LMU Munich
Faculty of Physics

Master's Thesis

Machine Learning based Reproduction of Thermoacoustic Oscillations

Marcus Andreas Winkler

Supervised by
PD Dr. Christoph Räth

September 1, 2023

LMU München
Fakultät für Physik

Masterarbeit

Machine-Learning basierte Reproduktion von thermoakustischen Oszillationen

Marcus Andreas Winkler

betreut und begutachtet von
PD Dr. Christoph Räth

1. September 2023

Contents

Abstract

This thesis is a comprehensive discourse on a machine learning application to real data that tackles data analysis and hyperparameter tuning using reservoir computing (RC). RC is known for its stable and fast predictions of nonlinear and chaotic dynamical systems. Here, it is discussed which properties of the data can be adopted by the reservoir, which measures are appropriate to assess the prediction quality, and which RC setup is most suitable to do so. The used data set contains pressure signals from an experimental rocket thrust chamber, which shows the critical thermoacoustic effect of combustion instabilities. By focusing on the reproduction of linear and nonlinear dynamics, the aim is to detect early shifts by constructing a digital twin of the system dynamics. Recurrence quantification analysis (RQA), statistics on amplitude distributions and frequencies, and short-term prediction capability are used as measures to investigate the system properties. It is shown that reservoir computing can reproduce different dynamical states of the system, in both linear and nonlinear properties. It turned out that comparatively high values for the spectral radius (3.0) and the regression parameter (100) are reliable choices in order to achieve good results. A sliding window technique is used to classify rapid changes in the dynamical characteristics of the system evolution. This technique could serve as a new adaptive monitoring method for control systems.

Introduction

At this moment, the era of artificial intelligence and big data, we are witnessing the world being mapped into the space of numbers and data at an ever-increasing rate, and algorithms are growing to be the go-to tool for answers in our daily lives. This development is also evident in the natural sciences. After the development of successful mathematical formulations for observations of nature, in the form of axioms, theorems and equations, a complementary methodology is now emerging, which is almost entirely data-based through self-learning software. Its results, as well as the analysis of itself, gives further natural-philosophical insights as well as new technical possibilities. This thesis will follow this new path, and uses on purpose a completely data driven approach. While hybrid methods, where machine learning tools get enhanced with existing knowledge about the theories behind the observed data, getting indeed good achievements, they will not be used. The motivation is to really develop universal strategies to get from a point of pure numbers, which are gained through digitized information from the real world, towards an understanding and hence prediction capability. So are the attention and the effort transferred away from the formulation of the problem, towards the conceptual design of the method and the pure structure of the data.

As a representative of machine learning, this thesis studies reservoir computing. It is a popularity gaining model type, which will be applied to a data set, obtained from real experiment. This will take the comfort of selected generated data and will take further steps towards the proof of applicability. The subject of interest are thermoacoustic oscillations, which are the theory behind crucial processes of the combustion process in rocket engines. Seemingly spontaneously occurring dynamical instabilities, caused by nonlinear processes, are a major harm to the industry and development of rockets[1].

The goal of this work is to reproduce the dynamic behavior through the standalone observed pressure oscillations, to pave the way for further possible steps to deal with the instabilities. Thereby it should not only be tested the strength of the machine learning method, but it should also reveal more information about the reservoir computing itself. Overall, to get an insight into the dynamical state and to supervise the performance, also measuring methods are discussed and tested in greater detail, as they are itself important tools in the interplay of data, model and supervision. In the spirit of the digital twin [2], future control mechanisms may be developed in connection with the AI-generated data [3], which will support the development of future rocket engine models.

State of the Art The 2001 invented method of reservoir computing by Jaeger [4] and Maass [5], has already proven itself in the most diverse fields of application, like speech recognition [6], computer vision [7] and climate prediction [8].

An already existing work where reservoir computing, especially ESNs, are applied to the subject of thermoacoustics, was done by Huhn in 2021 [9, 10]. As well pure ESNs, as also physics informed, so called hybrid models (hESNs) where used, and showed good results. The conventional ESN had especially good long term predictions, in terms of

[1] A great historical introduction to the subject can be found in the introduction to the work of Praveen Kasthuri [1]

attractor reproduction, where the hESN did both, short- and long term prediction in a high quality and on low computational cost. In contrast to our work, where only small reservoirs used, of the size of 400 and 100 nodes, respectively. The biggest difference, however, is that the data used as ground truth did not come from a real experiment, but was generated by using a differential equations based model.

A just recently published work by Kong et al. in 2023 [11], also used the terminology of digital twins. The authors aimed to develop autonomous-driven dynamical evolving systems that mimic the same properties as their real counterparts. They also used ESNs for this, as they are a suitable choice for this type of task. This extensive study contains various examples, such as optics, ecology, climate and a chaotic CO_2 laser system. To achieve forecasting and monitoring capabilities, sparse real-time updates were built-in, to capture environmental changes. But this work also only used simulated model data, generated by differential equations. Their results, though, even further motivate this thesis to be on the right track.

An inspiring work for us, is the work of Waxenegger-Wilfing et al. from 2021 [12]. Despite not using reservoir computing, it is a data-driven approach applied to the very same experimental data set as it is used in this thesis. It uses the approach of calculating RQA Measures (RR, DET, LAM, ENTR and RATIO) and there linear trends, all derived from sequential sampling windows of interval $[t - 200\text{ms}, t]$ and $[t - 100\text{ms}, t]$, respectively. Totaling 10 derived combustion noise features. In a second step an pretrained support vector machine (SVM) was used as a binary classifier to decide on those nonlinear dynamical representations, if there is a need for an early warning or not. The SVM was trained on data from several runs of the same experimental thrust chamber. In this paper a performance of a True-Positive-Rate up to 80 % for early warnings was reached, by a False-Positive-Rate of less than 5 %. This holds for a used forecasting time of 200 ms. Remarkable is that the detection also kind of worked for using test data of a different experimental setup, then used for the training data. It was also shown that the most significant features for the SVM were RR, ENTR and DET as seen in Figure 7 of [12]. Table III in [12] also shows the performance comparison of just using one of the mentioned RQA features. The main difference to this thesis, is that it used a detection mechanism for precursors of instability on the data, instead of using the digital twin approach, to imitate the whole dynamical process. The latter could make it possible to derive control mechanisms from the digital copy of the system [3].

Content Structure This thesis splits into three main parts. It starts with examining the data. Background information about the thermoacoustic oscillation is given, and numerical properties of the data set are discussed, which are important to understand the challenges for the machine learning part, as well as they are also important for the physical interpretation of the results in this work. The next chapter lays the theoretical groundwork for the methods tested in this thesis. This includes data preparation based on Takens' theorem, as well as introduction and implementation related details for the machine learning approach of reservoir computing. Additionally, the measures for the data analysis are introduced, which are also used to analyze the reproduced data by reservoir computing, and eventually, to evaluate the overall performance. The third part presents the results. Those are findings regarding the linear properties and phase space dimensionality of the data, as also examples for the predictions and general outcome of

the reservoir training. Furthermore, hyperparameter grid search dependencies on the reservoir are analyzed, to get a further understanding for the data related dependencies. Finally, a sliding window evolution for reservoir predictions alongside the actual data is applied, to get information about the behavior on state transitions.

Style Remarks

For data arrays in this thesis, the Einstein notation is used. This means that two equal indices mean, that a multiplication and summation like

$$C^{\nu}{}_{\mu} = B^{\nu}{}_{t} A^{t}{}_{\mu} := \sum_{t} A^{\nu}{}_{t} \cdot B^{t}{}_{\mu}$$

is initiated. Each element in this formulation, like $A^{t}{}_{\mu}$ is then just a number, i.e. an entry in the array. This notation, common in physics, is also intended here to be a support for clarity and to avoid ambiguities for an explicit software implementation, as can be the case with indirect matrix notation C=BA. In addition, use is made of the concept of a *design matrix*. A design matrix, as an often used convention within the field of machine learning [13], is an arrangement of samples i with certain features μ, not as a set of vectors

$$\left\{(x_1, ..., x_\mu, ..., x_{\mu_{\max}})^T_{(1)}, ..., (x_1, ..., x_\mu, ..., x_{\mu_{\max}})^T_{(i)}, ..., (x_1, ..., x_\mu, ..., x_{\mu_{\max}})^T_{(i_{\max})}\right\}$$

for all data points x, but already as a single matrix X, with access to its entries by $X^{i}{}_{\mu}$. This concept holds as well for the input data X, as it does for the target data Y in the same order. The advantage of this concept is the simpler realization of analytical solutions, such as those used in regression analysis. If one indice of an two-dimensional object isn't further specified, like for the object above $A^{t}{}_{\mu}$, it means that vectors of the form

$$A^t := (A^t{}_0, ..., A^t{}_{\mu_{max}})$$
$$A_\mu := (A^0{}_\mu, ..., A^{t_{max}}{}_\mu)^T$$

are provided.

Moreover, the whole thesis makes just use of "natural units" for the data description. Since the work focuses on the machine learning part and results should be directly comparable with other methods and data sets, we use these digital units, which are the only ones that are relevant for the computational machine. That means that instead of using seconds for the time, the unit *time steps* is used. Frequencies are also measured in oscillations per time step rather than in Hertz. The further treatment of the subject, and the rules for the translation between units can be found in section 2.2.

Combustion Data

In this part, we will concentrate on the data, and the physics behind them. Even if the used machine learning method is not knowledge-based, information about the data will help to analyze the results, and will help to make decisions for future design choices. In the first section, the physical phenomenon of thermoacoustic oscillations and combustion instabilities, both terms referring to the same subject, will be introduced and a physical basic understanding should be conveyed. Since it is an important topic for liquid rocket engines, solid rocket motors, tactical and strategic missiles, gas turbine engines, power-producing gas turbines, industrial boilers and many more [1], and it was already known in the 19th century [14], there is a lot of research and literature available, where this should be summarizing overview. The second section deals with the actual data set, which is used later on for supervised machine learning. It is an impression of the experimental setup, as it provides the connection to machine learning in the following chapters.

