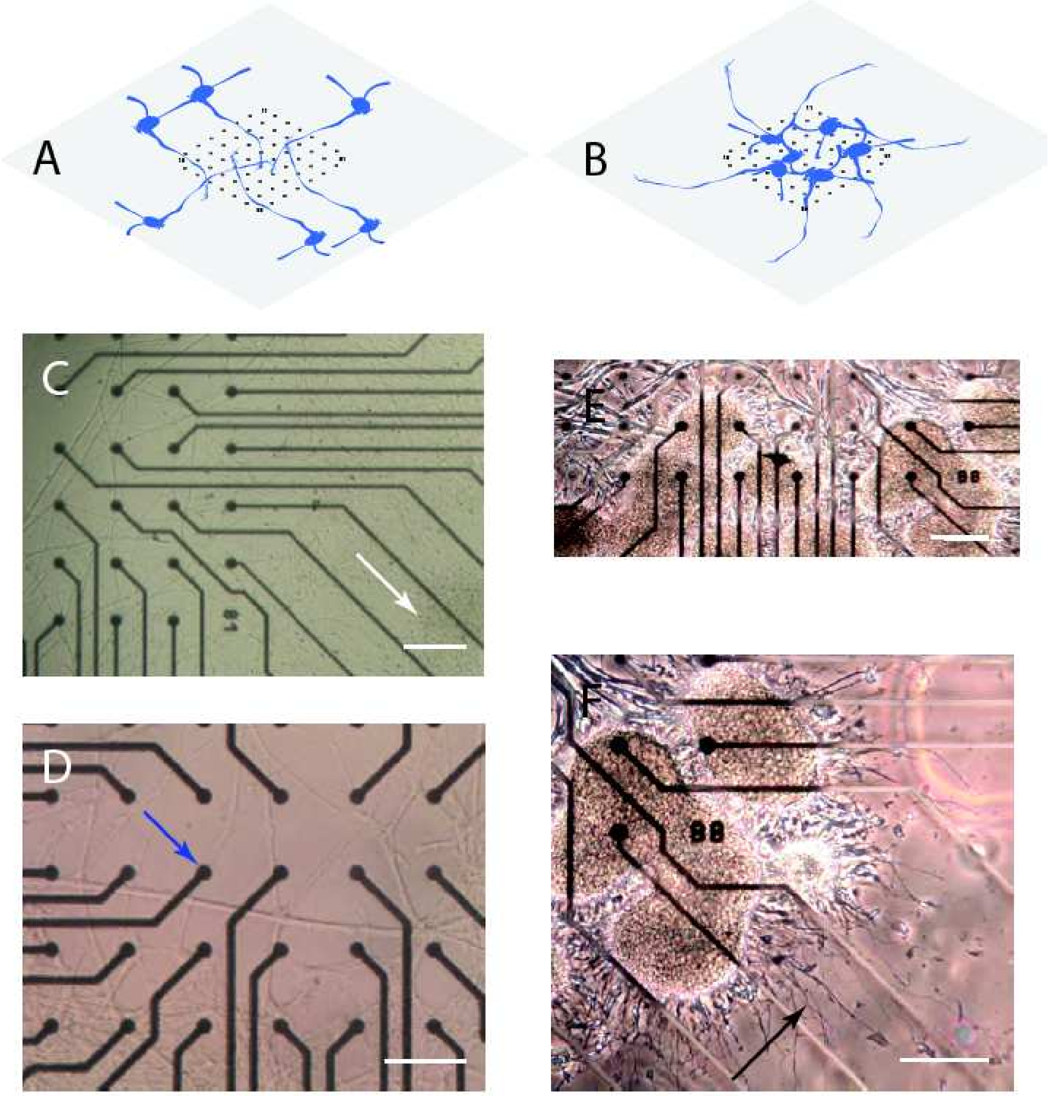


Figure 2.3: (A) schematic arrangement for neurite recording; (B) schematic arrangement for soma recording; (C)-(D) micrograph of chip with axonal network for the case shown in (A); (E)-(F) micrograph of chip with soma re-aggregate for the case shown in (B); (all pictures taken from [6]).

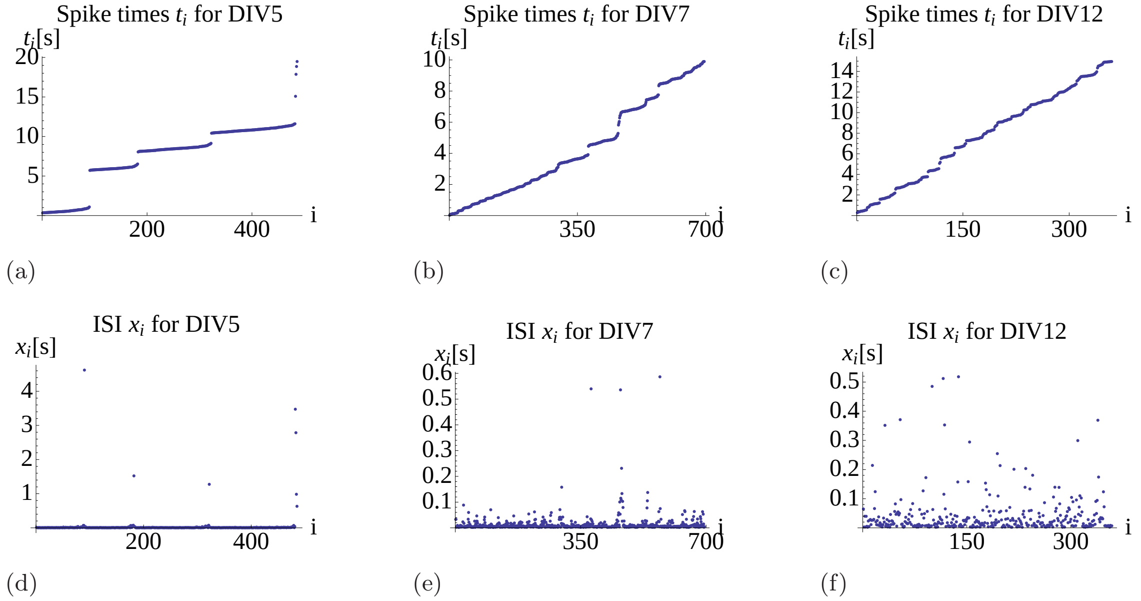
signal one obtains from this simplification is of the form

$$\{t_i\} = \{t_1, t_2, t_3, \dots, t_K\}, \quad t_1 < t_2 < \cdots < t_K, \tag{2.1}$$

where t_i are the times a spike is detected and K is the total number of spikes recorded in a certain time interval T. It is often useful to measure the time *between* to spikes in addition to *when* a certain spike occurs. A sequence of *interspike intervals* $\{x_i\}$ can be constructed,

$$\{x_i\} = \{t_2 - t_1, t_3 - t_2, t_4 - t_3, \dots, t_K - t_{K-1}\}, \quad i = 1, \dots, K-1, \tag{2.2}$$

where $L = K - 1$ is the number of interspike intervals. Due to the fact that a neuron stays in a refractory period for a certain time, there is a minimum value $x_{\min} \approx 2$ ms an interspike interval can take on.



(a) (b) (c)

(d) (e) (f)

Figure 2.4: Spike times $\{t_i\}$ for channel 12 recorded at DIV5 (a), DIV7 (b) and DIV12 (c) and corresponding interspike intervals $\{x_i\}$ for DIV5 (d), DIV7 (e) and DIV12 (f).

For the analysis data is used from three different days in vitro: (a) data from a relatively early stage of the experiment, i.e. from day in vitro 5, (b) data from day in vitro 7 and (c) data from a late stage of the experiment (day in vitro 12). In the following *day in vitro* is abbreviated by *DIV*. Fig. 2.4 shows the spike times $\{t_i\}$ and the interspike intervals (ISI) $\{x_i\}$ for channel 12 for MEA recording examples from DIV5-12. For DIV5 long burst sequences are interrupted by long interburst intervals which are of the order of some seconds. This changes at DIV7 where the burst sequences are not so clear any more and the interburst intervals are smaller (parts of a second maximally), for DIV12 the signal is characterized by short bursts and noise.

Data from all channels was recorded simultaneously. This is important when comparing data from different channels at the same DIV. Chap. 3 will concentrate on the analysis of single channels only while the analysis of the interaction between different channels is discussed in Chap. 4.

Chapter 3

Data Analysis of Single Signals

The analysis of the neural spike times can be accomplished at different qualitative levels, where the general behavior of the entire network is of main interest. Due to recorded signals being effected by noise, the analysis of neural spike times is performed by statistical methods. Useful statistical tools from the field of nonlinear data analysis can be found in [15] or [16]. In the following two methods, namely the *correlation integral* and the *recurrence plot* are used. Finally a method to estimate the average length of bursts in neural spike trains is developed.

3.1 Correlation Integral

The correlation integral was first introduced by P. Grassberger and I. Procaccia to determine the strangeness of attractors in dissipative dynamical systems [17]. It is also used for noise-cleaning of time series [18], and it is a useful tool for finding temporal patterns in neural spike trains [19, 20]. First the usage of the correlation integral is explained and then it is applied to artificial and experimental data.

The correlation integral is built by using the interspike intervals $\{x_i\}$ (Eq. 2.2). From this sequence *embedded points* $\xi_k^{(m)} = \{x_k, x_{k+1}, \ldots, x_{k+(m-1)}\}$ are constructed, where m is the *embedding dimension*. The total number of embedded points is $N = L - m + 1$. This embedding procedure is called *coordinate delay construction*, and can be used to reconstruct a higher dimensional object from a scalar time series measurement.

The basic idea of embedding comes from H. Whitney who showed for a generic connected smooth d-dimensional manifold that it can be smoothly embedded in a 2d-dimensional Euclidean space [21]. F. Takens showed that it is generally possible to reconstruct the state space from scalar time series measurements $\{t_i\}$ [22]. A sufficient condition for a successful embedding is $m > 2D_C$, where D_C is the *correlation dimension* of the phase space attractor (see Eq. 3.4). T. Sauer showed that reconstruction is possible for a sequence of interspike intervals $\{x_i\}$ by using an *integrate and fire model* [23].

Using the embedded points $\xi_k^{(m)}$ the correlation integral for a specific m is now defined as

$$C_N^{(m)}(\epsilon) = \frac{1}{N(N-1)} \sum_{i \neq j} \theta\left(\epsilon - \left|\left|\xi_i^{(m)} - \xi_j^{(m)}\right|\right|\right), \tag{3.1}$$

where $\theta(x)$ is the Heaviside step function. It is defined as

$$\theta(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0. \end{cases} \tag{3.2}$$

The derivative of the step function is the Dirac δ-function, $\theta(x)' = \delta(x)$. It is problematic to deal with such infinities numerically. For $\sigma \ll 1$ the following function $\theta_\sigma(x)$ gives a good approximation for $\theta(x)$,

$$\theta_\sigma(x) = \frac{1}{1 + e^{-x/\sigma}}, \tag{3.3}$$

where the limit $\lim\limits_{\sigma \to 0} \theta_\sigma(x) \to \theta(x)$ exists. The sum in Eq. 3.1 runs over all i and j, except for the cases where $i = j$. The distance $d_{ij,p}^{(m)} = ||\xi_i^{(m)} - \xi_j^{(m)}||_p$ is the p-norm between two embedded points $\xi_i^{(m)}$ and $\xi_j^{(m)}$. Due to the fact that all norms are equivalent in $\mathbb{R}^n$, one can use any norm. To simplify calculations the maximum norm instead of the common Euclidean norm is used. From now on the index p is skipped in all expressions.

Consider a fixed $\epsilon > 0$. If the distance between points $d_{ij}^{(m)}$ is larger than ϵ, the Heaviside function yields a one, if it is smaller than ϵ, it gives a zero. After summation over all possible pairs $(i \neq j)$ and division by $N(N-1)$, one obtains Eq. 3.1. If all $d_{ij}^{(m)}$ are smaller than ϵ, $C_N^{(m)}$ is one. This is the case when ϵ is sufficiently large, i.e. $\epsilon > \max\limits_{i,j=1\ldots N} d_{ij}^{(m)}$. If all $d_{ij}^{(m)}$ are larger than ϵ, $C_N^{(m)}$ is zero, which holds for $\epsilon < \min\limits_{i,j=1\ldots N} d_{ij}^{(m)}$. With $d_{ij}^{(m)}$ normalized between zero and one, $C_N^{(m)}(\epsilon)$ assents from zero to one in the range $0 < \epsilon < 1$. It is useful to plot both the correlation function and ϵ on a logarithmic scale and the resulting $\log C(\log \epsilon)$ diagrams are called *log-log-correlation plots*. A detailed description can be found in [19] or [20]. The natural logarithm is used being defined via $\exp[\log x] = x$ for an arbitrary x. The correlation dimension can now be defined as

$$D_C^{(m)} = \lim_{\epsilon \to 0} \frac{\log C_N^{(m)}(\epsilon)}{\log \epsilon}. \tag{3.4}$$

Before applying the method of log-log-correlation plots onto experimental data, some examples show how the correlation integral can be used in noisy systems.

Example 1: In the first example the effect of the embedding dimension on the correlation integral is considered. It is possible to determine the length of a periodic pattern in a neural spike train. The train of interspike intervals is given by

$$\{x_i\} = \{14, 9, 22, 5, 20, 2, 14, 9, 22, 5, 20, 2, \ldots\} \tag{3.5}$$

in our example. The pattern $\{x_i^0\} = \{14, 9, 22, 5, 20, 2\}$ with the length $n = 6$ repeats itself, so one can write $\{x_i\} = \{x_i^0, x_i^0, x_i^0, \ldots\}$. Fig. 3.1 shows log-log-correlation plots for different embedding dimensions m. The larger the embedding dimension, the smaller the number of steps. M. Christen et al. [20] showed how the number of steps $s(m, n)$ for a certain pattern length n and a given embedding dimension m can be calculated, i.e. for $n = 6$ $s(1, 6) = 15$, $s(2, 6) = 12$, $s(3, 6) = 9$, $s(4, 6) = 7$, $s(5, 6) = 5$ and $s(6, 6) = 3$.

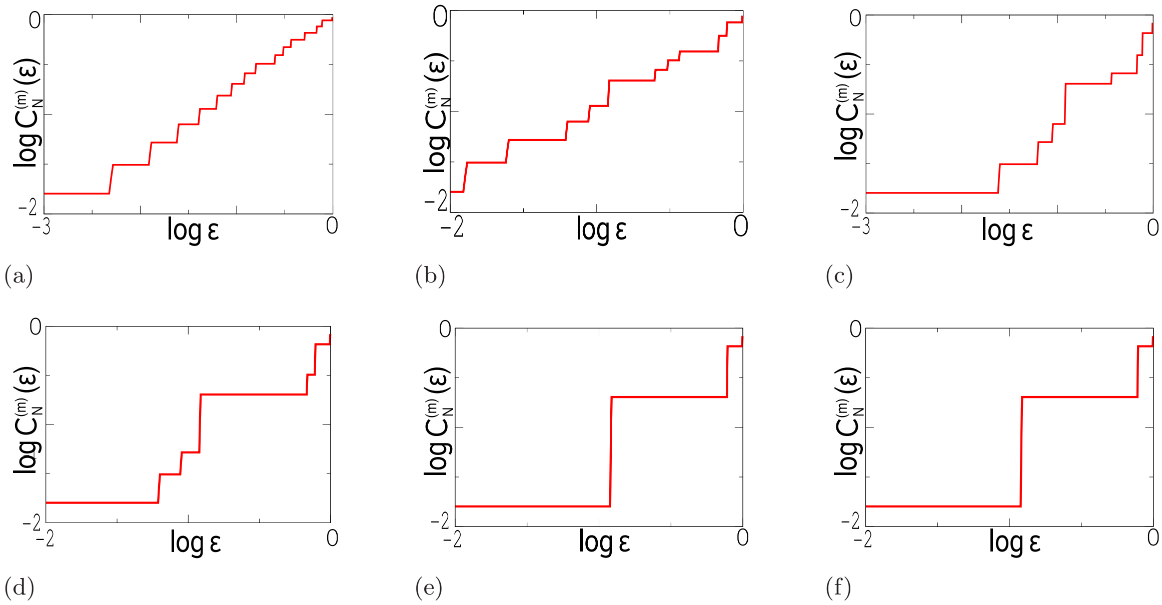


Figure 3.1: Log-log correlation plots for the sequence $\{x_i\} = \{14, 9, 22, 5, 20, 2, 14, 9, \dots\}$ with (a) m=1, (b) m=2, (c) m=3, (d) m=4, (e) m=6, (f) m=7.