2.1 Combustion Instabilities

The energy for the combustion process, and hence for the power of the rocket originates of course through chemical reactions, induced by shower head like, high pressure fuel injections and ignition. This results in an highly powerful and very complex dynamical system, that is often described by the physics of thermoacoustic oscillations [15]. It is partly linear and partly nonlinear, driven by the interaction between acoustics, heat source and turbulent hydrodynamics [9, 15]. To model such a system, it requires at a first step the Euler equations, for the mass flow field, and thermodynamical state functions, like the enthalpie [9, 15]. The principle of energy release in the thermoacoustic system within the combustion chamber has a nice explanation in [15]. It is compared to an combustion engine as one could find it in a car. Where the piston equivalent to an acoustic wave in the rocket engine compresses in a periodic process the gas, which then ignites and expands with an energy released greater than the initial work, which was used for the compression. If this energy overhead does not dissipate from the local acoustic and temperature field, then the acoustic pressure amplitude increases. In this process, effects like the combustion instabilities can arise, which leads to the machine learning task in this thesis. As written in [1], "Thermoacoustic instability is characterized by large harmful oscillations in pressure and heat release rate arising due to the positive feedback between the acoustic pressure oscillations in a confinement and the heat release rate oscillations in the flame." Regarding the confinement, the dynamical system has also a highly spatial geometric significance. Solutions to the linear part of the problem, given by the linear acoustic equation

$$0 = \frac{\partial^2 (p - p_0)}{\partial t^2} - c^2 \Delta (p - p_0)$$

with pressure fluctuations $p - p_0$, the speed of sound c, and the Laplace operator Δ, provides depending on the rocket chamber boundaries corresponding modes [16]. Thermoacoustic oscillations, which embody the combustion instability, show a strong linear

behavior and periodicity [15]. With this linearity, they represent a large energy storage in the mentioned modes, with geometric characteristic frequencies. This is the foundation for the signal, which will be used in this thesis. Since thermoacoustic oscillations are, as the name says, not only acoustic but also of thermodynamic nature, there also exists the approach to take the observational access to the system dynamics by coupling quantities, which originate from the thermodynamical part of the theoretical model. For this, measurements in the ultraviolet and visible spectrum of electromagnetic fields are undertaken [17], to access the second important dynamical coupling partner [14], the heat release rate. This was also done for non-premixed hydrogen-oxygen flames, with a focus on combustion instabilities like in liquid rocket engines [17]. The critical quantity for the appearance of instabilities ist the delay time for the heat release and acoustics. Since this is bound to the engine geometry, design can already have a strong influence. This allows on the other hand a good starting point for preventing critical situations [15], where digital twin models could be good support. In the following section, the dataset and its origin will be examined.

2.2 Origin and Properties of the Data Set

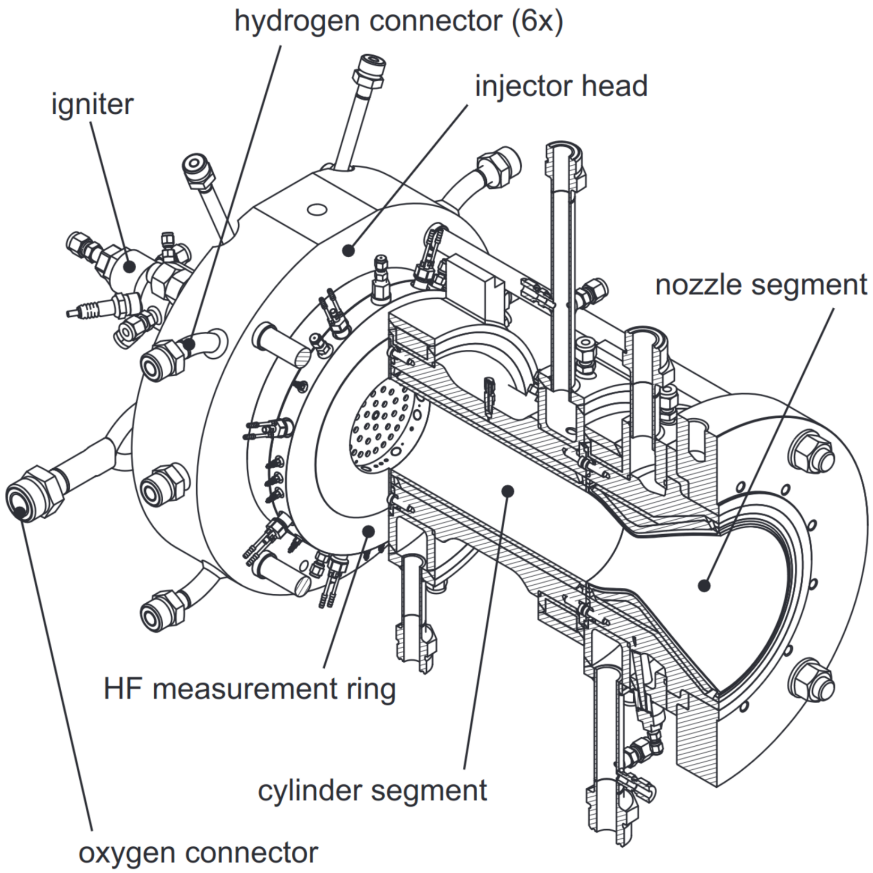


Figure 1: *Experimental "BKD" thrust chamber setup. The illustration originates from [18].*

The experimental data for this work originates from experiments at the DLR Insti-

tute of Space Propulsion in Lampoldshausen, from a recent research thrust chamber, which is referenced in the literature as "BKD" [19]. It is a cryogenic LOX/H2 multi-injector installation, on which combustion instabilities, under realistic conditions, are tested. The typical physical circumstances, within this 20 cm long, and 8 cm wide chamber, are 3600 K, 80 bar and a thermal power up to 100 MW [16, 12]. Through a 5 cm wide nozzle throat, with a fuel consumption of 6.7 kgs^{-1}, it reaches a thrust of about 24 kN. The *thermoacoustic oscillations*, also called *combustion instabilities*,

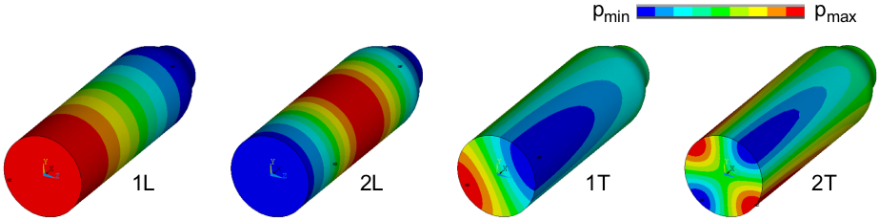


Figure 2: *Acoustic mode shapes for the "BKD" combustion chamber, calculated using modal analysis. The original figure can be found in [16].*

which are of interest for this work, do appear in the shape of different modes. In figure 2, one can see the solutions for the acoustic wave equation within the volume of the experimental combustion chamber. Each of those longitudinal (L) and transverse (T) modes, and their combinations, resulting in different pressure frequencies, resulting in a characteristic frequency spectrum for the geometry. Dominant frequencies are for example about 10.5 kHz for the T1 mode, 12.5 kHz for 1T1L and about 17 kHz for the T2 mode [18].

Eight high frequency pressure sensors, working with a membrane-piezoelectric combination, arranged around the test chamber as it is shown in figure 1, measured the acoustic evolution for several runs. Each run is done on different settings for the fuel, and contains millions of data points for just a few seconds. A total of 16 instabilities appeared during the experiments. Those events were distinguished into type 1 instability, with 6.25 % peak-to-peak amplitude and type 2 with a threshold of 20.0 % peak-to-peak amplitude, with respect to the mean chamber pressure [12].

The relative pressure oscillation around 0, as shown in figure 3 for the 8 sensors, is the sole basis for the dynamical reproduction done in this thesis.

Numerical Properties

Now, the viewpoint will change from the technical aspect towards a pure data scientific perspective. As we are using just one pressure sensor, at least outside the appendix, the raw data set has one spatial dimension, as one would say for dynamical systems, or in the terms of machine learning, it has one feature dimension.

Since the data is not generated, but from a real measurement, in contrast to the work of [9], and also comes from a complex dynamic system, as mentioned above, the data comes with the following challenges:

- noise (see section 4.1.1)
- exogenous variables, hence non-autonomous (like propellant mixture rate, propellant injection temperature, chamber pressure [12])

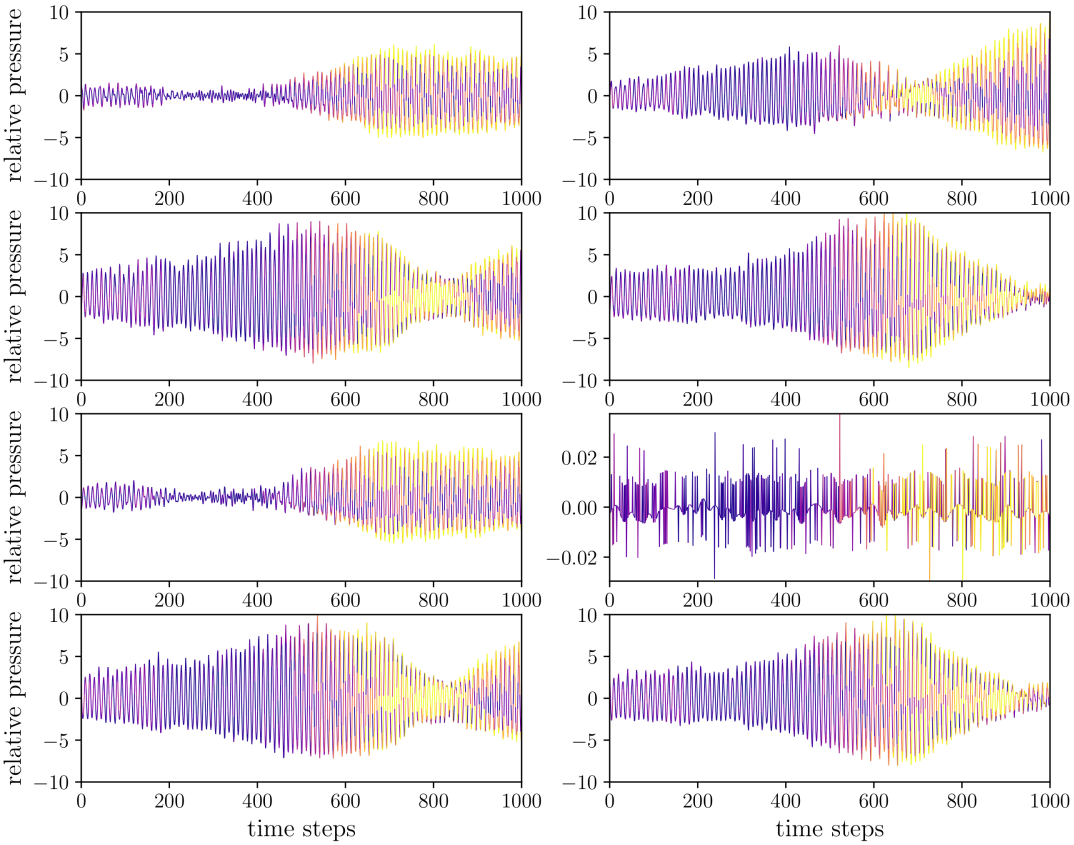


Figure 3: *This is a 10 ms temporary snippet out of the data from figure 4, where all eight pressure sensors are shown in parallel. Since, this picture has a more spread time axis, one can see that the dynamical shape of the signal shows a different stochastic behavior than in figure 4. As it is in the nature of physics, one has different governing dynamical properties at different scales, like in this case, different time scales. Furthermore, it seems like the sixth sensor had an operational error. Also belonging all signals to a different phase, which could be explained due to their different positions within the combustion chamber. The pressure sensors are numbered from left to right, top to bottom, and 0 to 7.*

- low temporal resolution (see below)
- nonlinearity, maybe chaotic behavior [1, 15]
- different numerical magnitudes (the mean long term peak-to-peak amplitude varies by a factor of 100, see figure 4)
- different dynamical features on different time scales (comparison of figure 3 and 4)