For $m \geq 6$ the correlation integral does not change and the number of steps remains at $s(m, 6) = 3$. Generally one obtains $s(m, n) = \text{const}$ for a fixed n when $m \geq n$. By counting the steps for different m, one can determine the pattern length from a given periodic sequence, such as the sequence in Eq. 3.5.

Example 2: *Jitter* is added to the sequence from Eq. 3.5. Jitter implies that $\{x_i^0\}$ is not replicated perfectly, every interspike interval has a noise term δ_i,

$$\{x_i\} = \{14 + \delta_1, 9 + \delta_2, 22 + \delta_3, 5 + \delta_4, 20 + \delta_5, 2 + \delta_6, 14 + \delta_7, 9 + \delta_8, \dots\}, \qquad (3.6)$$

in this example it is white noise taken from the interval $(-Y, Y)$, where $Y > 0$ defines the strength of the added jitter. The *relative jitter* is defined as $\pm\frac{Y}{\min_i x_i^0}$. Fig. 3.2(left) shows the interspike interval sequences for different jitter strengths. For a relative jitter of 100% (Fig. 3.2(c)) and larger it is impossible to detect the correct pattern length $n = 6$ by only looking at the interspike intervals. Fig. 3.2(right) shows the corresponding log-log-correlation plots, where plots for different $m = 1, \dots, 8$ are combined in every figure. The steps get *smoothed* due to jitter, the steps in the correlation function are less pronounced for large jitter. Even when the relative jitter is high (100%, Fig. 3.2(c)) one can count the number of steps at $m \geq 6$. Only when the jitter is very large (500%, Fig. 3.2(d)) it is impossible to detect the pattern length.

Example 3: Consider the regular sequence $\{x_i^0\} = \{2, 25, 10\}$ in a noisy background. This example was first published in [19]. Between two sequences one can have a noisy signal with some probability. Fig. 3.3(left) shows the interspike interval sequences for added background noise (0%, 25%, 75%). In Fig. 3.3(a) (0% noise) the log-log-correlation

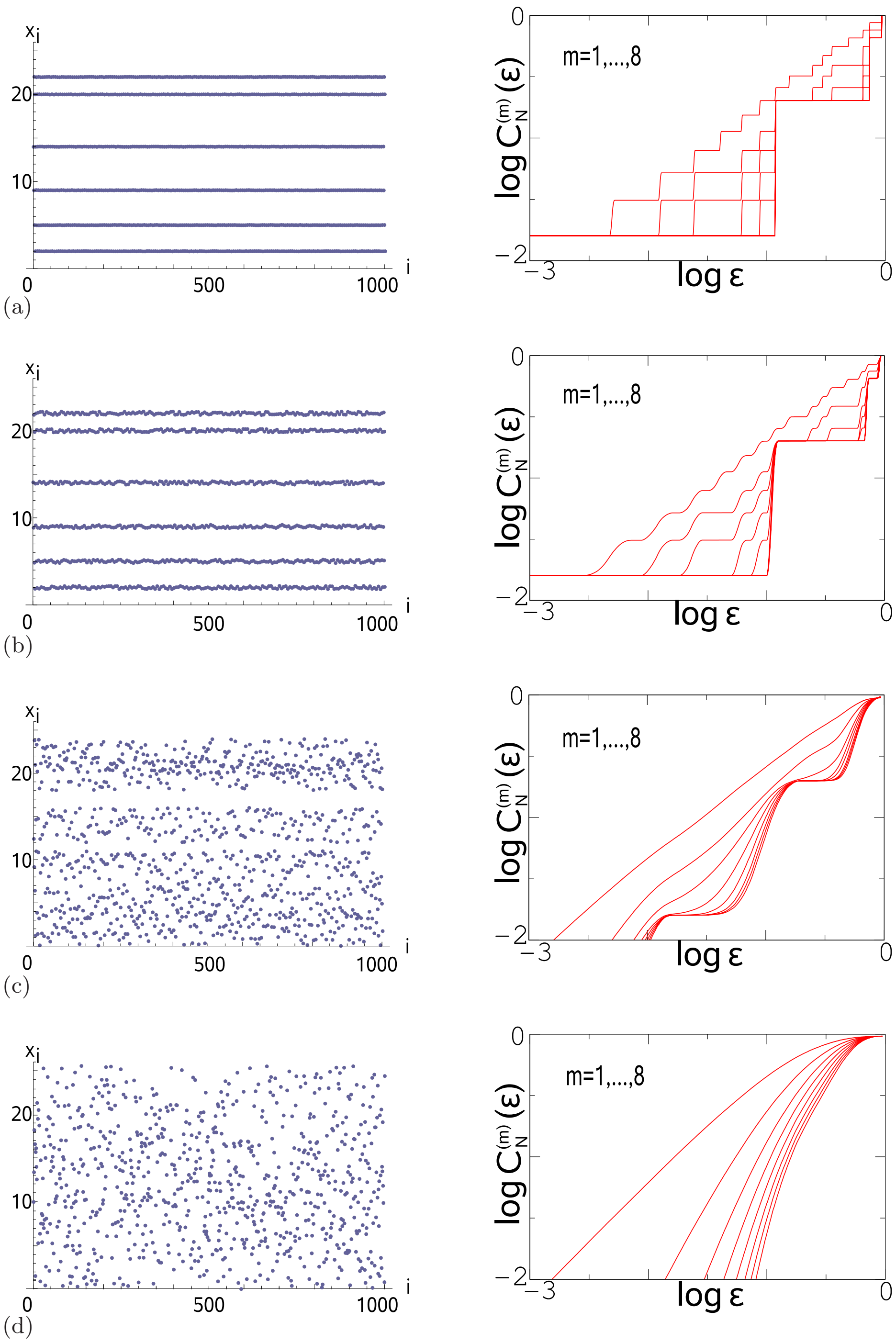


Figure 3.2: Interspike intervals and log-log-correlation plots for the sequence $\{x_i\} = \{14 + \delta_1, 9 + \delta_2, 22 + \delta_3, 5 + \delta_4, 20 + \delta_5, 2 + \delta_6, \dots\}$; relative jitter is ±0% in (a), ±10% in (b), ±100% in (c) and ±500% in (d).

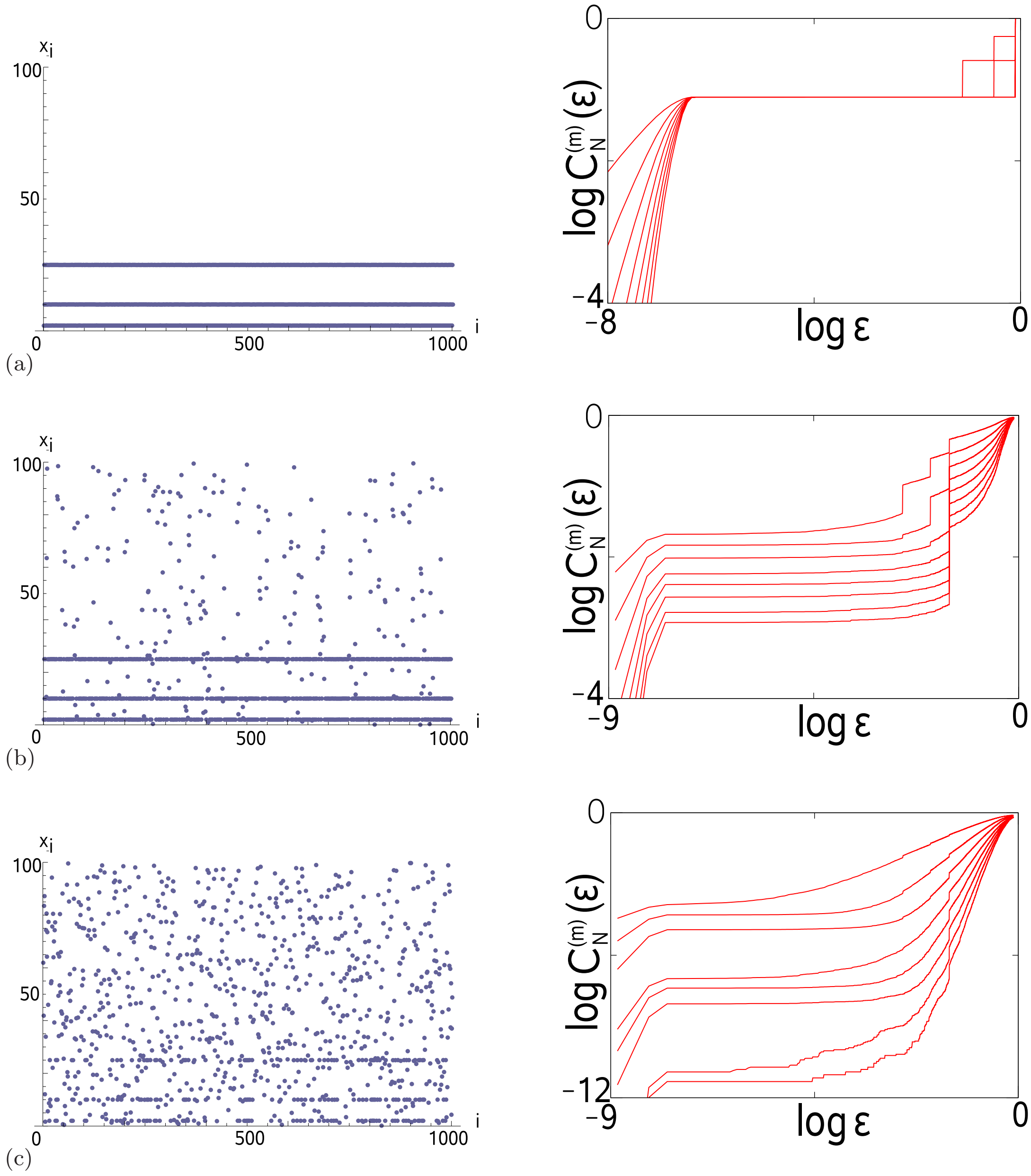
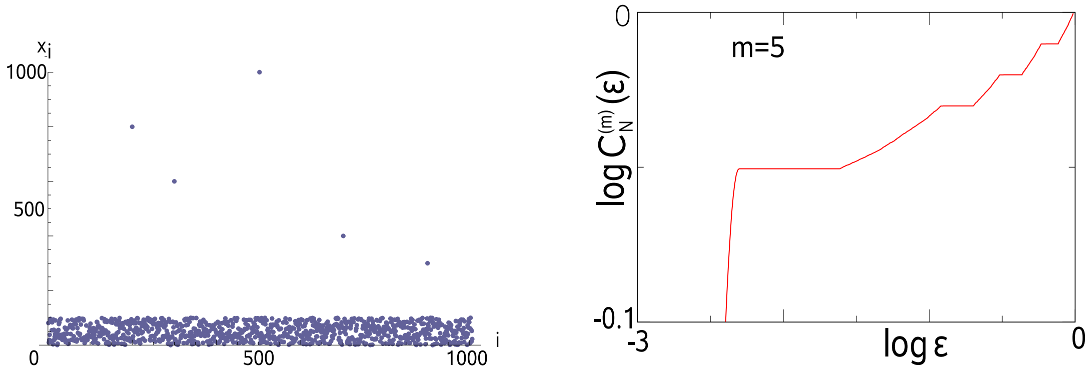


Figure 3.3: Sequence $\{2, 25, 10\}$ in noisy background; log-log-correlation plots and interspike intervals, where noise is 0% in (a), 25% in (b) and 75% in (c).



(a)

(b)

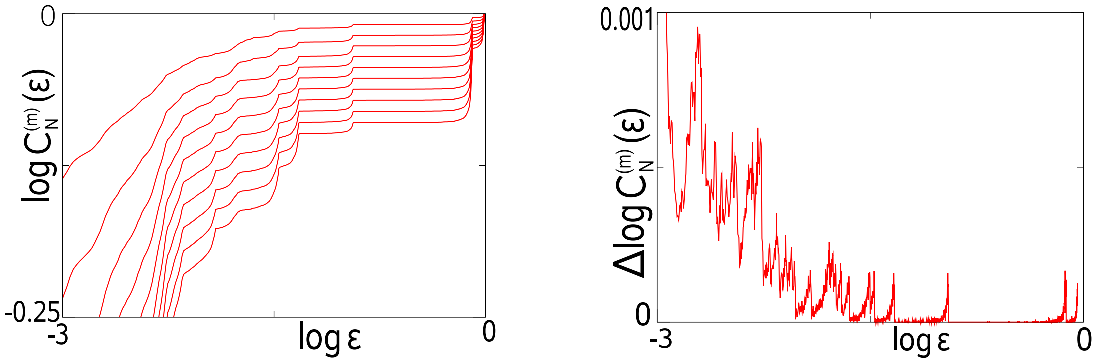
Figure 3.4: White noise with outliers; (a) interspike interval sequence; (b) log-log correlation plot for $m = 5$.

plots for $m = 1, \ldots, 8$ are shown. For $m \geq 3$ only one step can be seen. With added background noise (25% noise in Fig. 3.3(b) and 75% noise in Fig. 3.3(c)) it is not easy to detect the lowest embedding dimension m where the number of steps is minimal, although the log-log-correlation plots for different m are *grouped*. Due to the grouping the pattern length is three in this case.

Example 4: Sometimes the log-log-correlation plot shows steps even when no periodic pattern occurs. This is the case when outliers are embedded in a noisy sequence (Fig. 3.4(a)). For a signal containing only white noise one would not see any steps in the plot. Adding only a few (five in this example) outliers to a large number of interspike intervals imposes a structure on the log-log-correlation plot (Fig. 3.4(b)).

The examples above can be used as a guide to analyze the neural spike train data. For this purpose the interspike intervals recorded from channel 12 at DIV7 (see Fig. 2.4(b) and Fig. 2.4(e)) are considered. In Fig. 3.5(a) the correlation integrals are plotted for $m = 1, \ldots, 10$. Several steps in the plot can be seen but they do not reach a minimum at a certain m_{max}. Instead the number of steps remains the same for all m. Due to the fact that in a higher embedding dimension noisy effects get minimized, the steps become more and more visible with increasing m. The large interspike intervals (outliers) are responsible for the steps and it is not possible to detect a periodic pattern with the correlation integral method. Even with $m > 10$ (not shown in Fig. 3.5(a)) no periodicity can be seen. Fig. 3.5(b) shows the derivative $\Delta \log C_N^1(\epsilon)$ of the correlation function for $m = 1$. Every peak corresponds to a step in Fig. 3.5(a).

The correlation-integral method is useful for finding periodic patterns in a noisy background or within jittered signals. This was shown in the examples no. 1, 2 and 3 (additional examples can be found in [19] and [20]). However the occurrence of outliers destroys the structure which one would have seen without them. Analyzing data which contains short intra-burst intervals and large inter-burst intervals (which is a common case), one cannot find periodic patterns by counting the number of steps in the log-log-correlation plots. The distances $d_{ij}^{(m)}$ between a large-valued embedded point $\xi_i^{(m)}$ (corresponding to a large inter-burst interval)



(a)

(b)

Figure 3.5: (a) log-log-correlation plots for DIV7, channel 12 for $m = 1, \dots, 10$; (b) derivative $\Delta \log_2 C_N^{(1)}(\epsilon)$ for $m = 1$ shows peaks belonging to the steps in (a).

and small-valued points $\xi_j^{(m)}$ (corresponding to small intra-burst intervals) will be almost the same for all $j \in \{\text{intra-burst states}\}$. So the distances appear in groups around the lengths of the inter-burst intervals and produce steps in the log-log-correlation plots. Thus, if there is a periodic pattern within a burst one would not be able to see it. It would become invisible due to the large interspike intervals.