- systematic measurement error (like through the boiling coolant around the sensors [16])

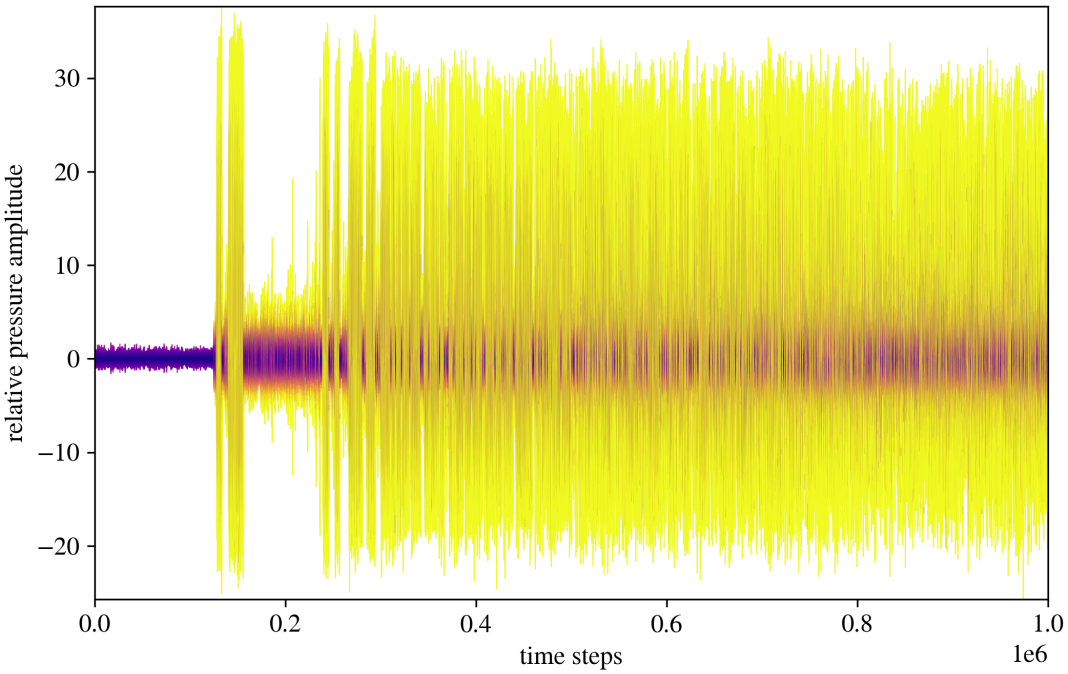


Figure 4: *This one million time steps and 10 seconds long data series is in itself part of one experimental run of the combustion chamber, and is mainly used during this thesis. It represents the values of one pressure sensor, which measured the relative pressure oscillation. As one can see by the difference in amplitude, it contains a transition from a stable target state to an thermoacoustic state, which is assigned as an instabillity of type 2.*

Those properties lead to an inhomogeneous data structure over time. Due to this it arises the situation, that despite there are a lot of data points available, the data batches for the time series learning are still limited, in the sense that for too long samples different features are captured. That is an essential difference to generated data. On the other hand, within those millions of time steps, there are just a few ten interesting global events, which means that there are just a few examples to learn from and validate the results. This is also the reason why in this thesis the machine learning is applied such that it distinguishes regions. all within the time series shown in figure 4, are selected to reproduce the exact dynamical state, instead of supervised learning on the 16 existing instabilities.

The physical signal is measured with a sampling rate of 100 kHz [16]. To keep the results comparable to other machine learning algorithms, and to focus on the characteristics of reservoir computing, all units in this work are given as "natural digital" units. This means in units of *time steps*, which are just the enumerated samples in the

data arrays. To switch between the units, the conversion rule

$$\boxed{1 \text{ time step} \mathrel{\widehat{=}} 10\ \mu\text{s}} \tag{1}$$

applies. For the sake of convenience, since the most plots are in this range, 1000 time steps equals 10 ms. For frequencies, one has to use

$$\boxed{0.1 \text{ periods per time step} \mathrel{\widehat{=}} 10 \text{ kHz}} \tag{2}$$

as it is found at the frequency-axis in every plot for this thesis. That means that 1 sine-shaped period of a signal, takes 10 time steps, or in other words, has a resolution of 10 data points. Which can also be understood as the *characteristic time scale* for the combustion data, since it has at the first glance a periodicity of 10 time steps, which will be understood in greater detail as part of the analysis in section 4.1. Also for convenience, (2) means that 0.2 periods per time step, means 20 kHz and 0.01 means 1 kHz.

Now that the data is introduced and motivated, the next chapter begins with the theoretical part for the methods that will be applied to the data.

Computational Methods

After we have examined the data, we will build the background for the tools which will be used, in the aspects of theory, motivation and implementation. The first part of this chapter will deal with Takens' theorem, as the method to increase the usability of the information content contained in the data. Afterwards, the heart of this thesis, the reservoir computing machine learning approach will be discussed. Finally, as the third big topic, the methods to measure the performance of the interplay between the data and the machine learning are introduced.

3.1 Data Preprocessing

In order to reduce the requirements for the pure machine learning model, the data is processed before it is handed over as input. To unfold the system dynamics, we can use the well known embedding methods based on Takens' Theorem. Because Takens' Theorem tells that it *is* possible to unfold the dynamics from a just one dimensional observation, but not *how*, the false nearest neighbors algorithm (FNN) and the auto-correlation function (ACF) came into play, for the final practical application of Takens' Theorem. All of this will be discussed in the following three sections.

3.1.1 Takens' Theorem

The gain of information out of measured data is always a topic of high interest. Here it is done with Takens' delay embedding theorem, which is a geometric based approach for it. This section begins with a brief mathematical guide to the basic concepts, before the theorems original appearance is discussed, to eventually show how it is explicitly applied in this work. First of all, an *embedding* is a mapping of a mathematical structure into another. For differentiable structures, as encountered in the physical world, an embedding

$$\phi : M \longrightarrow f(M) \subseteq N \tag{3}$$

of one differentiable manifold M in to another N, where $\dim(M) \leq \dim(N)$, is given if ϕ is smooth, injective and also

$$D\phi : TM \longrightarrow T\phi(M) \subseteq TM \tag{4}$$

is injective. This means that as well the individual points are mapped one-to-one as well as the derivatives on them. To read more on this, [20] is a good place for. At first, it should be started with the famous whitney embedding theorem from [21], that states that any smooth manifold M of dimension $\dim(M)$ can be embedded into a cartesian space $\mathbb{R}^m$ if

$$m > 2\dim(M) + 1 \tag{5}$$

holds and even $m > 2\dim(\mathrm{M})$ is sufficient if the manifold is of dimension $\dim(M) \geq 2$. But the embedding by Takens' Theorem is more special, since it does not just carry the

information of the given structure into another structure, it also tries to reconstruct information of a hidden structure, where the used data is just a part of. In the following, it is provided a sketch of the content of [22] to gain a bit of a historical connection with the theorem, on which a huge field of applications is based nowadays. Motivated by the ongoing research on turbulences, it was proven in the publication with the language of differential geometry, that for a (smooth) dynamical system, given one observational scalar function y and a certain flow φ_t for the systems time evolution according to a vector field X, that there is an $2m+1$ dimensional embedding

$$\Phi_{X,y}(x) = (y(x), y(\varphi_1(x)), ..., y(\varphi_{2m}(x)))$$

where x is a system state. As he has written, Takens aimed in his work for a method, which helps to decide whether experimental data attributes to the presence of a strange attractor. Experimental time series data has in this setup the theoretical form of $t \mapsto y(\varphi_t(x))$. It is further taken into consideration that a measured time series of a dynamical system of continuous time t does have just a countable structure. This means, it is a sequence, indexed by i, which is maybe only measured on one point x and for finite but many time steps $i = 1, ..., \bar{N}$. The measuring is taken in time steps of some delay $\alpha > 0$.[2] Finally realistic measured data is then a set of the from

$$\{y\left(\varphi_{i\alpha}(x)\right)\}_{i=0}^{\bar{N}} \ . \tag{6}$$

With this taken into account it was shown in the main theorem of the paper, how to calculate the limit capacity and the topological entropy for such a data set. And as intended, this result allows us to determine whether given data is attached to a strange attractor or not. For the complete theoretical model and the explicit conditions for the results, see [22]. Additionally to that it should be mentioned the publication of Sauer, Yorke and Casdagli [23] published in 1991. This work generalizes the conditions for an embedding, stated by Takens or Whitney. An important point which is generalized in [23], is that the criterion of (5) is generalized to the box-counting dimension. That is interesting, since it is a fractal dimension like the correlation dimension, which will be introduced in section 3.3.3.

To bridge the gap between pure mathematics towards real world modeling, the term *phase space* is very useful and is therefore introduced now. The phase space, the continuous case of a state space, is the space of all possible states of a system and will take on the role of the smooth manifold M from above. The *degrees of freedom*, referring to the dimensions[3] of the phase space, is the in time evolving minimal set of information of the system. "Minimal" refers to the fact that one can always build up an arbitrary number of new describing properties for the system out of the existing ones. On the other hand, "minimal" also means that there must also be enough information to infer the next unique state, if the governing equations are known[4]. The *state* of the system is a point in the phase space, while a trajectory is a subset of the phase space that is collecting all state points which are connected to each other through the describing dynamics of the system. Each trajectory is distinguished from others through some chosen point in it, which is called the initial state if one addresses the

[2] α was used for a delay time in [22], but in this work it is used τ

[3] More accurate the degrees of freedom are the chosen representing coordinates of the phase space.

[4] Stochastic dynamical processes are excluded in this discussion.

time $t = 0$ to it. Not part of the phase space are parameters and a possible actuation which shapes the phase portrait, i.e. the set of all possible trajectories building up the phase space. Those quantities have an influence on the real degrees of freedom but not the other way around. If the named influences on the system are constant, the system is autonomous, that means that the update equations from one time step to the other are independent of the time. For a deterministic system, it is, that each state must be already fully contained as information of its predecessors. This relation

$$x(t) = F(\{x(t')\}_{t'<t}) \tag{7}$$

intuitively motivates Takens' theorem, such that a proper chosen subset of the predecessors build the mentioned minimal set of dimensions, the degrees of freedom and can therefore be called a *phase space reconstruction*. Methods to find such a proper set of predecessors is topic of the sections 3.1.2 and 3.1.3. Further information on phase space reconstruction can be found in chapter 9 of [24]. A deepening in the direction of non-linear systems is offered by [25].

Finally, the actual implementation is explained. The delay embedding, as used for the machine learning topic introduced in section 3.2, is done via

$$\phi_m(X)^t = \left(X^{t+\tau m}{}_1, ..., X^{t+\tau m}{}_q, X^{t+\tau(m-1)}{}_1, ..., X^{t+\tau(m-1)}{}_q, X^t{}_1, ..., X^t{}_q\right) \tag{8}$$

for the $N \times q$ design matrix X at time entry $t = 0, 1, ..., N$ for feature dimension $\mu = 1, 2, ..., q$. It uses the embedding dimension m and a delay of τ time steps. So $\phi_m(X)$ is again a design matrix but now with the shape $N' \times q'$ while

$$N' = N - (m-1)\tau \tag{9}$$

$$q' = qm\ . \tag{10}$$

The following two sections introduce the methods for determining delay τ and embedding dimension m.

3.1.2 Autocorrelation Function

As talked about in the previous section, we are going to expand the system state by adding predecessors to it. To maximize the information efficiency of the new added coordinates, the delay for the previous system states should be chosen such that the states are uncorrelated [5]. The so called *autocorrelation function* or short ACF, calculates the correlation from a time series X^t with itself. Depending on the delay τ, the ACF maps to the interval of $[-1, 1]$ like the ordinary correlation and reveals at the first zero crossing a good value for the delay usable in time-delay embedding. This is because the trends of the system evolution are at this point completely uncorrelated. The explicit form of the ACF is

$$\mathrm{ACF}(\tau) = \frac{1}{N-\tau} \sum_{t=1}^{N-\tau} X^{t+\tau}{}_{\mu\equiv\mathrm{fix}} \cdot X^t{}_{\mu\equiv\mathrm{fix}} \tag{11}$$

[5] but they can still depent on each other

by just using one coordinate μ of the data series. It is also important that before the ACF is calculated, the data has to be normalized such that

$$X_\mu = \frac{X_\mu - \mathbb{E}_t[X_\mu]}{\sigma_t(X_\mu)}$$

with the mean $\mathbb{E}$ like (25) and the standard deviation σ like (24). The ACF yields for any phase shifted sine function again a cosine

$$\cos(\tau) = ACF(\tau)[\sin(t + \xi)] \tag{12}$$

where the correlation is 1 for $\tau = 0$, since the deviation of the data is exactly the same on the very same position for every time series. The point for a proper delay for the embedding would be $\tau = \pi/2$, since this is the smallest delay τ where the data is totally uncorrelated with itself and would perfectly embed the information of the phase in a two dimensional embedding. Where *perfectly* is emphasized, because for experimental data with noise it is the best chance to unleash the phase. On the other hand, for perfectly clean mathematical data also a $\tau = \epsilon$ for $\epsilon \ll 1$ would reveal the phase for a time shifted sine function.

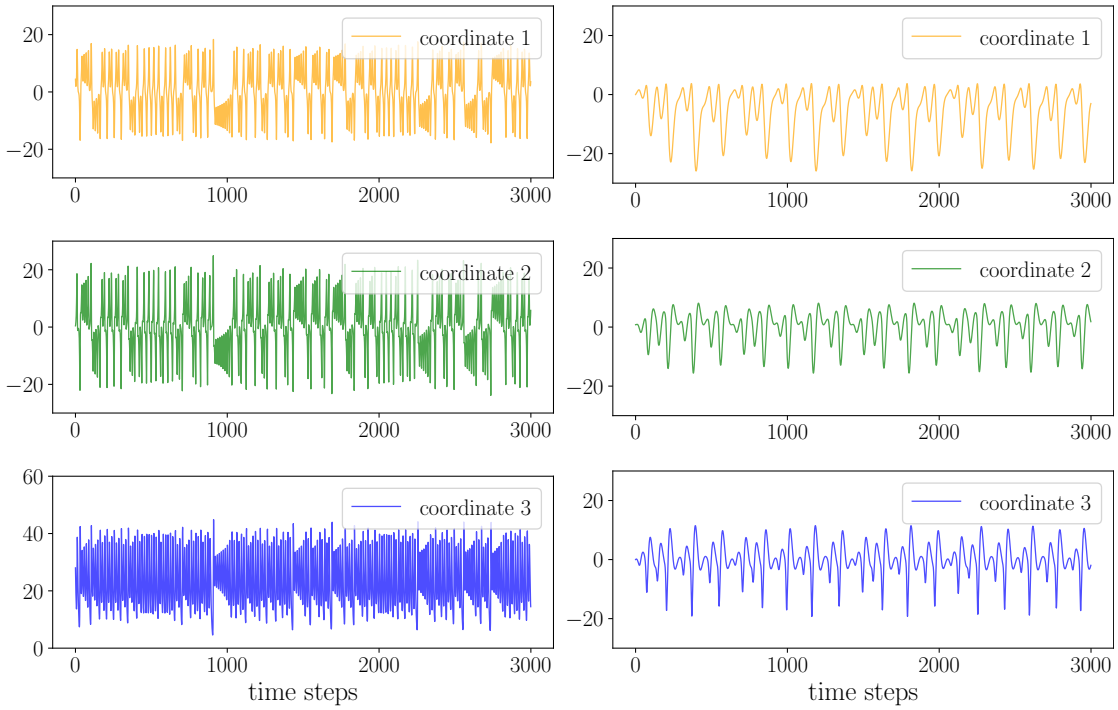
The Lorenz63 and WINDMI systems, which are described in the appendix and their projections to the 3 natural coordinates is shown in figure 5, are used here and also later on as demonstration and comparison for the results on the combustion data.

For determining the required delay, there are 5 samples used for both attractors respectively, to also get a standard deviation. There are not more than 5 samples used, to have the same number as for the combustion data later on, which in turn is limited. The results for the Lorenz63 and the WINDMI system can be seen in 6.

A different technique to estimate a proper time delay is the mutual information. It uses the Kullback–Leibler divergence [26] over the time series and its delayed version, instead the mean like in (11). Such that the first local minimum reveals the minimal time delay for two time instances of the data sharing as little information as possible [27]. Since the data has a strong linear behavior (see 4.1.1) and the ease of implementation, the ACF is chosen for the determination. In the next chapter, we will also determine the number of time delays which will be used for the time-delay embedding.

3.1.3 False Nearest Neighbors

For the false nearest neighbors algorithm (FNN) we need actually the estimated delay from the previous section. The FNN method goes back to the work of Kennel et al. in 1992 [28], where the minimal embedding dimension for, amongst others, observational data should be determined. Further context can be found in chapter 10 in [24]. To unfold the dynamics, or in other words, to construct the phase space, the FNN is based on the idea that *close stays close* applies to data points, if the embedding is properly chosen [29]. Therefore two data points ${X^{t_1}}_1$ and ${X^{t_2}}_1$, which must be nearest neighbors with their observational coordinate 1 and there successors τ time steps later, where the τ is determined through the previous section, are tested to see if they stay close together. With ϕ_d, the embedding function (8), the two data points are getting mapped in a phase space of increasing dimension d, starting by 1. If the two future points ${X^{t_1+\tau}}_1$ and ${X^{t_2+\tau}}_1$ in the data set keeping a close distance relative to the two claimed nearest neighbors in the embedding space $\phi_d({X^{t_1}}_1)$ and $\phi_d({X^{t_2}}_1)$, the statement



(a) *Lorenz63 attractor* ***(b)*** *WINDMI attractor*

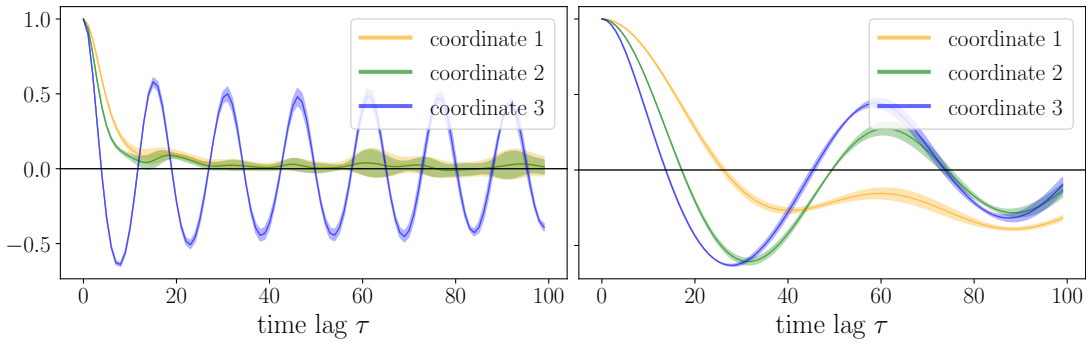
Figure 5: *The three projections of the generated phase space trajectories of a Lorenz63 and WINDMI attractor. These 3000 time steps long data sets are used as examples for the following analysis methods. They serve as a comparison for the combustor data evaluated later.*

of being nearest neighbors is proven true. Otherwise the two points $\phi_d(X^{t_1}{}_1)$ and $\phi_d(X^{t_2}{}_1)$ are wrongly declared as nearest neighbors because of a too small embedding dimension. The decision, if the two points kept their nearness, is doe to the inequality

$$\frac{|X^{t_1+\tau}{}_1 - X^{t_2+\tau}{}_1|}{\|\phi_d(X^{t_1}{}_1) - \phi_d(X^{t_2}{}_1)\|} \leq 15 \tag{13}$$

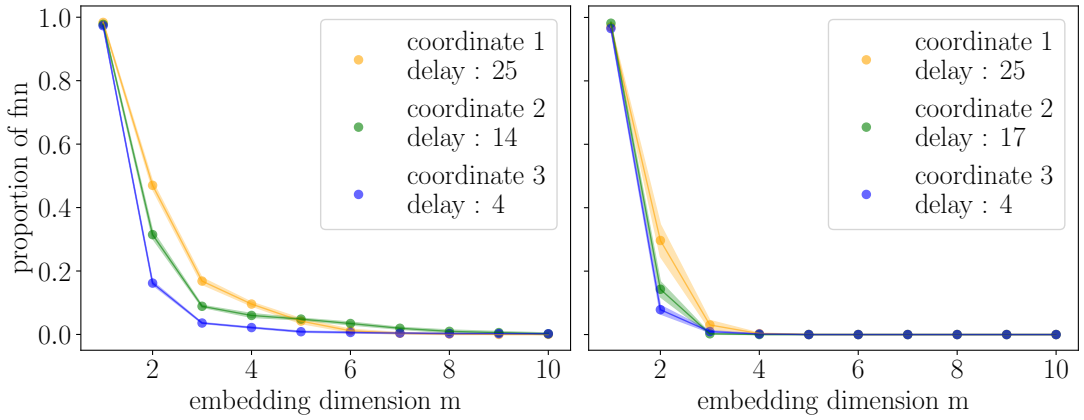
and a threshold, which is set to 15. In [29] a threshold between 10 and 15 is recommended, while a larger value is better for limited data. A visual representation of the algorithm is in figure 8. The procedure will be made until the embedding dimension d is large enough such there are no more false nearest neighbors detected. An example with the resulting diagram of the percentage of false nearest neighbors within the total number of pairs against the embedding dimension can be seen in figure 7. Since the measured data time series consists of just one trajectory, a chaotic behavior of the system does not harm the algorithm. A complete deterministic data series will ensure that the algorithm works, noise on the other hand, which usually exists for real data, does distort the outcome such that the result is subjective [30].