Thus, if no outliers occur, the correlation integral is a good method to find periodic patterns. If the noise contains outliers, the method becomes less useful.

3.2 Recurrence Plots

Another method for finding patterns in neural spike trains is that of *recurrence plots*, first introduced by J.-P. Eckmann in 1987 [24] to study the recurrence of trajectories $\{\vec{x}_i\}$ in phase space. Marwan et al. applied this method not only to nonlinear dynamical systems but also to problems from neuroscience, the financial market or geophysics [25]. The original concept goes back to Henri Poincaré who introduced the concept of recurrences in conservative systems already in his famous work *Sur le probleme des trois corps et les équations de la dynamique* in 1890 [26].

A recurrence plot is the visualization of the *recurrence matrix*, defined as

$$R_{ij} = \begin{cases} 1, & |\vec{x}_i - \vec{x}_j| < \epsilon \\ 0, & \text{else} \end{cases} \tag{3.7}$$

or,

$$R_{ij}(\epsilon) = \theta(\epsilon - ||\vec{x}_i - \vec{x}_j||), \tag{3.8}$$

where $\theta(\cdot)$ is the Heaviside step function and $\{\vec{x}_i\}$ is the phase space trajectory. To simplify calculations the maximum norm $||\cdot||_\infty$ is often used. A continuous signal $\vec{x}(t)$ is transferred to a discrete signal, where t takes on only the discrete values $t = i\Delta t$, $i = 1, \dots, N$, so $\vec{x}(t) \to \vec{x}_i$.

The recurrence plot is symmetric by definition,

$$R_{ij} = R_{ji}, \qquad R_{ii} = 1 \quad \text{(line of identity)}. \tag{3.9}$$

The determination of an appropriate ϵ is a nontrivial task. If ϵ is too small, the density of points in the plot is too low, i.e.

$$\frac{1}{n^2}\sum_{i,j=1}^{n} R_{ij} \ll 1 \tag{3.10}$$

and one cannot see the recurrence structure if there is one. When ϵ is too large, a lot of points are neighbors to almost all other points which cause artificial recurrences.

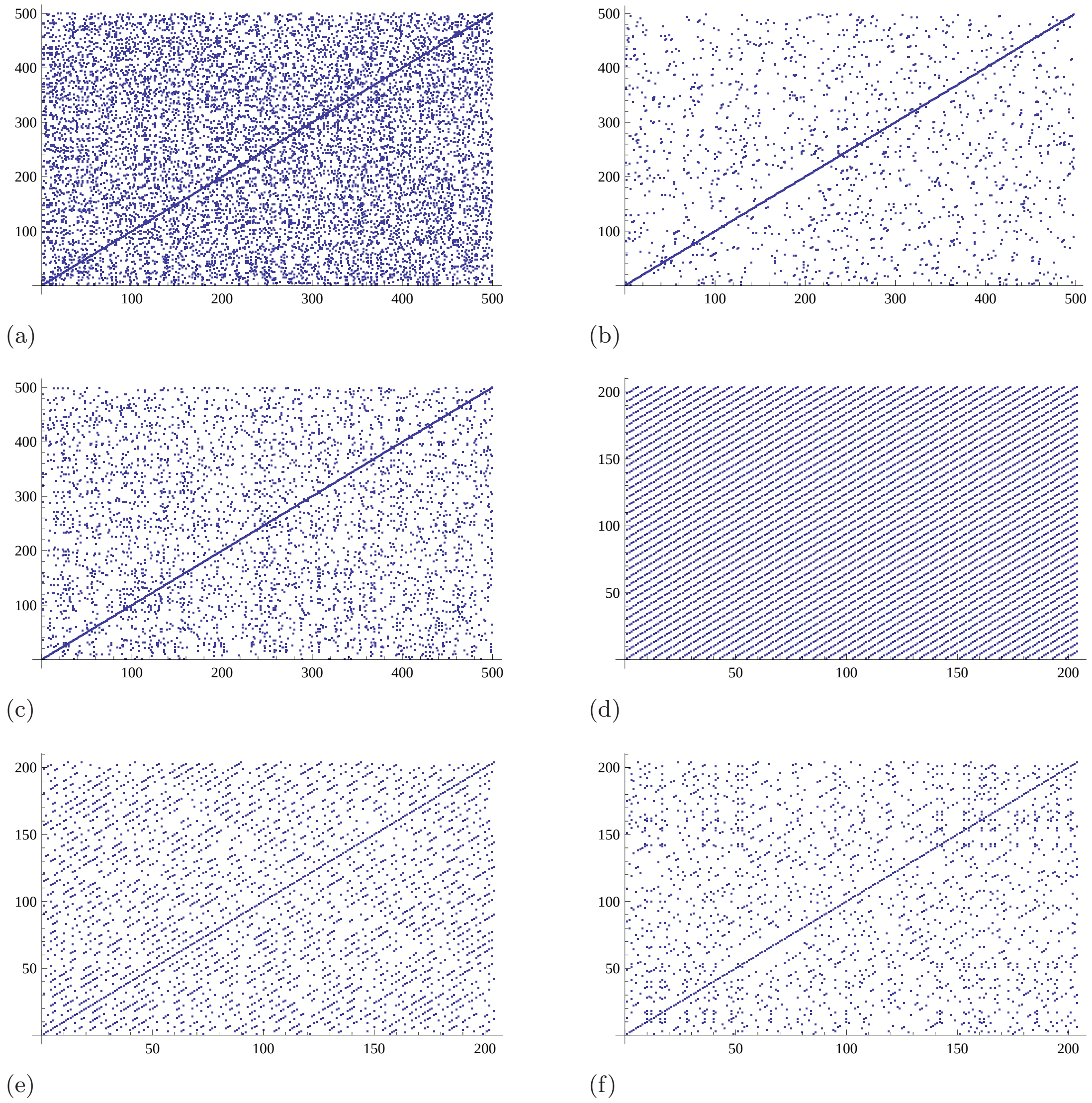
Additionally it would also be possible to calculate *cross-recurrence matrices* $R_{ij}^{\vec{x}\vec{y}}$ between two different phase space trajectories $\{\vec{x}_i\}$ and $\{\vec{y}_i\}$, i.e.

$$R_{ij}^{\vec{x}\vec{y}}(\epsilon) = \theta(\epsilon - ||\vec{x}_i - \vec{y}_j||). \tag{3.11}$$

In Fig. 3.6(a,b) the recurrence plots for a white noise signal $\{\vec{x}_i\}$, $i = 1, \ldots, 500$ are shown. The noisy signal, embedded in an m-dimensional phase space, causes a noisy recurrence as seen in Fig. 3.6(b) for $m = 3$. Depending on m and ϵ the quantitative amount of points changes but the qualitative picture remains the same. In Fig. 3.6(c) the signal follows a noisy log-normal distribution $p(x) \sim \exp[-a(\log x)^2]$, $a > 0$, and the qualitative picture does not change compared to the white noise signals. Fig. 3.6(d) shows the recurrence plot for the sequence shown in Eq. 3.5, where $i = 1, \ldots, 200$. Due to the fact that every value recurs after six steps, one can see diagonal lines, spaced by six units. It is a general feature of recurrence plots that periodicities are characterized by diagonal lines. In Fig. 3.6(e,f) jitter is added to the sequence (10% and 100%). For the 10% jitter case the diagonal lines are smeared and some additional points show up between them. If ϵ would be decreased, some of the fluctuations could be partly removed but on the other hand the diagonal lines would become sparser and the periodicity could not be seen so well any more. If ϵ would be increased, more and more fluctuations would occur. So it is indeed an important task to determine a proper ϵ. For 100% jitter it is still possible to see (or at least to *guess*) the periodicity.

From a sequence of interspike intervals $\{x_i\}$ embedded in an m-dimensional space the trajectory $\{\xi_i\}$ was constructed from the recorded data. Fig. 3.7 shows recurrence plots for three different series of data. In Fig. 3.7(a,b) the interspike interval sequence $\{x_i\}$ of channel no. 14 from DIV5 was embedded in dimension $m = 7$ and $m = 14$. To plot a reasonable amount of points, ϵ was chosen as $\epsilon = 5$ ms and $\epsilon = 10$ ms. Due to the fact that the distances are larger in higher dimensions, it is opportune to increase ϵ with increasing m. Because of the large burst lengths an embedding in a high dimension shows clear structures, where the colored boxes on the diagonal represent the bursts. For Fig. 3.7(c,d) data from channel no. 14 from DIV7 was analyzed. For a larger embedding dimension (Fig. 3.7(d)) the bursts in the recurrence plot occur more regularly than in the low-dimensional case (Fig. 3.7(c)). At DIV12 (Fig. 3.7(e)) the typical burst lengths are relatively small. When the embedding dimension is too high, it is almost impossible to see the bursts, but it is still possible to see a regular pattern (Fig. 3.7(f)).

In summery the recurrence plot is a tool to visualize the behavior of a phase space trajectory. It is an alternative for detecting periodicities in the trajectory by finding diagonals and measuring their distance. It does have however two major disadvantages compared to the correlation-integral method: First one has to find a proper ϵ which is a nontrivial task, as mentioned in [25]. While ϵ is a simple variable, $0 < \epsilon < 1$, in the correlation integral method, ϵ has to be reasonably selected in order to generate reliable recurrence plots. Secondly the



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.6: Different examples of recurrence plots; (a) white noise $\in (0, 1)$ with $m = 1$, $\epsilon = 0.02$ and (b) $m = 3$, $\epsilon = 0.1$; (c) randomly distributed log-normal distribution $\exp[-0.5(\log x)^2]$ with $m = 1$, $\epsilon = 0.02$; (d) periodically replicated sequence $\{x_i\} = \{14, 9, 22, 5, 20, 2, 14, 9, 22, 5, 20, 2, \dots\}$, (e) with ±10% jitter and (f) with ±100% jitter.

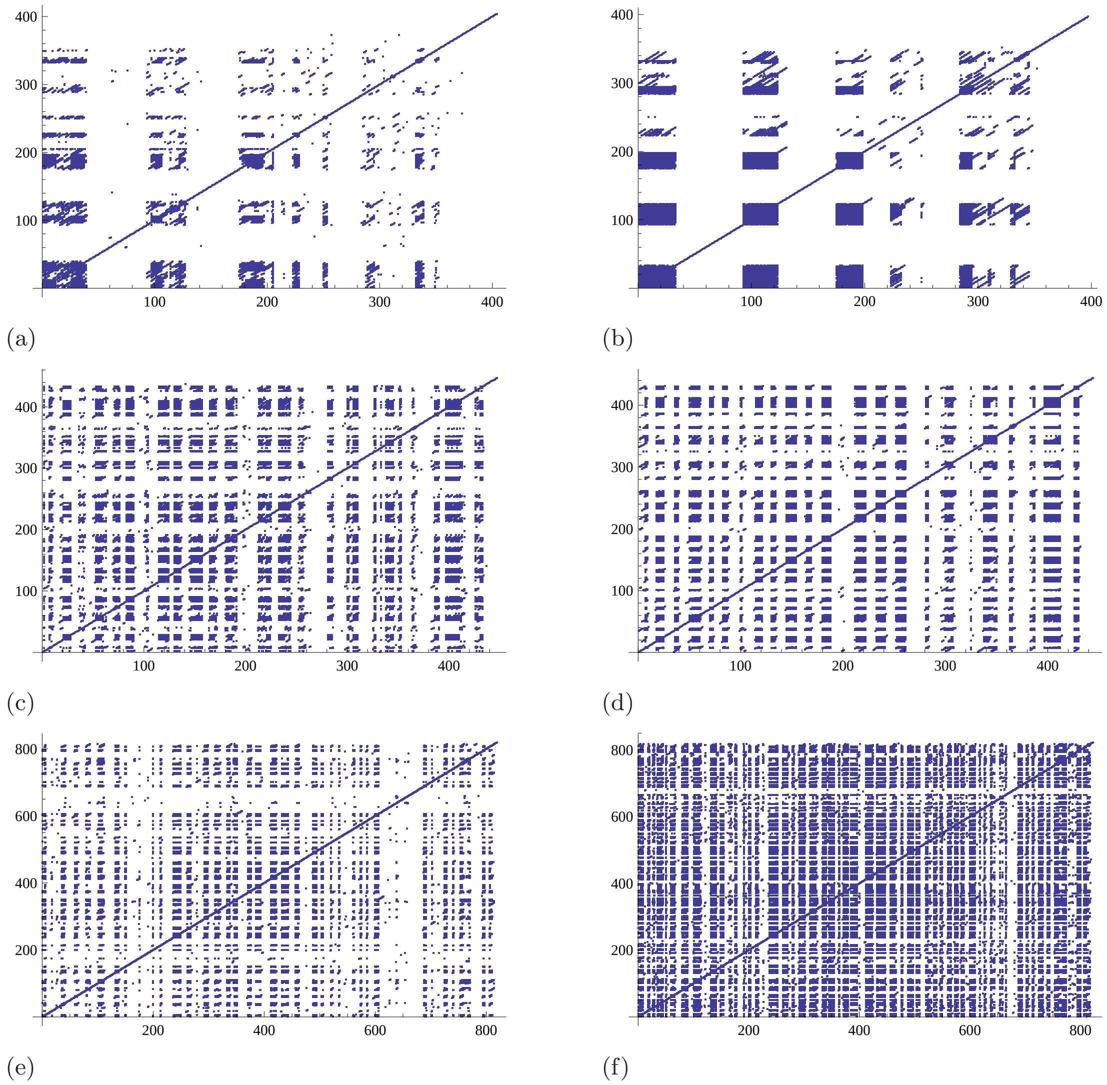


Figure 3.7: Recurrence plots for experimental data DIV5-12; (a) DIV5, channel no. 14, $m = 7$, $\epsilon = 5$ ms; (b) DIV5, no. 14, $m = 14$, $\epsilon = 10$ ms; (c) DIV7, channel no. 14, $m = 4$, $\epsilon = 5$ ms; (d) DIV7, no. 14, $m = 7$, $\epsilon = 10$ ms; (e) DIV12, channel no. 13, $m = 7$, $\epsilon = 30$ ms; (f) DIV12, no. 13, $m = 4$, $\epsilon = 30$ ms.

identification of periodic patterns is much easier in the log-log-correlation plots, one has just to count the number of steps. In the recurrence plots it is sometimes only possible to *guess* the periodicity, as in Fig. 3.6(f). For a quantitative result one has to use averaging methods for the recurrence matrix to determine the periodicity [25].

Further, the detection of periodicities in intra-burst sequences is not possible with the recurrence plot. If the periodicities are interrupted by (long) inter-burst intervals, the recurrences are different for every burst in the plot. Averaging methods would not succeed in this case.

3.3 Average Burst Length Detection

Now a method to determine the average burst length is developed. The number of spikes per burst is calculated with statistical methods. The determination is nontrivial if the bursts are not clearly separated because sometimes inter-burst intervals are filled with noise.

Consider the first $N_s = 280$ spikes from the data recorded at DIV7. A burst-like structure can be seen in Fig. 3.8(a), where both the spike times $\{t_i\}$ and the interspike intervals $\{x_i\}$ are plotted. A burst is defined by a sequence of small consecutive intra-burst intervals $\{x_i^b\} = \{x_1^b, \ldots x_l^b\}$, where l is the length of the burst, i.e. the number of spikes within this burst. The foregoing interspike interval x_0^b and the following one x_{l+1}^b are the large inter-burst intervals. Whereas the product of two intra-burst intervals is small (and the product between an intra-burst interval and an inter-burst interval as well), the product between two inter-burst intervals is large and it can be used for a measure of the burst length. For this purpose x_{i+s} vs. x_i is plotted for a fixed $s > 0$ (Fig. 3.8(b)). Additionally, hyperbola

$$(x - x_{\min})(y - y_{\min}) = p, \tag{3.12}$$

for different *focal parameters* $p = \{10^{-5}, 5 \cdot 10^{-5}, 10^{-4}, 5 \cdot 10^{-4}, 10^{-3}, 3 \cdot 10^{-3}\}$ are plotted.

As already discussed in Sec. 2.3, the duration of an interspike interval can never be smaller than 2 ms, so the offset $x_{\min} = 2$ ms and $y_{\min} = 2$ ms were chosen. The interesting points $\tilde{x}_i^s := (x_i, x_{i+s})$ are those belonging to two different inter-burst intervals, i.e. which have large components both in the x- and in the y-direction. These are points located *above* a hyperbola, whereas most of the points are located *below* the hyperbola. In order to calculate *how far* the points above the hyperbola are away from it, two regions U and W are defined, where U is the region below and W the region above the hyperbola with a certain p,

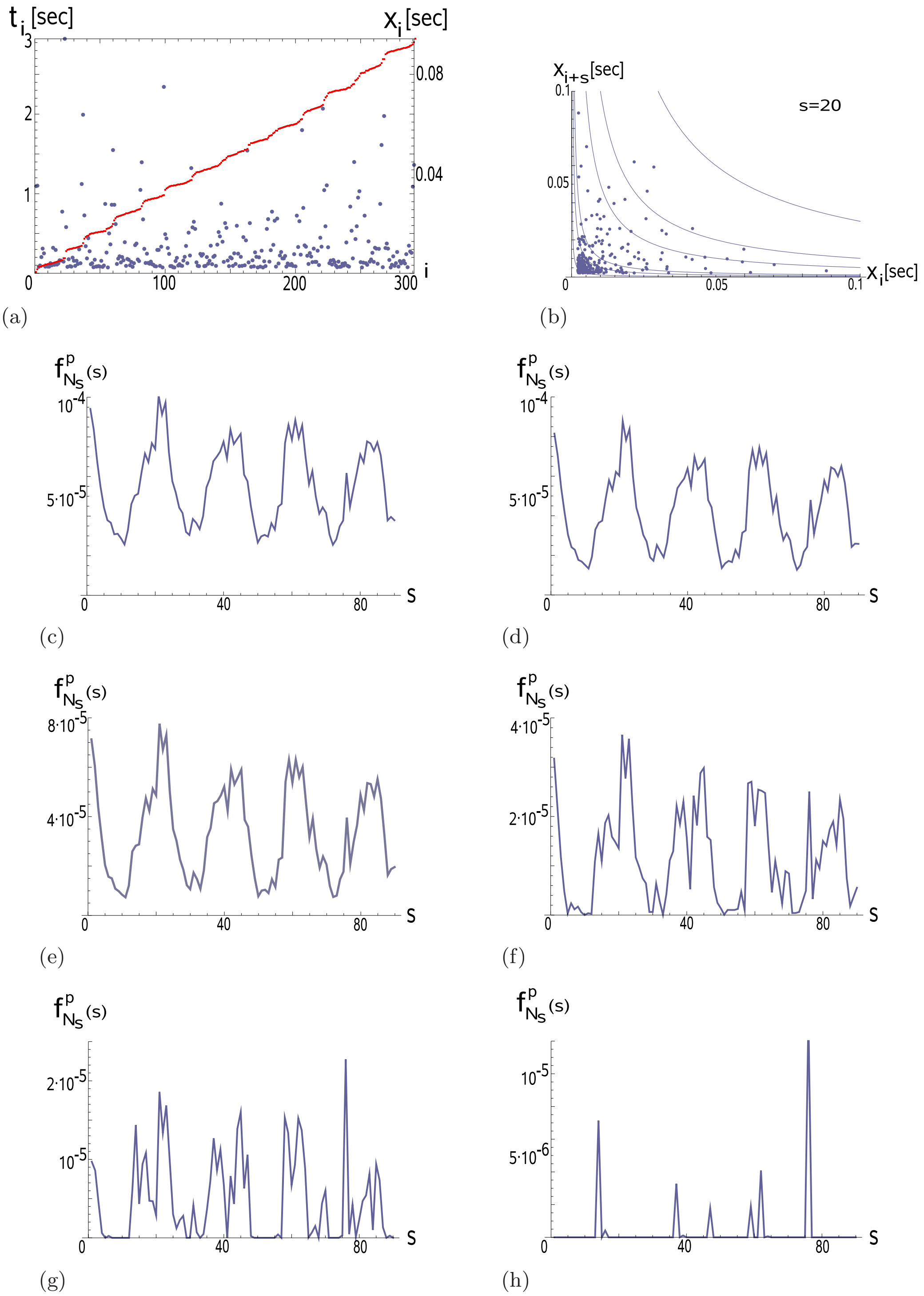
$$(x_i - x_{\min})(x_{i+s} - x_{\min}) < p \rightarrow \tilde{x}_i^s \in U, \tag{3.13}$$

and

$$(x_i - x_{\min})(x_{i+s} - x_{\min}) > p \rightarrow \tilde{x}_i^s \in W. \tag{3.14}$$

The *distance* of a point located in W from the hyperbola is determined and a sum over all of these distances adds up to the function

$$f_{N_s}^p(s) = \frac{1}{N_s} \sum_{\tilde{x}_i^s \in W} \left((x_i - x_{\min})(x_{i+s} - x_{\min}) - p \right). \tag{3.15}$$



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 3.8: (a) Spike times (red) and interspike intervals (blue) for data recorded at channel no. 12 at DIV7. The red curve shows a burst-like structure; (b) x_i vs. x_{i+20} with hyperbola $y = y_{\min} + \frac{p}{x - x_{\min}}$; $f^p_{N_s}(s)$ for $p = 0.00001$ (c), $p = 0.00005$ (d), $p = 0.0001$ (e), $p = 0.0005$ (f), $p = 0.001$ (g) and $p = 0.003$ (h).

Fig. 3.8(c-h) shows $f^p_{N_s}(s)$ for different p. The larger p, the less points are in W. In Fig. 3.8(c-e) there are points $\tilde{x}^s_i \in W$ for all s because p is small enough. The distance between two peaks gives the average burst length. There are 20 events per burst on average. In Fig. 3.8(f,g) p is too large to identify the average burst length clearly, in Fig. 3.8(h) it is impossible.

The above calculations were performed with respect to the interspike intervals and not in the time domain. This makes it possible to calculate *how many* events per burst occur in contrast to the temporal lengths of the bursts. Also in simple neuron models like the integrate-and-fire-model the number of predecessor spikes in one burst is decisive and determines when the neuron is firing again.

Chapter 4

Spatiotemporal Data Analysis

In Chap. 3 data for a specific channel at a *specific* DIV was analyzed. The comparison of *different*, simultaneously recorded signals allows a spatiotemporal analysis of the entire network. For this analysis only pairwise interactions between two channels are considered.

4.1 Autocorrelation Function

The autocorrelation function compares an arbitrary real valued signal function $\bar{x}(t)$, $t \in (0, T)$, with the same signal shifted in time $\bar{x}(t + \tau)$. It can be useful for finding special structures in the signal, e.g. periodicities, even within a noisy environment.

With the mean value

$$\langle \bar{x} \rangle_T = \frac{1}{T} \int_0^T \bar{x}(t)\, dt, \tag{4.1}$$

one can rewrite the signal $x(t) = \bar{x}(t) - \langle \bar{x} \rangle_T$ leading to

$$\langle x \rangle_T = 0. \tag{4.2}$$

An easy way to compare the signal at time t and the delayed signal at $t + \tau$ ($\tau > 0$) is to multiply them. To get detailed information, this multiplication is done for all times $t \in (0, \bar{T})$, $\bar{T} = T - \tau$, and integrated over all t. To get a result which is independent of the time scaling, the integral is divided by $\bar{T}$. In the end one obtains the *autocorrelation function*

$$\tilde{C}_{xx}(\tau) = \frac{1}{\bar{T}} \int_0^{\bar{T}} x(t) x(t + \tau)\, dt = \langle x(t) x(t + \tau) \rangle_{\bar{T}}. \tag{4.3}$$

This definition generally only makes sense for $T \gg \tau$. It is a measure how much the signal *knows* about itself at an earlier time. It is clear that the maximum of $\tilde{C}_{xx}(\tau)$ is for $\tau = 0$,

$$\max_{\tau} \tilde{C}_{xx}(\tau) = \tilde{C}_{xx}(0) = \frac{1}{T} \int_0^T x^2(t))\, dt = \langle x^2(t) \rangle_T, \tag{4.4}$$

which is used as a normalization factor, so

$$C_{xx}(\tau) = \frac{\tilde{C}_{xx}(\tau)}{\tilde{C}_{xx}(0)} = \frac{\langle x(t) x(t + \tau) \rangle_{\bar{T}}}{\langle x^2(t) \rangle_T}, \tag{4.5}$$

which is the *normalized autocorrelation function.*

In reality one can never measure $x(t)$ for all times. Usually the signal is recorded at discrete times $t_i = i\Delta t$, $i = 1, \ldots, N$, and so one uses discrete values $\xi_i = x(i\Delta t)$. The recording time is divided into N bins: $T = N\Delta t$. Instead of integration one sums over all $i \in (0, \bar{N})$, where $\bar{N} = N - n_\tau$ and n_τ is defined via $\tau = n_\tau \Delta t$, so

$$\tilde{C}_{\xi\xi}(\tau) = \frac{1}{\bar{N}} \sum_{i=1}^{\bar{N}} \xi_i \xi_{i+n_\tau} = \langle \xi_i \xi_{i+n_\tau} \rangle_{\bar{N}}. \tag{4.6}$$

The normalization procedure is similar to the continuous case,

$$\max_\tau \tilde{C}_{\xi\xi}(\tau) = \tilde{C}_{\xi\xi}(0) = \frac{1}{N} \sum_{i=1}^{N} \xi_i^2 = \langle \xi_i^2 \rangle_N \tag{4.7}$$

and so

$$C_{\xi\xi}(\tau) = \frac{\tilde{C}_{\xi\xi}(\tau)}{\tilde{C}_{\xi\xi}(0)} = \frac{\langle \xi_i \xi_{i+n_\tau} \rangle_{\bar{N}}}{\langle \xi_i^2 \rangle_N}. \tag{4.8}$$

Examples of autocorrelations for a continuous and a discrete signal are shown in Fig. 4.1. The continuous Gaussian function $x(t) = \exp[-(x-4)^2]$ and a delayed function $x(t+\tau)$ are plotted in Fig. 4.1(a) for $\tau = 1$. The autocorrelation function again has a Gaussian shape (Fig. 4.1(b)). The second example shows the autocorrelation of a discrete noisy sin-function $\xi_i = \sin(i\Delta t) + \eta(i\Delta t)$, where $\eta \in (0, 0.2)$ is white noise (Fig. 4.1(c)). A delayed function is also plotted ($\tau = 1.5$). Since the signal is almost periodic, the autocorrelation function is periodic as well (Fig. 4.1(d)).

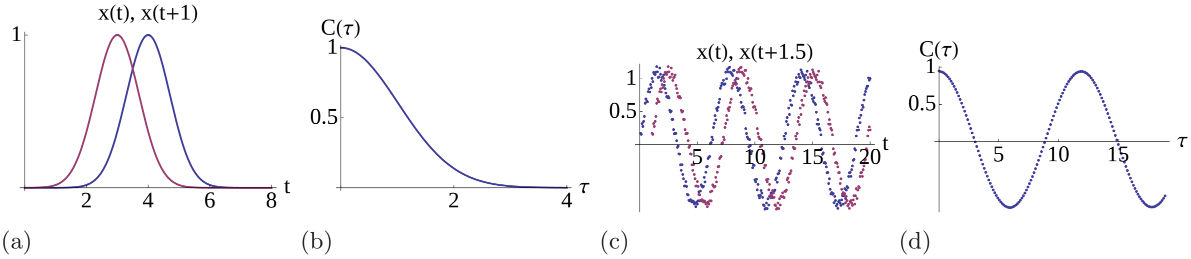


Figure 4.1: (a) Continuous signal $x(t)$ and shifted signal $x(t+1)$ for a Gaussian; (b) corresponding autocorrelation function ; (c) discrete signal x_i with added white noise; (d) autocorrelation function for the noisy sin function.

4.2 Crosscorrelation Function

Consider two signals $x(t)$ and $y(t)$ which are recorded in the same time interval T. The multiplication $x(t)y(t)$ for all times and the following average (integration and division by T) is now a measure how strong the two signals are correlated with each other. Again one can delay one of the signals with τ and calculate the dependence on τ, which gives the *crosscorrelation function*

$$\tilde{C}_{xy}(\tau) = \frac{1}{\bar{T}} \int_0^{\bar{T}} x(t)y(t+\tau)\,dt = \langle x(t)y(t+\tau)\rangle_{\bar{T}}. \tag{4.9}$$

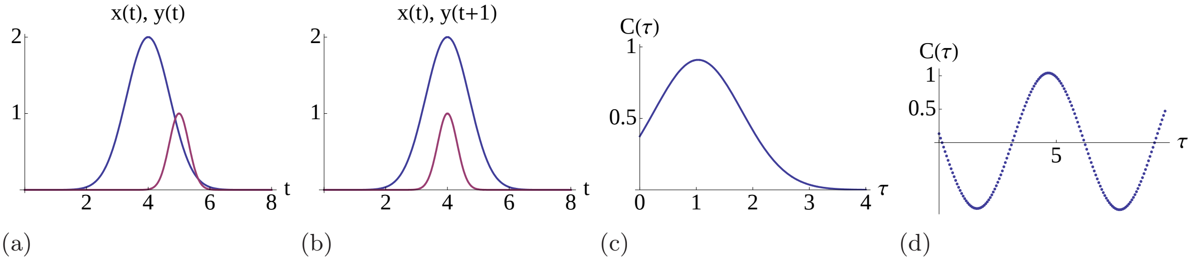
For the comparison of different functions the maximum does not have to be at $\tau = 0$. It is possible that the signals are strongest correlated at $\tau > 0$ or at $\tau < 0$. To illustrate this, consider two Gaussian signals $x(t) = 2\exp[-(x-4)^2]$ and $y(t) = \exp[-5(x-5)^2]$ (Fig. 4.2(a)). For $\tau = 1$ the maxima of both signals are at $t = 4$ (Fig. 4.2(b)). This constellation leads to the highest correlation, i.e. $\max_\tau \tilde{C}_{xy}(\tau) = \tilde{C}_{xy}(1)$. In Fig. 4.2(c) one can clearly see that the maximum of the crosscorrelation function is at $\tau = 1$. With the normalization factor

$$\frac{1}{T}\sqrt{\int_0^T x^2(t)\,dt}\sqrt{\int_0^T y^2(t)\,dt} = \sqrt{\langle x^2(t)\rangle_T}\sqrt{\langle y^2(t)\rangle_T} \tag{4.10}$$

the normalized crosscorrelation function is given by

$$C_{xy}(\tau) = \frac{\langle x(t)y(t+\tau)\rangle_{\bar{T}}}{\sqrt{\langle x^2(t)\rangle_T}\sqrt{\langle y^2(t)\rangle_T}}, \tag{4.11}$$

consistent with the definition of the autocorrelation function (Eq. 4.5).



(a) (b) (c) (d)

Figure 4.2: (a) Two different signals recorded at the same time. The maxima are separated from each other; (b) for $\tau = 1$ the maxima reach the same position in time; (c) the crosscorrelation function shows a maximum at $\tau = 1$; (d) crosscorrelation function for two sin-functions with different amplitudes and phases.

For time-discrete signals a similar procedure as in Sec. 4.1 results in the crosscorrelation function between signals ξ_i and ρ_i,

$$C_{\xi\rho}(\tau) = \frac{\tilde{C}_{\xi\rho}(\tau)}{\sqrt{\tilde{C}_{\xi\xi}(0)}\sqrt{\tilde{C}_{\rho\rho}(0)}} = \frac{\langle \xi_i \rho_{i+n_\tau}\rangle_{\bar{N}}}{\sqrt{\langle \xi_i^2\rangle_N}\sqrt{\langle \rho_i^2\rangle_N}}. \tag{4.12}$$

In Fig. 4.2(d) the crosscorrelation function for $\xi_i = \sin(i\Delta t) + \eta_1(i\Delta t)$ and $\rho_i = 2\sin(i\Delta t + 1.5) + \eta_2(i\Delta t)$ with $\eta_{1,2} \in (0, 0.2)$ is plotted where the maxima are shifted.