What is also interesting is that the FNN algorithm can also be used as a measure for determinism, as it is done in [31]. The work of [29] generalizes the approach by considering the conservation of neighborhood not just for the successors $X^{t_1+1}{}_1$, $X^{t_2+1}{}_1$



(a) Lorenz63 *(b) WINDMI*

Figure 6: *The autocorrelation functions for the data sets shown in figure 5 for the Lorenz63 and the WINDMI attractor. Each of the 3 dimensions is evaluated separately with 5 × 3000 time step examples and is given with the standard deviation. For the Lorenz63 attractor, the mean value crosses zero for the first time at 25, 14 and 4 time steps for the coordinates 1,2,3. With the WINDMI attractor, the middle first zero crossings are at 25, 17 and 4.*

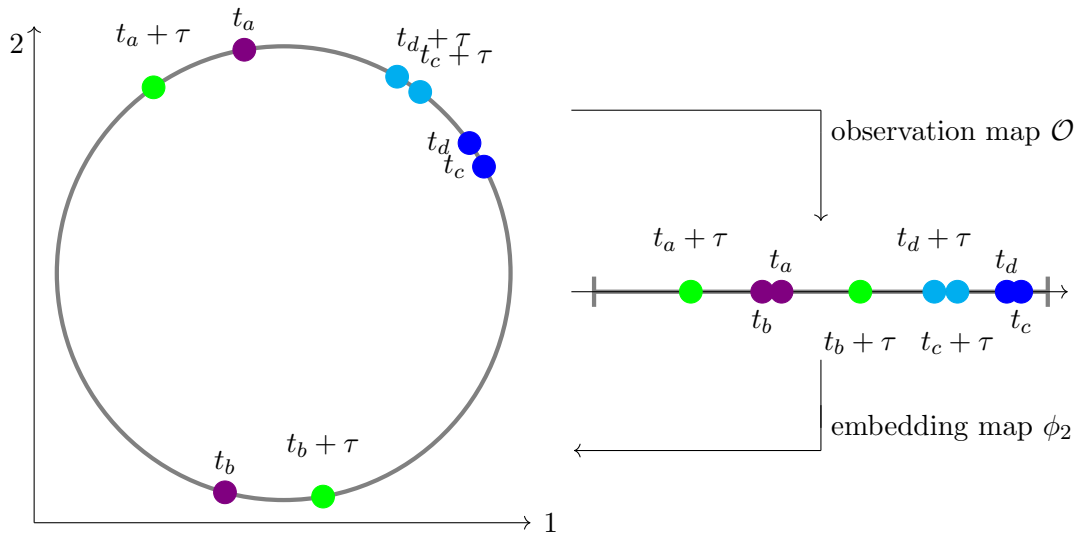


(a) Lorenz63 *(b) WINMDI*

Figure 7: *Here is the FNN algorithm for the data series introduced in figure 5. The mean world value and the standard deviation is given for 5 different samples, each with 3000 time steps. A threshold of 15 is used, for the inequality (13). The Lorenz63 attractor has an unusually long tail. The WINDMI attractor goes to zero at an embedding dimension of about 3 or 4, which is to be expected.*

of given points ${X^{t_1}}_1$, ${X^{t_2}}_1$ but regarding to mappings f in general $f({X^{t_1}}_1)$, $f({X^{t_2}}_1)$. An alternative for estimating the minimal embedding dimension for a dynamical system is Cao's method [30] which is also used on the combustion data in [12].

After the theory for the processing part for the data is completed, the next section actually starts over with the machine learning foundation on reservoir computing.



(a) *Original phase space* ***(b)*** *One dimensional phase space of data*

Figure 8: *Figure (a) shows the dynamics of a uniformly passed circle S^1 in $\mathbb{R}^2$ with sampled data points at time steps t_i and their successors with a lead of τ. One has to previously determine τ, for example, via the ACF method. This is a model for a not fully known dynamical system, where Figure (b) represents the observational data one has available. In this case, the observation is a map $\mathcal{O}$, which just projects on the 1 coordinate. If one applied a ϕ_1 embedding according to equation (8), the one-dimensional phase space would remain. The hypothesis that t_a and t_b referring to real nearest neighbors in this space would yield in equation (13) a label of "false" because the nominator is much larger than the denominator. If the embedding is done with ϕ_2, one would in general get a rotated ellipse, and if τ was well determined, it would be a circle again. The hypothesis of two points being real nearest neighbors, when determined in the two-dimensional embedding, the quotient of (13) would be close to 1 since two small distances are being compared and hence the hypothesis is tested to be true. The hypothesis test for t_c and t_d would be in both case ϕ_1 and case ϕ_2 close to 1 since two small distances are being compared. With that, smaller than the threshold and being true. Overall, in the two-dimensional embedding ϕ_2 would be no falsely labeled nearest neighbors. Hence, for an embedding dimension of 2, the portion of false nearest neighbors goes to zero.*

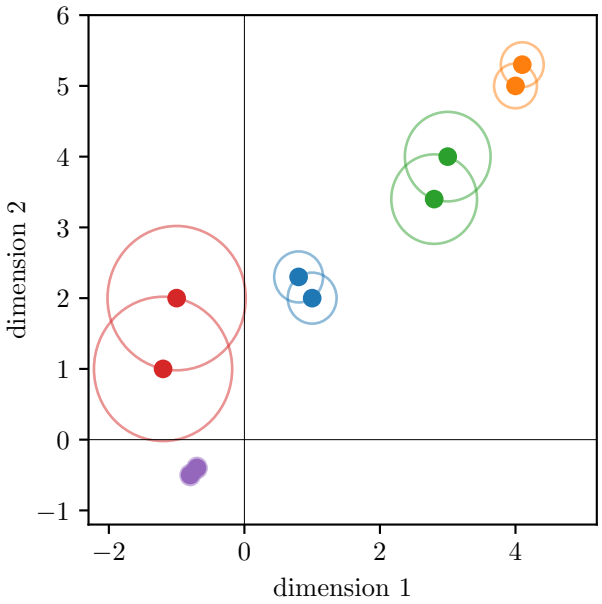


Figure 9: *Demonstration of the Find Nearest Neighbors algorithm as part of the FNN on a two-dimensional data set. As shown, one must calculate the distance for all combinations of the data points except the identities, and then choose the partner with the smallest distance for each data point.*

3.2 Reservoir Computing

This chapter saddles the foundation of the machine learning part for this work. It is split into two levels of detail. While the first part classifies and describes reservoir computing as an object of general machine learning methods, the second part specializes in the details of a concrete creation of an RC model.

3.2.1 Overview

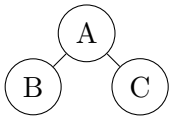
The method used, the reservoir computing (RC), also known as Echo State Machines (ESN)[6] based on the work of [4], or Liquid State Machine (LSM) referring to the Approach from [5], is a method from the field of supervised learning. That means that for given data X, there is also a well defined codomain of data Y, such that for each sample i, there is some relation $Y^i = f(X^i)$. Hence, if the machine learning model once adapted to a so called training set of given $\{(X^i, Y^i)\}_{i\in\mathbb{N}}$ it can (hopefully) later on generate new Y^i on unseen data X^i in the fashion of f. The most famous group of representatives for supervised learning are the group of neural network models, which achieve through information propagation through complex designed graph structures astonishing results in highly diverse fields of problems. For situations where the input X depends on its precursors, Recurrent Neural Networks (RNNs) [32] come into play. They are artificial neural networks that allow cycles in their network structure. Through the cycles, RNNs are capable of memory, that means that numerical artifacts of previous inputs are somehow saved and affect the computation of later inputs. Subcategorical to Recurrent Neural Networks, also Reservoir Computing models (RCs) are well suited to sequential, i.e. time series data, where the sample index is a time step $i \equiv t$. Therefore, the fundamental concept to learn is the relation $Y^{t+n} = f(X^t)$. RCs, where this loop-memory structure is realized through fixed reservoirs within neural networks, have shown good and fast learning behavior on complex and also chaotic dynamical time series data [33].

The reservoir computing can be understood in different ways. The first way to think about common RC, is that it is a 3 layer architecture, where the first layer is a once generated and fixed transformation matrix, which simply maps a certain time step of the input data X^t onto each node of the second layer. This matrix W_{in} is described by either statistical quantities or its generating algorithm. The second - hidden - layer is the reservoir itself, which are N_R nodes connected in a fixed manner to each other. This network suffices the sometimes used name *echo* state network, since every signal once propagated into the network, remains there through the large number of connections for a while, until it eventually vanishes. Like waves in a water bucket after a stone was thrown in [34] or a sound signal stays and overlaps with further ones in a valley of mountains. How long one signal stays within the network is a vast subject and often called the memory of the network [35]. This network is then be understood as its $N_R \times N_R$ dimensional adjacency matrix A, which contains the edges of the network like in figure 10 demonstrated. The third and final layer, called the *read-out layer*, is the only trainable part of this three-part structure and is basically an ordinary FNN. But as this FNN is usually chosen to be just one layer, i.e. it is a matrix W_{out}, it can be optimized by an analytical solution, as it is done in section 3.2.2. That's why RCs are fast to train, since there is no backpropagation needed, unlike most other RNNs.

[6] In this work it is often simply referred to as "reservoir".

	A	B	C
A	0	1	1
B	1	0	0
C	1	0	0

(a) adjacency matrix - table representation



(b) adjacency matrix - graph representation

Figure 10: *Relation between an adjacency matrix A and the corresponding network structure. Each matrix entry represents a link in the graph.*

The forward propagation in the ESN is built up by three equations. The time series X at time step t as input, gets together with the reservoir history R^t at the same time instance mapped to the next reservoir state

$$R^{t+1}{}_{\mu} \overset{\text{train}}{=} f_a \left(R^t{}_{v} A^{v}{}_{\mu} + X^t{}_{\xi} W_{\text{in}}{}^{\xi}{}_{\mu} \right) \tag{14}$$

by the already mentioned adjacency matrix A, and the input matrix W_{in}, while processed through an activation function f_a, which is set to $\tanh(\cdot)$. The indices ν and μ reaching up to N_R, while ξ ranges up to the spatial dimension of the input signal. Afterwards, the reservoir state is getting used to map it with the trainable W_{out} to the actual output

$$Y^t_{\mu} = \psi_e(R^t{}_{\nu}) W_{\text{out}}{}^{\nu}{}_{\mu} \tag{15}$$

of the reservoir, while here the range of ν is determined through the chosen embedding ψ_e which is further described in section 3.2.2. The t index ranges over the complete snippet of the time series that is handed over. For the reservoir computing there are three different phases distinguished, though. The first is the training phase, where the snippet of the time series is used to calculate the optimization on W_{out}. A second snippet serves in the so-called *synchronization phase* to fade out the initial state of the reservoir R^0 and adapt totally to the data, such that afterwards in the prediction phase a seamlessly connected output is generated. For the prediction does equation (14) changes to

$$R^{t+1}{}_{\mu} \overset{\text{pred}}{=} f_a \left(R^t{}_{\nu} A^{\nu}{}_{\mu} + Y^t{}_{\xi} W_{\text{in}}{}^{\xi}{}_{\mu} \right) \tag{16}$$

where instead of an input X now its own output Y from the previous time step serves as input. For that reason the RC runs after the training and synchronization stepwise autonomous for the future evolution of the data system. While this first way of understanding the RC has the advantage that it brings to the fore the fascinating fact, that the reservoir has its own dynamical properties, it is now also introduced a second and equivalent way of understanding RC, where the reservoir is not understood as the network but just as a hidden unit in the language of neural networks. So this work will not follow the conventional visualization of the RC model with the emphasis on the complex network structure caused by the random edges, as it was already done in the original work by Jaeger [4], but it introduces instead the mathematically more workable representation by *computational graphs*, which is more common for NNs and especially RNNs [36]. An introduction to the concepts of computational graphs can be found in literature like [37, 13]. In this representation the centerpiece of the reservoir becomes the N_R dimensional vector R^t of the reservoir states at time t, which is not to

be confused with the entire R matrix. Each node in a computational graph stands for a function on the information passed by the incoming edges and serves as a variable for the child nodes. The R^t vector is the part of the variable and (14) and (16) are the functions, respectively. Using this formalism, the entire training and synchronization process for the common RC method can be presented as a unfolded graph diagram like in figure 11 and the test or use case as in figure 12.

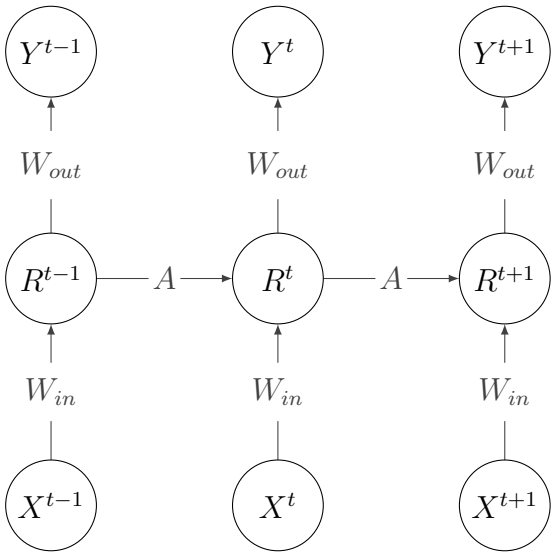


Figure 11: *Unfolded graph diagram for the used reservoir computing during the training and synchronization phase.*

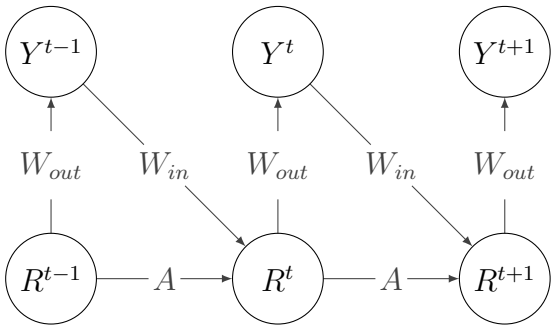


Figure 12: *Unfolded graph diagram for the used reservoir computing during the prediction phase.*

While the first explanation for the reservoir computings yields a good starting point for a further design of the reservoir itself, the second one enabled an easier comparison with other NN structures and therefore a better modification and integration in other structures. Now that the theoretical introduction is complete, we will move on to the concrete implementation for an ESN as used in this thesis.

3.2.2 Implementation

In this work it was used a relative large number for the nodes within the reservoir, i.e. $N_R = 3{,}000$. It is chosen such that it can run on current PCs with a flexible range for the embedding dimension and length of the data series, but is also large enough that this hyperparameter is in no means a limitation for the considered task. For works that are going for especially minimal N_R for maximal performance and RAM efficiency, onc can take a look for example to work of [38]. A complete list of all parameters which are set in this thesis and thus also those of the ESN, can be found in the appendix. The building blocks in this section should explain the explicit construction of the ESN in such a detail, that it is possible to transfer it into a code structure.

Generation of the Recurrency Matrix and the Spectral Radius

The adjacency or recurrence weight matrix A which maps the reservoir states R^t at time step t to itself at time step $t+1$, is sampled by a stochastic process. From the viewpoint of graph theory, the sample technique used is the Erdös-Rènyi model [39], where the number of vertices and the number of edges is fixed, but the assignment is done via a uniform distribution. The number of vertices, which are the same as the nodes, is called the *dimension of the reservoir* N_R and is an important parameter to set. In graph theory the number of edges for each node is the degree of the node. With the *average degree* parameter is the number of edges determined in such a manner, that the probability and hence roughly speaking the percentage of the adjacency Matrix elements $A^{\mu}{}_{\nu}$ not being zero is settled. The relation is

$$\text{percentage of non zero entries} \approx \frac{\text{average degree}}{\text{reservoir dimension} - 1}$$

such that the average degree has to be a number within the interval [0, reservoir dimension -1]. Since the average degree for this work is set to 280, about 9.3% of the matrix entries in A are non-zero. The generated recurrency matrix A doesn't have any diagonal elements, which means in the language of graph theory that the reservoir is loop free. This in turn means that a node can not contribute to itself via equation (14) or (16). So an activation signal has to travel through various nodes to eventually turn back to the node itself, which happens of course usually though several paths simultaneously as the reservoir is in general a cyclic graph. The second important step to design the matrix A, which maps the information back into the reservoir, is to adjust the strength of the connections between the nodes. Therefore each edge, that means each entry in the matrix, gets a random $\mathbb{R}$ number, positive or negative, sampled from a uniform distribution on a finite interval. The *spectral radius* defined as the maximal eigenvalue λ_n of A

$$r_{\text{spec}} = \max\left(\{\lambda_n\}_{n\in\{0,\dots,N_R\}}\right) \tag{17}$$

is the parameter that normalizes the initially random chosen connection strength within the reservoir and is related to the memory capability of the network [35]. Where at the beginning just spectral radii smaller than 1 are considered to be sufficient, the application of an activation function such as the $tanh(\cdot)$ used in this work, allows also spectral radii bigger than 1 [40], as it can be also seen later on in the results. On one hand, the matrix elements being zero or not is symmetric, but the graph is still a

directed graph, because of the random weighting of the edges. That means that the properties

$$\begin{gathered} A^{\mu}{}_{\mu} = 0 \\ A^{\nu}{}_{\mu} = 0 \iff A^{\mu}{}_{\nu} = 0 \\ A^{\nu}{}_{\mu} \neq A^{\mu}{}_{\nu} \end{gathered}$$

are true for the used reservoir. The network topology in reservoir computing is a topic of ongoing research [41, 42, 43], which also reflects mutual inspiration for the research in the field of artificial intelligence between biology and technology, as it is also used to understand the topology in our brains [44].

Generation of the Input Matrix

The matrix that transforms the input vector to the vector of the reservoir states, W_{in}, is generated through a random sampling from a uniform distribution of values in the interval determined by a *scale parameter*, which sets the max norm

$$||W_{in}||_{max} = \max_{(\mu,\nu)} \left(|W_{in}{}^{\mu}{}_{\nu}|\right) \stackrel{!}{=} W_{in} \text{ scale}$$

and is in this work W_{in} - scale $= 0.8$. Furthermore, W_{in} is set to be *sparse*. This means that just one input-dimension X_{μ} was projected for each reservoir state R_{ν}. As mentioned before, the input matrix W_{in} is after the initial generation fixed and does not participate in the training process.

Synchronization Process

The so-called synchronization of the reservoir is applied previously to the training and the prediction procedure. It applies just the same algorithm as for the prediction process like in equation (16), such that the initial zero state $R^{0}{}_{\mu} = 0$ is substituted with the current state right after the synchronization $R^{\text{sync}}{}_{\mu}$. The *sync-steps parameter* is set to 1000 time steps. Through the fading memory property [4] it should be guaranteed that the initial zero state of the reservoir does not affect the dynamics of the reservoir anymore, but just the input data. The data for the synchronization process is the same as for the training or prediction phase but just the first time steps respectively.

Ridge Regression, the Output Matrix and the Regression Parameter

For a fixed RNN structure, like in this work, where only a linear readout layer has updatable weights, the training can be done via Regularization [45, 46]. The regressand $\hat{Y}$ corresponds to the complete time series, but one time step ahead $\hat{Y}^{t}_{\mu} \equiv X^{t+1}{}_{\mu}$, where X is the design matrix with sample dimension t and feature dimension μ, which is used as training-input data. As cost functional $C(\cdot)$ for the training, the least squares objective with an additional L^2-regularization term

$$C(W_{\text{out}}) = \left(\hat{Y}^{t}_{\mu} - \psi_e(R^{t}{}_{\nu}) W_{\text{out}}{}^{\nu}{}_{\mu}\right) \left(\hat{Y}^{\mu}_{t} - W_{\text{out}}{}^{\mu}{}_{\nu}\, \psi_e(R^{\nu}{}_{t})\right) + \beta W_{\text{out}}{}^{\mu}{}_{\nu}\, W_{\text{out}}{}^{\nu}{}_{\mu} \quad (18)$$

is used, and the complete history of the reservoir states R, which evolved during the prediction process $X^t \mapsto X^{t+1}$ on the training data. This objective has the well known Ridge-estimator [47]

$$\hat{W}_{\text{out}}{}^{\mu}_{\nu} = \left(\psi_e(R^{\mu}{}_t)\psi_e(R^t{}_{\nu}) + \beta\delta^{\mu}{}_{\nu}\right)^{-1} \psi_e(R^{\mu}{}_t)\hat{Y}^t_{\nu} \tag{19}$$

as an analytical solution, where β is the scalar Ridge- or *regression parameter* to choose. As it was proven in the work [48], that a numerically manipulation of the model is necessary to break the symmetry, which is that the objective $\hat{W}_{\text{out}}$ is the same for $R^t(+X)$ and for $R^t(-X)$ in the case of a $\tanh(\cdot)$ architecture, as it is used in this work. Otherwise the reservoir would be just universal for point-symmetric problems. A way to successfully tackle this problem, is to use a in [48] introduced *extended Lu readout*

$$\psi_e(R)^t = \left(R^t{}_1, ..., R^t{}_{\mu_{\max}}, \left(R^t{}_1\right)^2, ..., \left(R^t{}_{\mu_{\max}}\right)^2\right) \tag{20}$$

which is an embedding function, that concatenates the reservoir states with its squared values. It is named after the work of [49].

These 4 steps are the basis for the creation of all objects needed for reservoir computing as it is used in this thesis. In conjunction with equations (14), (15) and (16), the machine learning implementation is finally complete.