Allowing a delay in the negative time direction, i.e. $\tau \to -\tau$, one must be careful with the integration limits. Instead of $t_{\text{low}} = 0$ and $t_{\text{up}} = T - \tau$ we now have to shift the limits to $t_{\text{low}} = \tau$ and $t_{\text{up}} = T$. Due to the fact that it matters which signal is in front or behind the other signal, the crosscorrelation function is not symmetric under the exchange of $x(t)$ and $y(t)$. Instead

$$\tilde{C}_{xy}(\tau) = \tilde{C}_{yx}(-\tau), \tag{4.13}$$

which can be shown easily:

$$\tilde{C}_{xy}(-\tau) = \frac{1}{T-\tau}\int_{\tau}^{T} x(t)y(t-\tau)\,dt = \frac{1}{T-\tau}\int_{0}^{T-\tau} x(t+\tau)y(t)\,dt = \tilde{C}_{yx}(\tau). \tag{4.14}$$

Only the autocorrelations are symmetric,

$$\tilde{C}_{xx}(-\tau) = \tilde{C}_{xx}(\tau). \tag{4.15}$$

4.3 Correlations between Neural Network Spike Frequencies

Spike times of the cultured neural network were measured with a multielectrode array where the signal was recorded separately for every channel. The auto- and crosscorrelation functions were then used to compare the different channels.

The recording time T was split into N bins and within $\Delta t = T/N$ the number of spikes was counted. We denote the *relative spike frequencies* for every channel c in the i-th bin by f_i^c, where $i = 1, \dots, N$. As mentioned in Sec. 4.1 it is reasonable to analyze a shifted signal, which has its mean around zero. Calculating the mean $\langle f^c \rangle_N$ for every channel,

$$\langle f^c \rangle_N = \frac{1}{N}\sum_{i=1}^{N} f_i^c, \tag{4.16}$$

with

$$F_i^c = f_i^c - \langle f^c \rangle_N, \tag{4.17}$$

results in

$$\langle F^c \rangle_N = \frac{1}{N}\sum_{i=1}^{N}(f_i^c - \langle f^c \rangle_N) = \frac{1}{N}\left(\sum_{i=1}^{N} f_i^c - N\langle f^c \rangle_N\right) = \frac{1}{N}\left(N\langle f^c \rangle_N - N\langle f^c \rangle_N\right) = 0. \tag{4.18}$$

Fig. 4.3 shows the vector F_i^j for channels $j = 11, 21$ measured at different DIV. The network is highly synchronized at DIV5 (Fig. 4.3(a)), so the normalized relative frequencies for different channels are similar. The high frequency bins indicate four clear burst sequences. In Fig. 4.3(b) (DIV7) the signals are synchronized as well. For the about first 50 bins the frequencies seem to be rather random, at later times burst sequences occur. At DIV12 (Fig. 4.3(c)) the signals are not synchronized any more.

The correlation function between channels j and k is given by

$$C_{jk}(\tau) = \frac{1}{C_{jk}(0)}\frac{1}{\bar{N}}\sum_{i=1}^{\bar{N}} F_i^j F_{i+n_\tau}^k = \frac{\langle F_i^j F_{i+n_\tau}^k \rangle_{\bar{N}}}{\sqrt{\langle (F_i^j)^2 \rangle_N}\sqrt{\langle (F_i^k)^2 \rangle_N}}, \tag{4.19}$$

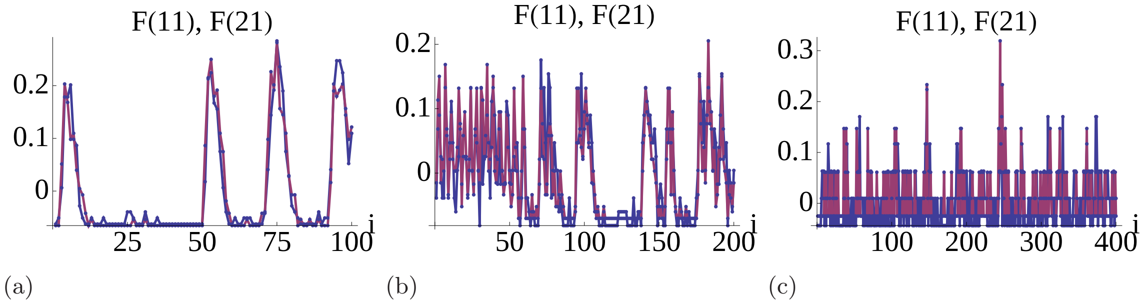


Figure 4.3: $F_i^j/\sqrt{\langle(F_i^j)^2\rangle_N}$ for different channels measured at DIV5 (a), DIV7 (b) and DIV12 (c).

where $\bar{N} = N - n_\tau$, $\tau = n_\tau \Delta t$, $n_\tau = 0, 1, 2, \ldots$. The normalization factor is given by

$$C_{jk}(0) = \sqrt{\frac{1}{N}\sum_{l=1}^{N}(F_l^j)^2}\sqrt{\frac{1}{N}\sum_{m=1}^{N}(F_m^k)^2} = \sqrt{\langle(F_i^j)^2\rangle_N}\sqrt{\langle(F_i^k)^2\rangle_N}. \tag{4.20}$$

Fig. 4.4 shows examples of auto- and crosscorrelation functions, $C_{1,1}(\tau)$, $C_{1,2}(\tau)$ and $C_{23,52}(\tau)$. For DIV5 (Fig. 4.4(a)) all correlations decay similarly with τ. The correlation time is defined via

$$C_{jk}(\tau_c) := \frac{1}{2}\max_{\tau} C_{jk}(\tau), \tag{4.21}$$

and yields $\tau_c \approx 0.25$ s for DIV5(Fig. 4.4(a)), $\tau_c \approx 80$ ms for DIV7 (Fig. 4.4(b)) and about $\tau_c \approx 20$ ms for DIV12.

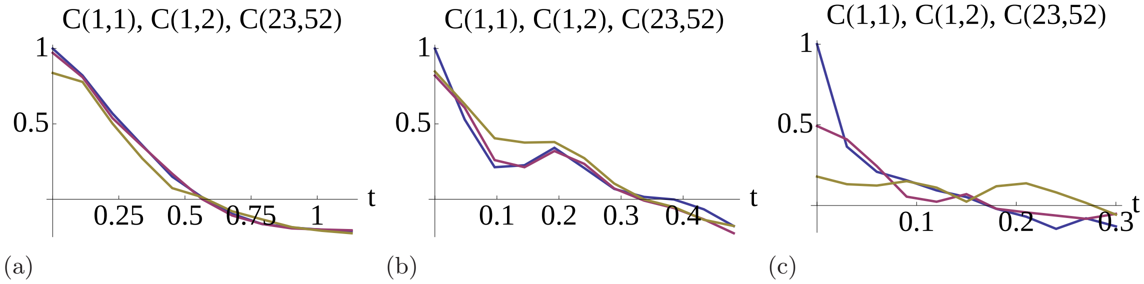


Figure 4.4: Different correlation functions for different days in vitro: $C_{1,1}(\tau)$ (blue), $C_{1,2}(\tau)$ (red), $C_{23,52}(\tau)$ (green) for DIV5 (a), DIV7 (b) and DIV12 (c).

4.4 Correlation Matrix and Network Dynamics

Sec. 4.3 illustrated the usefulness of the correlations $C_{jk}(\tau)$ between all channels. The *strength* of the correlations is characterized by two quantities. On the one hand the height of the crosscorrelation function indicates how strong two signals are correlated at a specific τ. On the other hand the correlation time τ_c is a measure how fast the correlations decay.

An appropriate way to take both quantities of correlation strength measures into account is to integrate the correlation function up to a boundary $\tau_0 = n_0 \Delta t$. Due to the fact that the correlation function (Eq. 4.19) can also take negative values, it is useful to take the time average of the squared function. This results in

$$a_{ij} := \sqrt{\langle C_{ij}^2 \rangle_{n_0}} = \sqrt{\frac{1}{n_0} \sum_{n=1}^{n_0} C_{ij}^2(n\Delta t)}, \tag{4.22}$$

where τ_0 has to be chosen properly. If $\tau_0 < \tau_c$, long time information is lost, if $\tau_0 \gg \tau_c$, noise dominates the correlations for large τ. A rule of thumb is to set τ_0 a few times larger than τ_c, keeping in mind that τ_0 has to be the same for all correlation pairs.

The *correlation matrix* a with the entries a_{ij} gives a measure for the correlations between all channels. It can be visualized by plotting the entries a_{ij} in different colors belonging to different correlation strengths (Fig. 4.5(left)). For every plot the values are scaled such that the maximum value is dark red, the minimum value is dark blue, and the values between are yellow and white. So a red (and yellow) spot means strong correlation and blue means weak correlation. In Fig. 4.5(a) (DIV5) vertical and horizontal red lines indicate that some channels are strongly correlated with most of the other channels, and the few blue lines show that some channels are correlated weakly with most of the other ones. The fact that the autocorrelations do not dominate over the crosscorrelations denotes a highly synchronized network. At DIV7 (Fig. 4.5(c)) a diagonal line emerges which means that the autocorrelations become more dominant and the network is not so strongly synchronized any more. This effect becomes even more drastic at DIV12 (Fig. 4.5(e)), where the network is not synchronized any more and the autocorrelations strongly dominate over the crosscorrelations.

In order to explain the network dynamics, Eq. 4.22 is applied to a system containing only two clear signals, namely $x(t)$ and $y(t)$ from Fig. 4.2(a). Assuming the two signals were recorded at two different spatially separated locations, in analogy to different sites in a neural network. The analysis of this simple system allows to interpret the asymmetry of a.

Fig. 4.6(a) shows all possible correlation functions $C_{ij}(\tau)$, $i = x, y$, $j = x, y$. The autocorrelations C_{xx} (blue) and C_{yy} (red) are symmetric with respect to τ. Due to the fact that $x(t)$ is broader than $y(t)$, C_{xx} is broader than C_{yy}, or $a_{xx} > a_{yy}$, i.e. "*x is more strongly correlated than y*", which can be seen quantitatively via integration ($\tau_0 = 4$),

$$a_{xx} = \sqrt{\frac{1}{\tau_0} \int_0^{\tau_0} C_{xx}^2(\tau)\, d\tau} = 0.969, \qquad a_{yy} = \sqrt{\frac{1}{\tau_0} \int_0^{\tau_0} C_{yy}^2(\tau)\, d\tau} = 0.638. \tag{4.23}$$

Even more interesting is the interpretation of the cross terms C_{xy} and C_{yx}. The peak of $x(t)$ heads that of $y(t)$ which is delayed. This can be seen in Fig. 4.6(a), where C_{xy} (orange) has the peak at $\tau > 0$ and C_{yx} (green) has its peak at $\tau < 0$. After integration,

$$a_{xy} = \sqrt{\frac{1}{\tau_0} \int_0^{\tau_0} C_{xy}^2(\tau)\, d\tau} = 1.050, \qquad a_{yx} = \sqrt{\frac{1}{\tau_0} \int_0^{\tau_0} C_{yx}^2(\tau)\, d\tau} = 0.188, \tag{4.24}$$

and the full correlation matrix is determined by

$$a = \begin{pmatrix} 0.969 & 1.050 \\ 0.188 & 0.638 \end{pmatrix}. \tag{4.25}$$

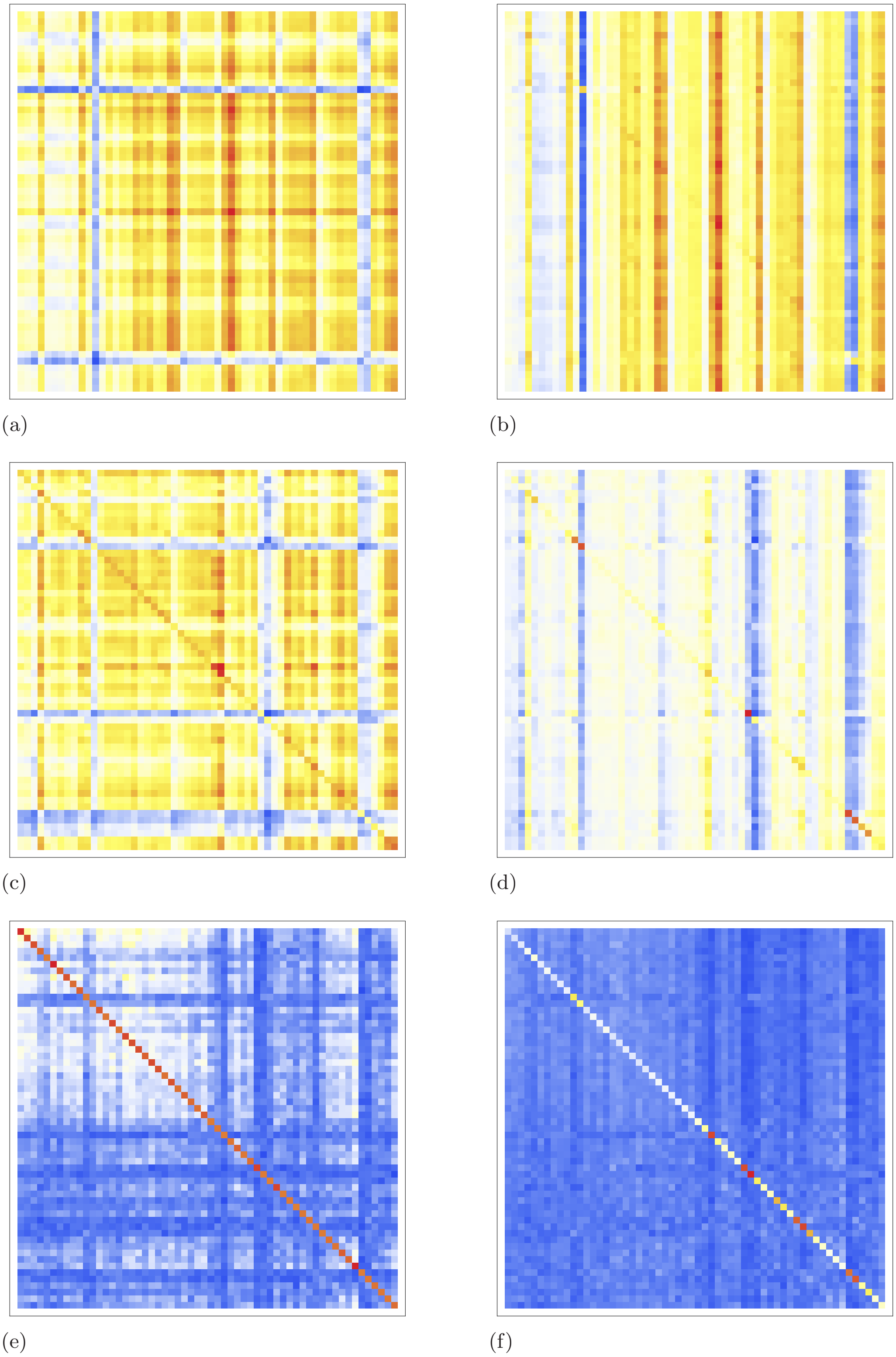
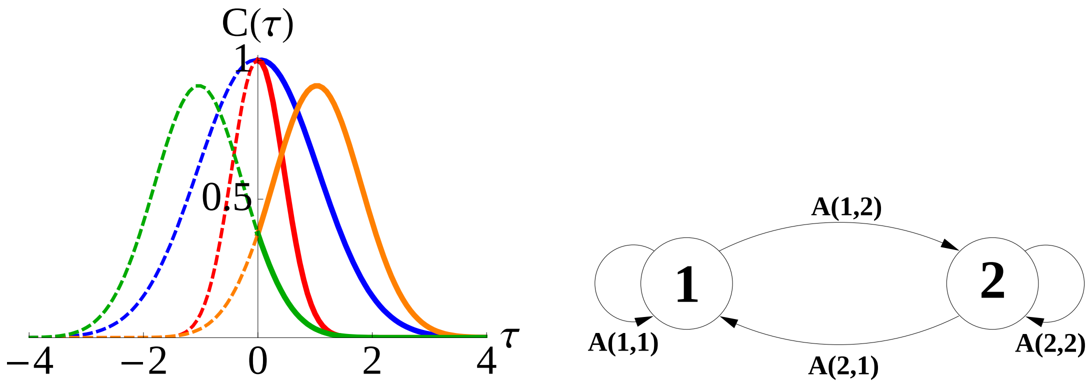


Figure 4.5: Correlation matrix a_{ij} for DIV5 (a), DIV7 (c) and DIV12 (e) and transfer matrix A_{ij} for DIV5 (b), DIV7 (d) and DIV12 (f).