3.2.3 Reservoir Computing vs Time Delay Embedding

Since both methods, reservoir computing and time-delay embedding, combine current states with previous states, and thus both produce a memory effect, they should be compared side-by-side in this section. quote from [jeager 2001 and short] to reservoir computing (RNNs): Jaeger introduced in his original work on ESNs [4] the concept *input echo function* $\mathbf{E}$ that maps any "left-infinite input histories", simply said a, of course discrete, time-series data set $\{u(t)\}_t$ of arbitrary length N, to the current reservoir state $\mathbf{x}(t)$

$$\mathbf{x}(t) = \mathbf{E}(u(t), u(t-1), ...) \tag{21}$$

in a unique, injective way. The essential requirement for the non-trainable RC part, was called *echo state property* which states "the orbit of the reservoir in the state space should be asymptotically determined uniquely by the input signal, and the influence of initial conditions should progressively fade away" [50]. That means two reservoir states under the same input signal, but with different initial conditions, become asymptotically the same for sufficiently long exposure to the signal. Where time delay embedding uses a fixed and finite number of instances from the input history, and those also with a certain delay τ such that

$$\tilde{u}(t) = (u(t), u(t-\tau), ..., u(t-m\tau)) \tag{22}$$

does the ESN use as described flexible many instances with a $\tau = 1$ delay, but with vanishing contribution. Time delay embedding in combination with the ESN, the state

$$x(t) = E(\tilde{u}(t), \tilde{u}(t-1), ...) \tag{23}$$

gets theoretical redundant information. But by implementation there is an additional memory limitation through the precision of floating-point digit numbers, and the properties of the existing randomized reservoir network. This doubly induced memory capability, may be more effective for data with a high intrinsic delay τ like discussed in section 3.1.2, or a large product of delay and delay embedding dimension $\tau \cdot m$, to overcome limitations of the memory capacity of the reservoir [35] and computational restrictions. A great work regarding this topic, where the ESN is considered as an embedding on its own, can be found in the 2020s work [51].

Now that we have concluded the discourse on the machine learning model and data preprocessing, we will present both known and new methods to assess the performance of the system and to study the data and ESNs themselves.

3.3 Performance Measures

In this third methodological chapter, we will define a broad range of measures that capture different geometrical properties of data and the prediction. The measures are not just tools to test the performance of the RC on the combustion data, but they are also research topics by themselves. Each measure has different strengths for different situations, like whether or not they capture linear or nonlinear features, or are prone to noise. Those measures can also be used for early detection of transitions in dynamical states as they give quantitative feedback on the training performance or information about the original data itself.

3.3.1 NRMSE

To compare the quantities, introduced in the following sections, we want a positive semidefinite map, that is strictly decreasing with a strictly decreasing deviation between the predicted and the real values. This is necessary to get an ordered and well defined term of quality. It should hold for a discrete set of observational data and is applied to one-dimensional or scalar data. A root mean squared error (RMSE) meets these requirements, but it will be used in a normalized version (NRMSE) as will be described below. Let us consider two data sets $A = \{A_i\}_i$ and $B = \{B_i\}_i$. We will use the standard deviation, which is defined as

$$\sigma_i(A) := \sqrt{\mathbb{E}_i[(A - \mathbb{E}_i[A])^2]} =: \sqrt{\mathbb{V}(A)} \tag{24}$$

with the variance $\mathbb{V}$. The mean $\mathbb{E}$ is defined as

$$\mathbb{E}_i[A] := \sum_i^m p(A^i) X^i \overset{\text{uni.}}{=} \frac{1}{m} \sum_i^m A^i \tag{25}$$

where the second equivalence holds if the data is already a sampled set or the probability of X^i, noted as $p(X^i)$, originates from an uniform distribution. With that given, one can construct a NRMSE as

$$\text{NRMSE}_i(A, B) := \frac{1}{\sigma_i(B)} \sqrt{\frac{1}{m} \sum_{i=1}^m (A_i - B_i)^2} \tag{26}$$

or alternatively, e.g. for data where the standard deviation is zero, like single value data, the

$$\mathrm{NRMSE}_i(A,B) := \frac{1}{|\mathbb{E}_i[B]|}\sqrt{\frac{1}{m}\sum_{i=1}^{m}(A_i - B_i)^2} \tag{27}$$

is used. To motivate the construction and usage of the NRMSEs, we will consider some of its properties. Both the standard deviation and the mean are homogeneous of degree 1, i.e. it holds

$$\begin{aligned}\mathbb{E}_i[\lambda \cdot A] &= \lambda \cdot \mathbb{E}_i[A] \\ \sigma_i(\lambda \cdot A) &= \lambda \cdot \sigma_i(A)\end{aligned}$$

for any $\lambda \in \mathbb{R}$. This means that the NRMSE is not only neutral in dimensions ("unit-free") but also completely unaffected from the order of magnitude of the data values

$$NRMSE_i(\lambda \cdot A, \lambda \cdot B) = NRMSE_i(A,B)$$

and can better be used as a method for performance comparison. Furthermore it is also independent from the sample size m, which is for time series data the number of time steps. A property that is directly inherited from standard deviation and the mean. One should notice, that the NRMSE is not symmetric

$$NRMSE_i(A,B) \neq NRMSE_i(B,A)$$

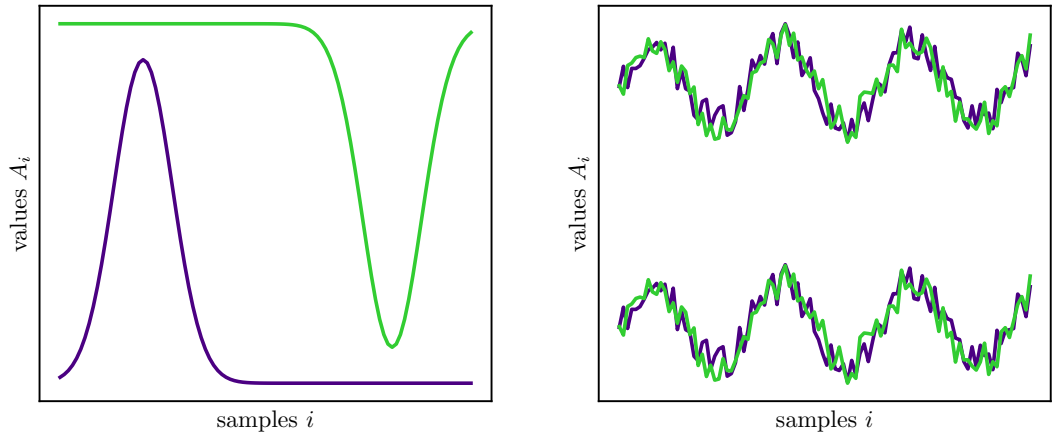
where it is recommended to use for the first entry the generated data and for the second argument the ground truth. This prevents falsely truncated errors when the prediction raises to values much larger than the true values. Up to this point, (26) and (27) behave absolutely equal. A big difference appears at the translation invariance

$$\begin{aligned}NRMSE_i(A+\lambda, B+\lambda) &= NRMSE_i(A,B) \quad \text{for (26)} \\ NRMSE_i(A+\lambda, B+\lambda) &\neq NRMSE_i(A,B) \quad \text{for (27)}\end{aligned} \tag{28}$$

since (26) is invariant to shifts but not the version which is normalized by the mean (27) as demonstrated in figure 13b. This has an impact on preference of those two normalization variants. Time series data for instance, have an advantage on the translation symmetry because they could generally also evolve around a zero mean but with a large range of amplitudes. This is what the normalization by the standard deviation does, it divides out the effective amplitude of the data. Although within this work the time series is not measured by itself with the NRMSE, but indirect quantities, there is still another difference between (26) and (27). The $1/\sigma$ normalization yields the symmetry

$$\begin{aligned}NRMSE_i(A,-A) &= NRMSE_i(-A,A) \quad \text{for (26)} \\ NRMSE_i(A,-A) &\neq NRMSE_i(-A,A) \quad \text{for (27)}\end{aligned} \tag{29}$$

in contrast to the mean normalization. This is a very intuitive property for an error measure, as visualized in figure 13b, and it is also fulfilled by the simpler RMSE and MSE. For those reasons equation (27) used for scalar comparisons only, and (26) for all others.



(a) ***(b)***

Figure 13: *In these symmetry plots, the prediction data is shown as a green line and the purple line shows the true data. Figure 13a shows data that, by exchanging both data, would give the same error value if evaluated with σ-normalization due to symmetry (29). For figure 13b, the upper pair of prediction and ground truth would also give the same error as the pair below. This is due to the translation invariance of (28). Both symmetries do not hold for the normalization with the means.*

3.3.2 Prediction Length

For time series prediction it is always an important measure, how long the predicted data stays close to the original data within the test case. In some cases this can already be the only quantity of interest, if the precise very next outcomes in time of an evolution are asked for. But in many cases, as in this work here, the more overall adaptation of a dynamical system is of interest, or in other words, to mimic the system. So this measure of exact forecast is just of complementary nature to the other measures, which makes it more or less mandatory for time series models to investigate. The measure is set as

$$\text{prediction length} := \underset{t}{\operatorname{argmin}} \left(\sqrt{\left(Y^t_{\mu=1} - \hat{Y}^t_{\mu=1} \right)^2} \geq \xi \cdot \mathbb{E}_t \left[\left| \hat{Y}_{\mu=1} \right| \right] \right) - 1 \tag{30}$$

which calculates for each time step t the absolute difference between the predicted Y and the original $\hat{Y}^t$ data for the projection on the first spatial coordinate $\mu = 1$. The prediction length is then the last time step where a certain threshold of difference has been exceeded. The free parameter to choose is the ξ, which is chosen to be 0.5 in this work. This is because of data which is oscillating with a very small period, as the combustion pressure, even a small phase difference can yield a huge diversion because the slopes are so steep. Through the parameterization with the proportion of the mean absolute value, one has a data independent comparability that also works for strictly positive data, as well as for oscillating data.

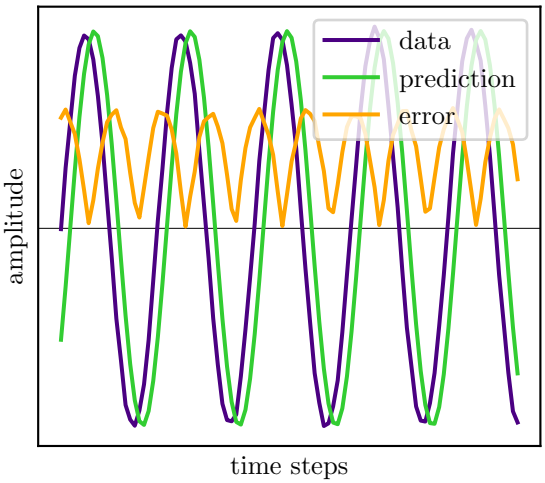


Figure 14: *This figure shows a prediction with perfect frequency and amplitude reproduction and how even a small change in phase can result in a large error in data with short periodicity.*