As $a_{xy} > a_{yx}$, *"x strongly affects y"* and *"y weakly affects x"*. If a signal $\tilde{y}(t)$ appears after $\tilde{x}(t)$ has already vanished, $a_{yx} = 0$, and one would say *"x totally affects y"* and *"y does not affect x at all"*. There is a *net-information flow* from site one (with the corresponding signal $x(t)$) to site two (with corresponding signal y(t)). This reflects the fact that the correlation matrix is asymmetric. Additionally, if the autocorrelations are much stronger than the crosscorrelations, no information is transported. And if all entries in a are almost the same, the sites are highly synchronized.



(a)

(b)

Figure 4.6: (a) Correlation functions $C_{xx}(\tau)$ (blue), $C_{xy}(\tau)$ (orange), $C_{yx}(\tau)$ (green) and $C_{yy}(\tau)$ (red) for $x(t) = 2\exp[-(x-4)^2]$ and $y(t) = \exp[-5(x-5)^2]$; (b) Markov process from transfer matrix $A(i,j)$.

In general, information at site i can *stay* at site i with probability p_{ii} and it can *move* with probability p_{ij}, $j \neq i$, where the probabilities have to fulfill the normalization condition

$$\sum_j p_{ij} = 1 \quad \forall i. \tag{4.26}$$

The entries from a are now associated with probabilities. The larger the entries a_{ij} in the correlation matrix, the larger the probabilities p_{ij}. For this purpose a is normalized as well and one can define a *transfer matrix* A (see for example [27]),

$$A_{ij} = \frac{a_{ij}}{\sum_j a_{ij}}, \tag{4.27}$$

that fulfills

$$\sum_j A_{ij} = 1 \quad \forall i, \qquad 0 \leq A_{ij} \leq 1 \quad \forall i,j, \tag{4.28}$$

and the entries A_{ij} can be identified as the probabilities p_{ij}. The transfer matrix defines a *Markov process*, which generates the spatiotemporal dynamics of the system. For our example (Fig. 4.2(a), Fig. 4.6(a)) the transfer matrix A corresponding to a (Eq. 4.25) is given by

$$A = \begin{pmatrix} 0.480 & 0.520 \\ 0.228 & 0.772 \end{pmatrix}. \tag{4.29}$$

The Markov process is schematically shown in Fig. 4.6(b).

Now this method is applied to recorded data for DIV5-12. The transfer matrix A is plotted in Fig. 4.5(right) where every colored spot stands for a probability. For DIV5 (Fig. 4.5(b)) some red vertical lines which characterize *absorbing* sites within the network can be seen. Blue vertical lines indicate sites which are unlikely to be reached. In terms of a Markov process the information flows to the absorbing states. At DIV7 (Fig. 4.5(d)) a red and yellow diagonal line appears which indicates that the information is now more likely to stay on its site. The red vertical lines have vanished, there are no global absorbing states any more. For DIV12 (Fig. 4.5(d)) all transition probabilities are much smaller than the diagonal entries of the transfer matrix. So the probabilities to stay at a certain site dominate. Information is not flowing any more which means that the network is not synchronized.

It is illustrative to examine the eigenvalue spectrum λ_i of A as well. The Perron-Frobenius theorem, which holds for $N \times N$ matrices with real non-negative entries $A_{ij} \geq 0$, states that there is a real largest eigenvalue $\lambda_{\max} > |\lambda_i|$ for which the following proposition holds:

$$\min_i \sum_j A_{ij} \leq \lambda_{\max} \leq \max_i \sum_j A_{ij}. \tag{4.30}$$

Due to Eq. 4.28, the largest eigenvalue $\lambda_{\max} = 1$, and the spectrum of eigenvalues is restricted to

$$1 = \lambda_{\max} \geq \cdots \geq |\lambda_k| \geq \cdots \geq |\lambda_{\min}| \geq 0. \tag{4.31}$$

4.5 Correlation Weighting

With Eq. 4.22 a way to calculate the correlation strengths a_{ij} was defined. This is of course only one possible choice. With a different correlation measure one would generally get a different matrix a. To take this into account a more general correlation matrix is introduced. One way to do this is re-weighting the correlations. This allows to tune the correlation measure. Instead of only considering a_{ij}, the correlations are taken to the power of a $\beta \in \mathbb{R}$,

$$a_{ij} \to a_{ij}^{\beta}. \tag{4.32}$$

Tuning β from $-\infty$ to $+\infty$ gives different relative weights of the correlations. For $\beta = 1$ one gets the *natural* correlation matrix. When β is larger than one, the larger entries are more important than the smaller ones. At $\beta = 0$ all correlations contribute with the same weight, i.e.

$$a_{ij}^{0} = 1 \quad \forall i, j, \tag{4.33}$$

no matter how the actual correlations look like. For $\beta < 0$ the smaller correlations become more important than the larger ones. This is a way for finding out which correlations are less dominant in the network.

Again one can define a transfer matrix, similar as in Eq. 4.27,

$$\mathcal{A}_{ij}(\beta) = \frac{a_{ij}^{\beta}}{\sum_j a_{ij}^{\beta}}. \tag{4.34}$$

So there are different Markov processes for different β. For $\beta \to +\infty$ only the largest probability per row in $\mathcal{A}$ will survive, whereas at $\beta = 0$ one gets uniform probabilities,

$$\mathcal{A}_{ij}(0) = \frac{a_{ij}^0}{\sum_j a_{ij}^0} = \frac{1}{N} = \text{const}, \tag{4.35}$$

every path of a Markov process has the same probability. In fact one gets a different Markov process for every β. High β means considering the more probable paths.

The eigenvalue spectrum λ_i for $\mathcal{A}$ is now β dependent. Due to the fact that the Perron-Frobenius theorem (Eq. 4.30) still holds, the largest eigenvalue is still one for every β, all other eigenvalues change with β.

One way to describe the Markov process with a *global* property is the trace of $\mathcal{A}$,

$$\operatorname{Tr} \mathcal{A}(\beta) = \sum_{i=1}^{N} \mathcal{A}_{ii}(\beta) = \sum_{i=1}^{N} \lambda_i(\beta). \tag{4.36}$$

Due to the fact that $\mathcal{A}$ (and its trace) is real valued, all possible complex eigenvalues have to show up in pairs $\lambda_k = \mathrm{Re}(\lambda_k) + i\mathrm{Im}(\lambda_k)$ and $\bar{\lambda}_k = \mathrm{Re}(\lambda_k) - i\mathrm{Im}(\lambda_k)$. So one only has to consider the summation over the real parts of the eigenvalues in Eq. 4.36. Denoting the trace by $Z(\beta)$, one gets

$$Z(\beta) = \operatorname{Tr} \mathcal{A}(\beta) = \sum_{i=1}^{N} \mathrm{Re}(\lambda_i(\beta)). \tag{4.37}$$

The eigenvalues can be interpreted as *microstates* of the underlying system whereas $Z(\beta)$ is a global property of $\mathcal{A}$. It is the *sum* over all microstates and plays the rôle of a *partition function* (*Zustandssumme*).

In statistical physics the partition function is an important tool to calculate thermodynamic quantities. The real parameter β is a thermal variable such as the inverse temperature or any other intensive variable (see for example [27]). Usually the partition function is not fully known due to the large number of particles ($\approx 10^{23}$).

4.6 Phase Transitions

In physics phase transitions (for an overview see for example [28]) occur, if by a change of a thermal variable the behavior of the system changes drastically. This happens at a so-called *critical point*. It is also possible that systems have more than one of such points. Important examples are transitions between solid, liquid and gas states, transitions in magnetic systems (e.g. between paramagnetic and ferromagnetic states), transitions between normal and superconducting states or the transition between normal and superfluid He^4.

Mathematically phase transitions are characterized by a non-analyticity of the partition function or the derivative of it. In physics usually the non-analyticity of derivatives of the *free energy* of a system is the quantity to look at. A phase transition of the order l occurs when the l-th derivative of $\log Z$ with respect to the thermal variable does not exist, whereas the (l-1)-th derivative exists. Due to the fact that the partition function is a sum of N smooth functions of β, a non-analyticity only occurs in the limit $N \to \infty$ (*thermodynamic limit*). In real systems N is very large ($\approx 10^{23}$) but finite. The phase transition is then not characterized by a non-analyticity but by a continuous change in the system.

In analogy to physics, the behavior of $\log Z(\beta)$ with respect to β is a measure to detect transitions in the neural network. Due to the small $N = 60$ one is not able to see a real phase transition. Nevertheless the slope of $\log Z$ is employed to detect these "non-analyticicities". At a critical point the slope changes drastically, and in analogy to physics, the system is changing from one *phase* to another one. The strength of the transition is given by the change of the slope of $\log Z$ at the critical point, i.e. by the second derivative of $\log Z$.

Now $\log Z(\beta)$ for a sufficiently large range of β is considered and the first and second derivatives are calculated to detect possible transitions. Fig. 4.7 shows $\log Z(\beta)$ (blue), the first derivative (red) and the second derivative (brown) for DIV5-12, where the values of the derivatives are rescaled for better visualization.

At DIV5 (Fig. 4.7(a)) $\log Z(\beta)$ is shown for positive and negative β, where $\log Z(\beta) = \log \operatorname{Tr} \mathcal{A}(\beta)$ increases with increasing β, so the entries in the diagonal become more and more important. With higher β the probability to move from one site to another shrinks on average. In Fig. 4.8(a) the transfer matrix is plotted for five different values of β. The blue picture in the middle is at $\beta = 0$ where every process occurs with the same probability. For $\beta = 10$ absorbing sites occur (red and yellow vertical lines) and at $\beta = 40$ one specific site attracts most of the flow. For $\beta \to +\infty$ we would get a completely blue picture with except one red spot per line in the matrix. For negative β only one channel becomes dominant in the limit $\beta \to -\infty$. The derivate of $\log Z$ in Fig. 4.7(a) now tells us *how* the system is changing by varying β. It shows a step-like structure which indicates a critical point. The second derivative hints at $\beta_c \approx -14$. The height $h_c \approx 0.025$ of the second derivative at β_c is a measure for the strength of the transition, i.e. for the curvature of $\log Z$ at the critical point.

For DIV7 (Fig. 4.7(b)) $\log Z$ saturates faster with increasing β, compared to DIV5. The Markov process converges faster to a specific process and the amount of red spots in Fig. 4.8(b) for $\beta = 40$ is higher than for DIV5. The derivative in Fig. 4.7(b) shows two slopes corresponding to three different phases. The second derivate has two peaks at $\beta_{c_1} \approx -7$ and $\beta_{c_2} \approx 10$, where the heights $h_{c_1} \approx 0.15$ and $h_{c_2} \approx 0.08$ are much larger than for DIV5. So the transitions are stronger, the system is more sensitive to a change of β.

DIV12 (Fig. 4.7(c)) is even more drastic, again one can detect two transitions at $\beta_{c_1} \approx -1.8$ and $\beta_{c_2} \approx 2.1$ with $h_{c_1} \approx 1.1$ and $h_{c_2} \approx 2.3$. Due to the small β_c the transitions occur very fast. The diagonal elements of $\mathcal{A}$ are already much larger than the off-diagonal terms for $\beta = 1$ (see Fig. 4.5(f)), and the picture does not change for higher β (see Fig. 4.8(c)). The partition function converges to

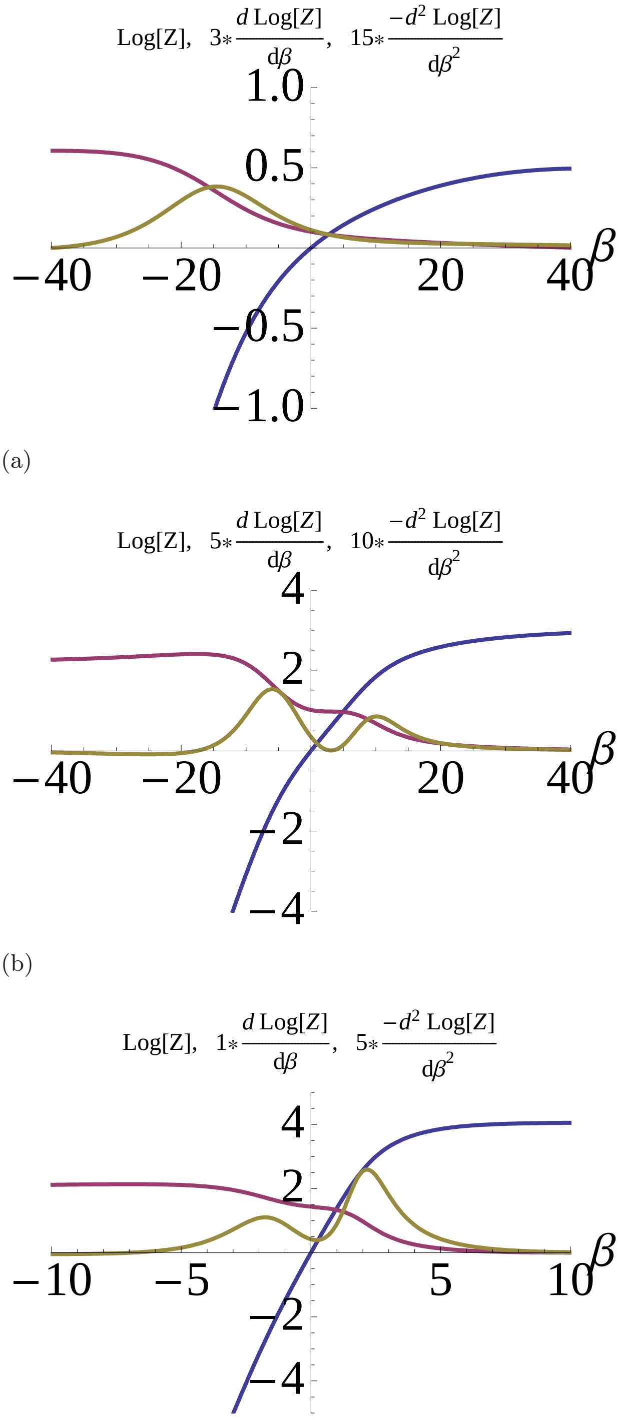
$$\log Z(\beta \to \infty) \to \log N \approx 4.1, \tag{4.38}$$

because all crosscorrelations die out with high β, so

$$\mathcal{A}_{ii}(\beta \to \infty) \to 1 \qquad \text{and} \qquad \mathcal{A}_{ji}(\beta \to \infty) \to 0 \quad \text{for} \quad i \neq j. \tag{4.39}$$

To illustrate the differences between the three data sets more precisely, the corresponding $\log Z$ and its derivatives are displayed in one plot on the same scale, respectively. Fig. 4.9(a) shows $\log Z$ for DIV5 (blue), DIV7 (red) and DIV12 (green). Due to the fact that

$$\log Z(\beta = 0) = \log 1 = 0, \tag{4.40}$$



(a)

(b)

(c)

Figure 4.7: Partition function $\log Z$ (blue) and its first (red) and second (brown) derivative for DIV5 (a), DIV7 (b), DIV12 (c).

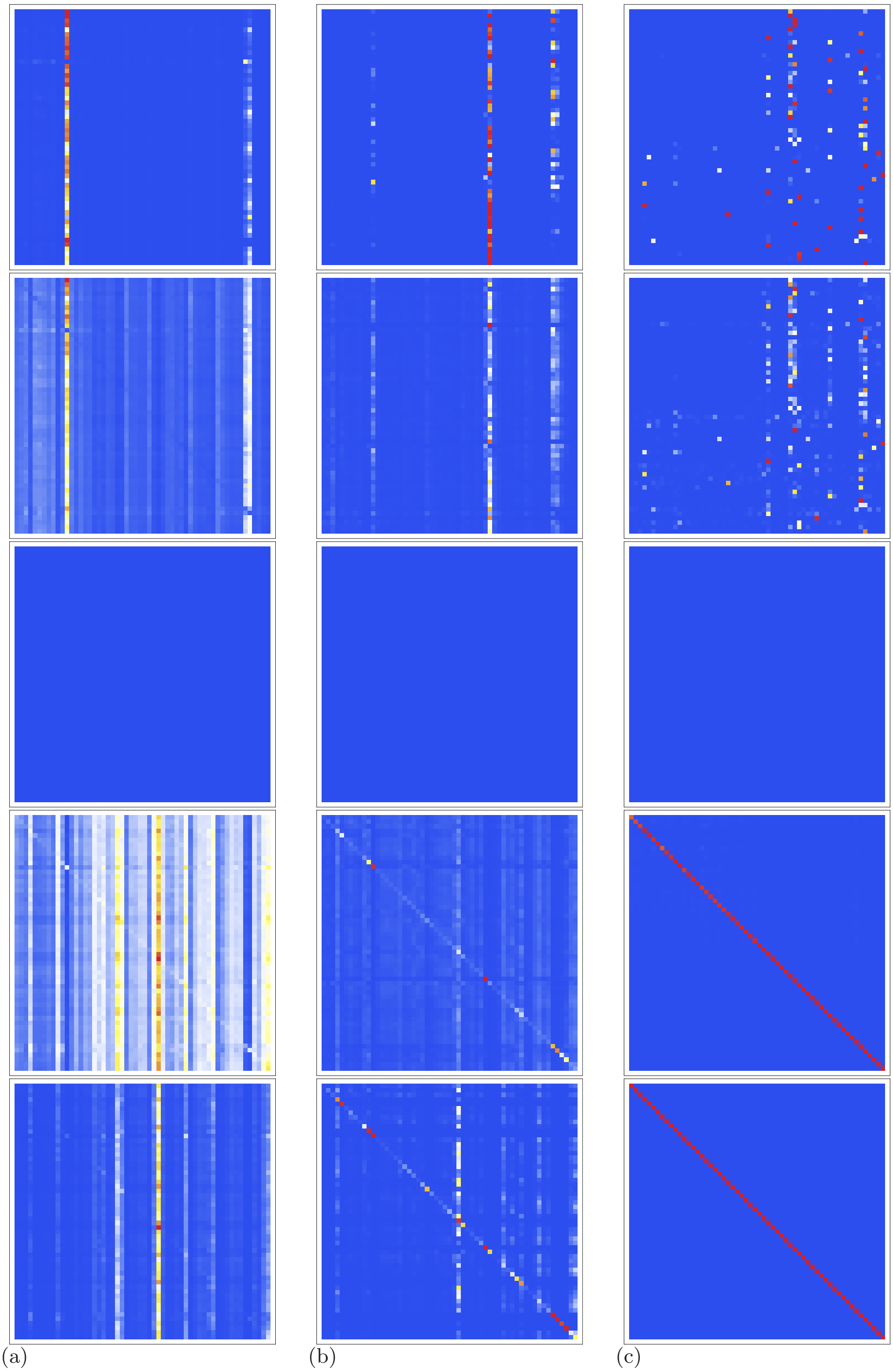
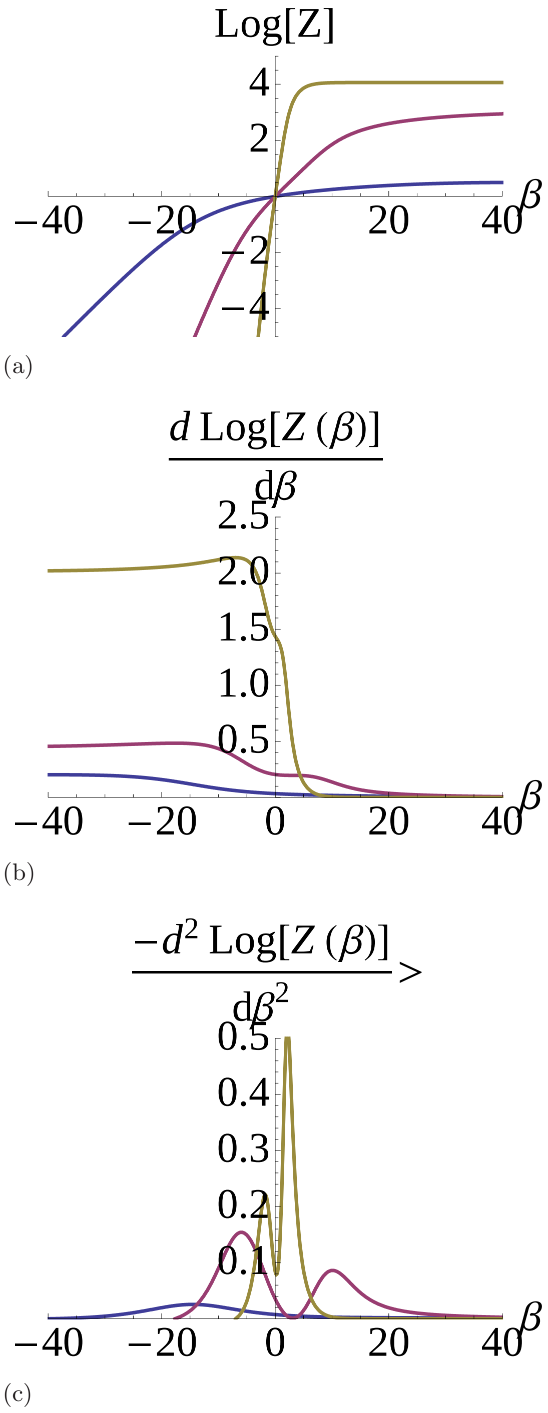


Figure 4.8: Transfer matrix $\mathcal{A}(\beta)$ for different β. In line one $\beta = -40$, in line two $\beta = -10$, in line three $\beta = 0$, in line four $\beta = 10$ and in line five $\beta = 40$. (a) DIV5; (b) DIV7; (c) DIV12.

the curves cross each other at $\beta = 0$. Fig. 4.9(b) shows the first derivative of $\log Z$. While for DIV5 and DIV7 the steps are not well pronounced, indicating that the phases are not clearly separated, the steps for DIV12 are quite steep, so three phases can be seen. This is reflected by the second derivative (Fig. 4.9(c)) which has the highest peaks for DIV12.

The correlation weighting parameter β is a variable which provides different ways of measuring the correlations. Depending on β, a certain Markov processes reflects the spatiotemporal evolution of the system. *Choosing* a small positive β, the correlations a_{ij} do not differ much, while a larger β implies larger differences in the correlations and for the transfer matrix, respectively. Far away from the critical point the result is rather insensitive to a specific β. Close to a transition point the correlations change drastically with β, especially for an asynchronous network. The less synchronized the network the more sensitive the correlation measurement around the critical β_c. A synchronized network is less sensitive to the correlation measure, the transition from one phase to another proceeds slowly.



(a)

(b)

(c)

Figure 4.9: (a) $\log Z$ for DIV5 (blue), DIV7 (red) and DIV12 (brown); (b) $\frac{\partial \log Z}{\partial \beta}$ for DIV5 (blue), DIV7 (red) and DIV12 (brown); (c) $-\frac{\partial^2 \log Z}{\partial \beta^2}$ for DIV5 (blue), DIV7 (red) and DIV12 (brown).

Chapter 5

Summary and Outlook

This work analyzed neural spike time signals from cultured in vitro neural networks. The data results from multielectrode array (MEA) experiments which have been carried out at EMPA. An in vitro neural network of chicken motor neurons was cultured on one side of the MEA and myofibrils were grown on the opposite side to measure the activity between the neurons and the myofibrils. Due to the spatial separation of the recorded channels, a spatiotemporal analysis of the neural signals in the network has been performed.

Statistical methods such as the correlation integral and recurrence plots were used to evaluate interspike intervals resulting from a specific MEA channel. To analyze the interspike intervals they were embedded in an m-dimensional space. While these methods work in many examples, their application to the neural data does not result in satisfying new insights. Outliers (large interspike intervals) between bursts are responsible for artificial structures in the correlation integral.

Considering the spatial structure, simultaneously recorded signals from all channels have been compared by calculating correlation functions $C_{ij}(\tau)$ between two channels i and j. Integrating C_{ij}^2 results in a correlation matrix a_{ij} which is a measure for the strength of the correlations. Due to the fact that the crosscorrelation functions are asymmetric, which results in an asymmetric a_{ij} as well, one can define a net-information flow between two channels. By interpreting the flow as a Markov process, a corresponding transfer matrix A_{ij} was defined which describes a spatiotemporal dynamics in the network.

Taking the correlations a_{ij} to the power of a real parameter β reflects different ways to measure the correlations resulting in different Markov processes. By considering the β-dependent trace of the matrix a_{ij} the system can be characterized by a single quantity, the partition function $Z(\beta)$. In analogy to statistical physics, transitions in the system occur by changing β.

We have shown that at early stage (day five) of the in-vitro experiment the neural network is strongly synchronized, while at later times (day twelve) the network is less synchronized. This is reflected by the strengths of the transitions, i.e. by the peaks of the second derivatives of $\log Z(\beta)$, which turned out to be much higher for the data recorded at the twelfth day than for the fifth day. Our analysis has shown that the transitions in a synchronous network are rather weak, while an asynchronous network shows phase-transition-like behavior at critical β_c. In the latter case the correlation measurements at slightly different values of β result in large differences for the correlations near a critical β_c. Since the number of channels is 60 only, we were not able not detect sharp phase transitions.

For future investigations it would be interesting to study systems which are stimulated locally but not globally by a chemical solution as in the underlying experiments. An advantage of MEAs is that they both measure action potentials and can also be used to stimulate neurons. With such a local stimulation it should be possible to determine a direction of information flow. As a result, one may obtain some knowledge about the transport mechanism of electric signals within the network. Maybe it would hence be possible to detect spatiotemporal patterns, e.g. *avalanches* [29].

Appendix A

Mathematica Code

```
In[1]:=   div5 := Import[
             "/home/andreas/Desktop/DA090305/Diplomarbeit/ExperimentData/DIV5/DIV5-60.xls",
             "XLS"];

In[2]:=   div7 := Import[
             "/home/andreas/Desktop/DA090305/Diplomarbeit/ExperimentData/DIV7/DIV7-60.xls",
             "XLS"];

In[3]:=   div12 := Import[
             "/home/andreas/Desktop/DA090305/Diplomarbeit/ExperimentData/DIV12/DIV12-60.xls"
             , "XLS"];

In[4]:=   d = 7;

In[5]:=   div = Which[d == 5, div5, d == 7, div7, d == 12, div12];

In[6]:=   divStr = "div" <> ToString[d];

In[7]:=   nomin = 1; (*channel no. Start*)

In[149]:=  nomax = Which[d == 5, 57, d == 7, 58, d == 12, 59];

In[10]:=  bord = Which[d == 5, 0.6, d == 7, 0.4, d == 12, 0.2];
           (*Integration time_max for correlation averaging*)

In[11]:=  NBins = Which[d == 5, 100, d == 7, 200, d == 12, 400];(*no. of bins*)

          taumin = 0; taumax = Which[d == 5, NBins / 10, d == 7, NBins / 20, d == 12, NBins / 40];

In[14]:=  T = Which[d == 5, 11.3, d == 7, 9.6310, d == 12, 12.];

In[15]:=  deltaT = T / NBins;

In[16]:=  loc0 = "/home/andreas/Desktop/DA090305/Diplomarbeit/";

In[17]:=  loc1 = loc0 <> "ExperimentResults/adjDIV" <> ToString[d] <> "/";

In[18]:=  loc2 = "/local/d0/zoettl/adjDIV" <> ToString[d] <> "/";

In[19]:=  ext1 = "." <> ToString[nomin] <> "to" <> ToString[nomax] <> "." <> divStr <> ".dat";

In[20]:=  ext3 = "." <> ToString[nomin] <> "to" <> ToString[nomax] <> "." <> divStr <> ".eps";
```

```
In[21]:= asf = ColorData["TemperatureMap"];

In[22]:= BinStr = ".N" <> ToString[NBins];

In[23]:= cutz = 0;

In[50]:= mmin = 1; mmax = 1; mstep = 1; (* Embadding dimension m; for output normIJ.dat *)
         m0 = 1; (* actual embadding dimension m (e.g. for recurrence plots) *)
         ch = nomin; (* selected channel *)
         maxISI = 1000.; minISI = 0.; tmin = 0.; tmax = 10 005.0;
         nmin = 1; ncut = 30 000; (* ISI cutoff *)
         normIJout = 0; (* =1: embadded points output, e.g. for Corr-Integral*)
         spar = 0; smax = 100; sval = 20; (* =1: create x_i vs x_{i+s} plots;
         s-range (smin=0) and specific s *)
          (*p={0.00001,0.00005,0.0001,0.0005,0.001,0.003,0.05};*)
         p = {0.0004, 0.001, 0.005, 0.01, 0.03, 0.05}; delta = 0.002;
         parn = p[[4]];(* hyperbola xy=p; for x_i vs x_{i+s} plots; *)
         outpar = 0; (* =1: times and ISI output *)
         recplpar = 0; (* =1: create recurrence plots *)
         ps = PlotStyle -> PointSize[0.01];

In[36]:= emb := Module[{m, i, j, hh, k}, Print["Start Embedding"]; lnumb = Length[data[[no]]];
            nnumb := lnumb - m + 1; pp = Table[data[[no, i]], {i, lnumb}, {j, 1}];
            Do[
             ll = pp;
             k = 1;
             While[
              k < m,
              ll = Table[Append[ll[[hh]], data[[no, hh + k]]], {hh, nnumb}];
              k ++]; bTab[m] = ll; aTab[m] = Prepend[ll, {m, nnumb}]; bTL[m] = Length[bTab[m]],
             {m, mmin, mmax, mstep}]; Print["End Embedding"]];

In[37]:= normM := Module[{m}, Print["Start Norm"]; Do[norm[m] =
              Table[Max[Abs[bTab[m][[i]] - bTab[m][[j]]]], {i, 1, bTL[m]}, {j, 1, bTL[m]}];
             maxnorm[m] = Max[norm[m]], {m, mmin, mmax, mstep}]; Print["End Norm"]];

In[38]:= corrIntM := Module[{i, j}, Print["Start Correlation Integral"];
           eps0min = -3; eps0max = 0; eps0step = 0.05;
           CorrInt = Table[{eps, Log[(1 / (bTL[m] * (bTL[m] - 1))) *
                (Sum[UnitStep[Exp[eps] - norm[m][[i, j]] / maxnorm[m]],
                   {i, 1, bTL[m]}, {j, 1, bTL[m]}] - bTL[m])]},
             {m, mmin, mmax, mstep}, {eps, eps0min, eps0max, eps0step}];
           Print["End Correlation Integral"]
          ]

In[39]:= dataB :=
           Module[{j}, data = Append[data, Select[Table[div1c[[no, j + 1]] - div1c[[no, j]],
                {j, 1, nmax - nmin}], minISI < # < maxISI &]]];

In[46]:= recplM := Module[{rec, rec2, recpl, i, j, eps}, emb; normM; eps = 0.001;
           rec = Flatten[Table[{norm[m0][[i, j]], i, j}, {i, 1, bTL[m0]}, {j, 1, bTL[m0]}], 1];
           rec2 = Cases[rec, {x_ /; x < eps, _, _}][[All, {2, 3}]];
           Print[recpl = ListPlot[rec2, PlotStyle → {PointSize[0.001]}]];
           outname0 = ToString[no] <> "." <> ToString[no] <> "." <> divStr;
           RecPlOut = "recpl." <> outname0 <> ".m" <> ToString[m0] <>
             ".e" <> ToString[eps * 1000] <> "eps"; Export[RecPlOut, recpl];]
```

```
sxii := Module[{s, k},
  xii = {}; xiiS = {}; xiiT = {}; xiipl = {}; Do[
   xii = Append[xii,
     Table[{data[[no, k]], data[[no, k + s]]}, {k, 1, Length[data[[no]]] - s}]];
   sel := Cases[((xii[[s, All, 1]] - delta) * (xii[[s, All, 2]] - delta) - parn),
     x_ /; x > 0.];
   xiiS = Append[xiiS, sel];
   xiiT =
    Append[xiiT, Sum[sel[[k]], {k, 1, Length[sel]}] / Length[xii[[s, All, 1]]]];
   outfile2 = "xii." <> divStr <> "." <> "cut" <> ToString[minISI] <>
     "." <> "s" <> ToString[s] <> "." <> "eps";
   xiipl = Append[xiipl, ListPlot[xii[[s, All]], ps]];
   (*Export[outfile2,ListPlot[xii2[[s,ch,All]],
     PlotRange→{{0.,Max[data[[ch]]]+0.01},{0.,Max[data[[ch]]]+0.01}}]]*),
   {s, 1, smax}];
  xii2 = Append[xii2, xii]; xii2S = Append[xii2S, xiiS];
  xii2T = Append[xii2T, xiiT]; xiipl2 = Append[xiipl2, xiipl];]

In[42]:= splots := Module[{s1},
  Print[s1 = ListPlot[xii2T[[ch]], Joined → True, PlotRange → {0, Max[xii2T[[ch]]]}]];
 ]

In[161]:= data = {}; delta0 = {}; delta1 = {}; delta2 = {}; spcor = {}; div1c = {};
jk = {}; div1d = {}; new = {}; xii2 = {}; xii2S = {}; xii2T = {}; xiipl2 = {};
If[nomin > 1, data = Table[{}, {i, nomin - 1}]; div1c = data; div1d = data; jk = data;
  new = data; xii2 = xii2S = xii2T = xiipl2 = delta0 = delta1 = delta2 = data;];
Do[
  appChannel = ToString[div[[1, 1, no]]];
  appDIV = ToString[div]; div1 = Append[Drop[div[[1, All, no]], 1], 0.];
  n = 1; While[div1[[n + 1]] >= div1[[n]] && div1[[n + 1]] ≤ tmax, n++];
  nmax = Min[{n, ncut}];
  div1c = Append[div1c, Take[div1, {nmin, nmax}]];
  dataB;
  If[spar == 1 , sxii];
  If[normIJout == 1 , emb; normM; corrIntM];
  If[recplpar == 1 , recplM],
  {no, nomin, nomax}];

In[165]:= ext2 := "." <> ToString[i] <> "." <> ToString[j] <> "." <> divStr <> ".dat"

In[166]:= outcorrpar = 0; outfreqpar = 0; outFpar = 0;

In[167]:= DT = div1c;

In[168]:= FreqM := Module[{i, k, j, m0, no},
    f0 = Table[0, {j, 1, nomax}, {k, 1, NBins}];
    mean0 = Table["", {l, 1, nomax}];
    Normf0 = Table[0, {l, 1, nomax}]; delta0 = Table["", {j, 1, nomax}, {k, 1, NBins}];
    Do[nmax = Length[DT[[no, All]]]; i = nmin;
     Do[
      While[
        DT[[no, i]] < m0 * deltaT && i ≤ (nmax - nmin),
        f0[[no, m0]] = f0[[no, m0]] + 1; i = i + 1];
      , {m0, 1, NBins}];
     mean0[[no]] = Sum[f0[[no, k]], {k, 1, NBins}] / NBins;
     Normf0[[no]] = Sqrt[Sum[(f0[[no, j]] - mean0[[no]])^2, {j, 1, NBins}]];
     Do[delta0[[no, m1]] = N[(f0[[no, m1]] - mean0[[no]]) / Normf0[[no]]],
      {m1, 1, NBins}], {no, 1, nomax}]];
```

```
In[169]:= CorrM := Module[{v, i, j, k}, Cut = NBins;
             CorrFunct = Table[{i * T / NBins,
                (NBins / (Cut - i)) * N[Sum[delta0[[j, v]] * delta0[[k, v + i]], {v, 1, Cut - i}]]},
               {i, taumin, taumax}, {j, 1, nomax - nomin + 1}, {k, 1, nomax - nomin + 1}]
            ];

In[170]:= FreqM;

In[171]:= ch1 = 43; ch2 = 26;

In[172]:= Fplot = ListPlot[{delta0[[ch1, All]], delta0[[ch2, All]]}, PlotRange → Full,
             Mesh → Full, Joined → True, PlotRange → Full, PlotStyle → {Thick},
             PlotLabel → Style["F(" <> ToString[ch1] <> "), F(" <> ToString[ch2] <> ")", Large],
             AxesLabel → {"i", None},
             Ticks → {{0, NBins / 4, NBins / 2, 3 * NBins / 4, NBins}, {-0.1, 0, 0.1, 0.2, 0.3}},
             LabelStyle → Directive[Black, Large],
             AxesOrigin → {0, Min[{delta0[[ch1, All]], delta0[[ch2, All]]}]}];

In[174]:= CorrM;

          Ndelta0 = N[delta0];

In[177]:= ListPlot[{CorrFunct[[All, 34, 34]], CorrFunct[[All, 1, 2]]},
            PlotRange → {-0.4, 1.4}, Joined → True];

In[178]:= outa1par = 0; outc2par = 0;

In[180]:= Integrate2M := Module[{},
            a = Table[0, {l, nomin, nomax}, {k, nomin, nomax}];
            Do[
             p = 1;
             While[DT2[[p, l, k, 1]] < bord, a[[l, k]] =
               a[[l, k]] + deltaT * (DT2[[p, l, k, 2]])^2; p++]; a[[l, k]] = Sqrt[a[[l, k]]]
             , {l, nomin, nomax}, {k, nomin, nomax}];
           ]

In[182]:= DT2 = N[CorrFunct];

In[183]:= Integrate2M

In[184]:= ch01 = 1; ch02 = 2; ch03 = 23; ch04 = 52;

In[185]:= corrpl = ListPlot[{DT2[[All, ch01, ch01, All]], DT2[[All, ch01, ch02, All]],
              DT2[[All, ch03, ch04, All]]}, Joined → True, PlotRange → Full,
             PlotStyle → {Thick}, PlotLabel → Style["C(" <> ToString[ch01] <> "," <>
                ToString[ch01] <> "), C(" <> ToString[ch01] <> "," <> ToString[ch02] <>
                "), C(" <> ToString[ch03] <> "," <> ToString[ch04] <> ")", Large],
             AxesLabel → {"t", None}, Ticks → {{0, 0.1, 0.2, 0.3, 0.4, 0.5}, {0.5, 1}},
             LabelStyle → Directive[Black, Large]
            ];

In[187]:= ArrayPlot[a, ColorFunction → asf, ColorFunctionScaling → True];

In[256]:= maxTD = Which[d == 5, 20, d == 7, 10, d == 12, 5];

In[191]:= outa2par = 0; trafterpar = 0; outnewpar = 0; discoutpar = 0;
          discimppar = 0; spectpar = 0; spectimppar = 0; abetaout = 0;
```

```
In[257]:= cutCh = Which[d == 5, {4}, d == 7, {4}, d == 12, {4}];

In[193]:= DT3 = N[a, 20];

In[195]:= ArrayPlot[DT3, ColorFunction → asf, ColorFunctionScaling → True];

In[196]:= Do[DT3 = Transpose[Drop[Transpose[Drop[DT3, {cutCh[[l]]}]], {cutCh[[l]]}]];,
           {l, 1, Length[cutCh]}]

In[258]:= nomax = Length[DT3];

In[259]:= bmin = Which[d == 5, -40, d == 7, -40, d == 12, -40];

In[260]:= bmax = Which[d == 5, 40, d == 7, 40, d == 12, 40];

In[261]:= bstep = Which[d == 5, 0.1, d == 7, 0.1, d == 12, 0.1];

In[202]:= ArrayPlot[DT3, ColorFunction → asf, ColorFunctionScaling → True];

In[203]:= truncpl = DT3;
          Do[If[DT3[[i, j]] == 0, DT3[[i, j]] = Black], {i, 1, nomax}, {j, 1, nomax}]

In[205]:= a0pl = ArrayPlot[truncpl, ColorFunction → asf, ColorFunctionScaling → True];

In[206]:= beta = Table[j, {j, bmin, bmax, bstep}]; bL = Length[beta];

In[207]:= DT300 = Table[DT3[[k, p]]^beta[[b]], {k, 1, nomax}, {p, 1, nomax}, {b, 1, bL}];

In[209]:= norma00 = Table[Sum[DT300[[k, p, b]], {p, 1, nomax}], {k, 1, nomax}, {b, 1, bL}];

In[210]:= aaaa00 = Table[If[norma00[[k, b]] == 0, 0, DT300[[k, p, b]] / norma00[[k, b]]],
             {k, 1, nomax}, {p, 1, nomax}, {b, 1, bL}];

In[263]:= bbsp = (bL + 1) / 2;

In[215]:= norma = Table[N[Sum[DT3[[k, p]], {p, 1, nomax}]], {k, 1, nomax}];

In[216]:= aaaa =
            Table[If[norma[[k]] == 0, 0, DT3[[k, p]] / norma[[k]]], {k, 1, nomax}, {p, 1, nomax}];

In[217]:= apl = ArrayPlot[aaaa, ColorFunction → asf, ColorFunctionScaling → True];

In[218]:= abetapl = Table[ArrayPlot[aaaa00[[All, All, actb]], ColorFunction → asf,
              ColorFunctionScaling → True], {actb, 1, bL, (bL - 1) / 8}];

In[264]:= abL = Length[abetapl];

In[221]:= abetapl;

In[265]:= Det[aaaa00[[All, All, bbsp]]];

In[266]:= Det[aaaa];

In[267]:= Tr[aaaa00[[All, All, bbsp]]];

In[268]:= Tr[aaaa];

In[228]:= tra = Table[Tr[aaaa00[[All, All, b]]], {b, 1, bL}];

In[269]:= parE = Which[d == 5, 3, d == 7, 5, d == 12, 1];

In[270]:= parS = Which[d == 5, 15, d == 7, 10, d == 12, 5];
```

```
In[271]:= maxp = Which[d == 5, 1, d == 7, 4, d == 12, 5];

In[272]:= minp = Which[d == 5, -1, d == 7, -4, d == 12, -5];

In[235]:= logZ = Table[{beta[[b]], Log[tra[[b]]]}, {b, 1, bL}];

In[236]:= Energy = Table[
            {beta[[b]], -parE * (logZ[[b - 1, 2]] - logZ[[b + 1, 2]]) / (2 * bstep)}, {b, 2, bL - 1}];

In[237]:= EnergyOrg =
           Table[{beta[[b]], - (logZ[[b - 1, 2]] - logZ[[b + 1, 2]]) / (2 * bstep)}, {b, 2, bL - 1}];

In[238]:= susc = Table[{beta[[b + 1]],
             parS * (Energy[[b - 1, 2]] - Energy[[b + 1, 2]]) / (2 * bstep)}, {b, 2, bL - 3}];

In[239]:= suscOrg = Table[
            {beta[[b + 1]], (Energy[[b - 1, 2]] - Energy[[b + 1, 2]]) / (2 * bstep)}, {b, 2, bL - 3}];

In[247]:= trapl00 = ListPlot[{logZ, Energy, susc}, PlotRange → {{bmin, bmax}, {minp, maxp}},
                                                          d Log[Z]
            PlotLabel → Style["Log[Z],    " <> ToString[parE] <> "* ────────,    " <>
                                                             dβ
                                       -d² Log[Z]
               ToString[parS] <> "* ──────────"], PlotStyle → {Thick},
                                          dβ²
            AxesLabel → {"β", None}, Joined → True, LabelStyle → Directive[Black, Large]];

In[248]:= ListPlot[suscOrg, PlotRange → All];

In[255]:= ListPlot[EnergyOrg];
```

Bibliography

[1] A. Tscherter, M. O. Heuschkel, P. Renaud, J. Streit, Spatiotemporal characterization of rhythmic activity in rat spinal cord slice cultures, Eur. J. Neuroscience 14, 179-190 (2001)

[2] G. Buzsaki, Large-scale recording of neuronal ensembles, Nature Neuroscience 7, 446-451 (2004)

[3] D. A. Wagenaar, R. Madhavan, J. Pine, S. M. Potter, Controlling bursting in cortical cultures with closed-loop multi-electrode stimulation, J. Neuroscience 25, 680-688 (2005)

[4] D. A. Wagenaar, J. Pine, S. M. Potter, An extremely rich repertoire of bursting patterns during the development of cortical cultures, BMC Neuroscience 7, 11 (2006)

[5] M. Hutzler, A. Lambacher, B. Eversmann, M. Jenker, R. Thews, P. Fromherz, High-Resolution Multitransistor Array Recording of Electrical Field Potentials in Cultured Brain Slices, J. Neurophysiology 96, 1638-1645 (2006)

[6] A. E. Oshiorenoya, S. Weigel, R. Stoop, A. Bruinink, Fused tetanus is created by modulation of spinal circuits by higher motor centres (to be published)

[7] G. Santhanam, S. I. Ryu, B. M. Yu, A. Afshar, K. V. Shenoy, A high-performance brain-computer interface, Nature 442, 195-198 (2006)

[8] http://www.ninds.nih.gov/research/epilepsyweb/curingepilepsy/Rickus.pdf

[9] http://en.wikibooks.org/wiki/Anatomy_and_Physiology_of_Animals/Nervous_System

[10] http://www.shelfieldpeonline.co.uk/html/motor_units_and_motor_neurone_.html

[11] http://faculty.etsu.edu/currie/somatic.htm

[12] http://www.empa.ch

[13] http://www.admiroutes.asso.fr/larevue/2008/90/gordon.htm

[14] for more information on bursting see e.g. http://www.scholarpedia.org/article/Bursting

[15] H. Kantz, T. Schreiber, Nonlinear Time Series Analysis (Cambridge University Press, Cambridge UK, 2000)

[16] J. Peinke, J. Parisi, O. E. Roessler, R. Stoop, Encounter with Chaos (Springer, Berlin, 1992)

[17] P. Grassberger, I. Procaccia, Characterization of Strange Attractors, Phys. Rev. Lett. 50, 346-349 (1983)

[18] A. Kern, W.-H. Steeb, R. Stoop, Local Correlation's Potential for Noise Reduction and Symbolic Partitions, Z. Naturforsch. 54a, 404 (1999)

[19] M. Christen, A. Kern, A. Nikitchenko, W.-H. Steeb, R. Stoop, Fast spike pattern detection using the correlation integral, Phys. Rev. E 70, 011901 (2004).

[20] M. Christen, A. Nikitchenko, W.-H. Steeb, R. Stoop, Combinatorial laws underlying log-log correlation integral steps generation (to be published)

[21] H. Whitney, Differentiable manifolds, Ann. Math. 37, 645-680 (1936)

[22] F. Takens, Detecting strange attractors in turbulence, in: *Dynamical Systems and Turbulence*, Lecture Notes in Mathematics 898, eds. D. A. Rand and L. S. Young (Springer, Berlin, 1981) pp. 366-381

[23] T. Sauer, Embedology, J. Stat. Phys. 65, 579-616 (1991)

[24] J.-P. Eckmann, S. O. Kamphorst, D. Ruelle, Recurrence plots of dynamical systems, Europhys. Lett. 5 (1987)

[25] N. Marwan, M. C. Romano, M. Thiel, J. Kurths, Recurrence plots for the analysis of complex systems, Phys. Rep. 438, 237-329 (2007)

[26] H. Poincaré, Sur le probleme des trois corps et les équations de la dynamique, Acta Mathematica 13, 1-271 (1890)

[27] L. E. Reichl, A Modern Course in Statistical Physics (Wiley, Weinheim, 1998)

[28] H. E. Stanley, Introduction to Phase Transitions and Critical Phenomena (Oxford University Press, New York, 1971)

[29] see for example http://www.scholarpedia.org/article/Neuronal_avalanche