3.3.3 Correlation Dimension

Chaotic systems, i.e. dynamical systems which are non-periodic and numerically unpredictable, play a central role in nature, technology and everyday life. Because of this, belongside to multiple statistical approaches, which were invented to determine quantitative characteristics belonging to a specific chaotic system, the spatial dimensions playing a central role. The following method is a computational feasible and hence common way to estimate the fractal $\mathbb{R}$-valued dimension of the attractor. This concept is heavenly based on the work of [52] where the correlation exponent and its estimator, the correlation integral, is analyzed as characteristic for strange attractors. The discrete version of the correlation integral, the correlation sum

$$C(r) = \frac{1}{N^2}\left[-N + \sum_{(t_1,t_2)=(1,1)}^{(N,N)} \theta\left(r - ||X^{t_1} - X^{t_2}||_2\right)\right] , \tag{31}$$

counts through the Heaviside step function $\theta(\cdot)$ the proportional number of neighbors for each time series element X^t depending on the ball-radius r. The number $C(r)$ is supposed to rise like the

$$C(r) \sim r^{\nu} \tag{32}$$

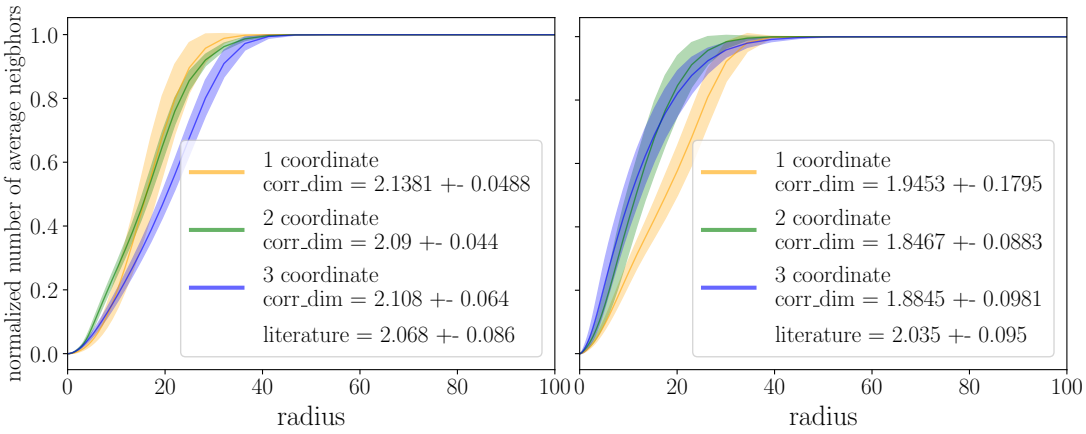
exponential, where the correlation exponent ν is also called the *correlation dimension*. The latter one is applied as a representative measure for the time series data used and produced in this work. The algorithmic realization created for this work samples the radius r values in a logarithmic allocation from 0.1 times the smallest distance within the data to 10 times the longest distance given in the data. This allows more stress on smaller distances and stabilizes the ν-estimation. The distance refers to the euclidean distance:

$$||X^{t_1} - X^{t_2}||_2 = \sqrt{\left(X^{t_1}{}_{\mu} - X^{t_2}{}_{\mu}\right)\left(X^{\mu}{}_{t_1} - X^{\mu}{}_{t_2}\right)} \tag{33}$$

The resulting $C(r)$ curve like in figure 15 gets logarithmic rescaled such that equation (32) gets

$$\ln(C(r)) \sim \nu \cdot \ln(r) \ . \tag{34}$$

In this form it is possible to linearly fit the slope and thus get ν. The algorithm is tested on generated Lorenz63 and WINDMI data and compared with the literature values. The resulting correlation sum and its standard deviation can be seen in figure 15.



(a) *Lorenz63* ***(b)*** *WINDMI*

Figure 15: *Calculation of the correlation dimension for the data sets in figure 5. Each coordinate is again embedded and calculated separately with the delay determined in figure 6. It is used an embedding dimension of 3. For each coordinate 5 samples, á 3000 time steps are used. All calculated correlation dimensions have an overlap with the literature values of [24], except for the second coordinate of the WINDMI attractor.*

Since the correlation sum does not have any slope for radii which are smaller than the space between the data points or larger then the complete occupied area in the phase space, one can not compute the slope of (34) for the complete correlation sum $C(r)$. So it was decided that the slope is taken between 30% and 70% of the y-axis-range for the "log-log relation" (34). This results in a stable algorithm for all tests done. It also shows little dependence on the data size, as it was tested from 2000 to 6000 data points, each result within the standard deviation of the other. Furthermore it should be mentioned that the dimension of a dynamical system is generally lower than the degrees of freedom within its phase space.But for the case of a totally unknown mathematical framework, as in the case of measured data from a complex natural phenomenon that is exposed to a lots of interplay, there is no fixed boundary given for the dimension of interest. But it is interesting to keep in mind during the evaluation of the combustor data. The most critical point on this measure is how the embedding dimension of the data is chosen. As one can see in figure 23, proper simulated dynamical systems should be largely independent of the chosen embedding dimension if it once surpasses the value of the intrinsic dimension.
At last, it should be mentioned the exact NRMSE measure, which is used to calculate

the performance of the predictor. It is the difference

$$\mathrm{NRMSE}_{\mathrm{corr\_dim}} = \mathrm{NRMSE}\,(\nu, \hat{\nu}) \tag{35}$$

between the correlation dimension of the predicted data ν and the original data $\hat{\nu}$. To get further scalar quantities for the comparison, the next section will introduce the recurrence quantification analysis.

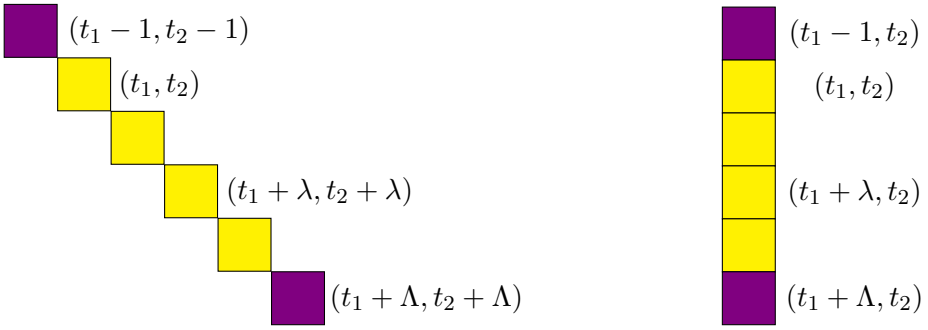
3.3.4 Recurrence Quantification Analysis

Some kind of similar to the prediction length, but in a global manner, does the field of recurrence quantification analysis (RQA) measure a 'staying close by each other" of trajectories, but this time each time series by itself. A deterministic dynamical system within its bounded phase space has to recurrent to a certain phase point at some time. Since dynamical systems within a proper embedding do not have crossing trajectories, a hyperparameter for the radius r of the neighborhood size has to decide if the system is returned to a state or not. Based on that idea, RQA is a powerful way to geometrically represent an arbitrary dimensional time series data set. Consisting of $\Delta t \times \Delta t$ sized pixels, an image can be created with 0 or 1 values that contains the information, if the trajectory at time i is close to its instance at time j or not. This image, called *recurrence plot* (RP) has a matrix equivalent R, thats formal expression

$$R_{t_1 t_2} := \theta(r - ||X^{t_1} - X^{t_2}||_2) \tag{36}$$

is based on the *Heaviside step function* θ and the euclidean norm $||\cdot||_2$. The resulting pattern in the image can then be described by quantities, which lead back up to the original geometry of the phase space trajectory. Based on the very comprehensive book [53] about this topic, a subset out of the broad range of existing RQA measures will be introduced yet, to reflect certain properties of the dynamical time series. The selection is based on the work of Waxenegger-Wilfing et al. [12] for the sake of comparability. First of all, one distinguishes two different micro structures in the recurrence plot, which are shown in figure 16. While *diagonal lines* (D) are related to the deterministic structure, i.e. low divergence of the flow in the phase space, do horizontal and *vertical lines* (V) refer to singularities and laminarity. The latter means that the system state stagnates at some points in the phase space [54]. This is possible since the discrete time points of an embedded line in a n-dimensional space vary in distance, dependent on the one dimensional parameter change, i.e. the speed. So can state X^{t-1} be within the neighborhood of X^t or not. Furthermore, since the neighborhood is here constructed in a way that the matrix R is symmetric, i.e. $R_{t_1 t_2} = R_{t_2 t_1}$, horizontal and vertical lines are equivalent. Diagonal lines, on the other hand, are from top left to bottom right if two parallel paths are going in the same direction, or they are from bottom left to top right if the two paths are counter-directed. The elements from figure 16 can be investigated in the recurrence plot through histograms, dependent on their appearance with a certain pixel length Λ. The bins for the diagonal lines $H_D(\Lambda)$ and the vertical

[7] Since the recurrence matrix R is symmetric, the indices in figure 16b and the horizontal and vertical alignment of pixels can be interchanged.



(a) *diagonal line* ***(b)*** *vertical line*

Figure 16: *This is the basic method of making statistics about line length in a recurrence plot. In equations (3) and (4), inverting terms are used to set the start and end points to zero, which corresponds to purple, and then the multiplication term is used to test whether all pixels in between are from the recurrence matrix R 1, i.e. yellow. Thus, all lines with their respective lengths Λ are only counted once.*[7]

ones $H_V(\Lambda)$ are therefore

$$H_D(\Lambda) = \sum_{(t_1,t_2)=(1,1)}^{(N,N)} (1 - R_{t_1-1,t_2-1})\,(1 - R_{t_1+\Lambda,t_2+\Lambda}) \prod_{\lambda=0}^{\Lambda-1} R_{t_1+\lambda,t_2+\lambda} \tag{37}$$

$$H_V(\Lambda) = \sum_{(t_1,t_2)=(1,1)}^{(N,N)} (1 - R_{t_1,t_2-1})\,(1 - R_{t_1,t_2+\Lambda}) \prod_{\lambda=0}^{\Lambda-1} R_{t_1,t_2+\lambda} \tag{38}$$

where the first two factors inside round brackets refer to the purple zero cells in figure 16 and the last product term to the yellow cells. First of all, the quantities *determinism* (DET) and *laminarity* (LAM)

$$\mathrm{DET} := \frac{\sum_{\lambda=\lambda_{min}}^{N} \lambda H_D(\lambda)}{\sum_{(t_1,t_2)=(1,1)}^{(N,N)} R_{t_1 t_2}} \tag{39}$$

$$\mathrm{LAM} := \frac{\sum_{\lambda=\lambda_{min}}^{N} \lambda H_V(\lambda)}{\sum_{(t_1,t_2)=(1,1)}^{(N,N)} R_{t_1 t_2}} \tag{40}$$

are the relations of the total number of diagonal and vertical lines respectively, to the total number of points (1s) in the recurrence plot. Both are based on the parameter λ_{min} which determines the minimal number of pixels to be counted as a line, which is set as usual to 2. The *average diagonal line length* (L) is similar to (39),

$$\mathrm{L} := \frac{\sum_{\lambda=\lambda_{min}}^{N} \lambda H_D(\lambda)}{\sum_{\lambda=\lambda_{min}}^{N} H_D(\lambda)} \tag{41}$$

but just considers the number of lines in the denominator. Also the straight forward quantity is used, which is the proportion of 1s and 0s in the plot, which is basically the

mean of the recurrence matrix

$$\mathrm{RR} := \frac{1}{N^2}\left[-N + \sum_{(t_1,t_2)=(1,1)}^{(N,N)} R_{t_1 t_2}\right] \tag{42}$$

and is called the *recurrence rate* (RR). Since it describes the density of points within the recurrency plot, it is similar to the correlation dimension, which also measures how close the phase space trajectory gets to itself. From inserting (36) in (42) one gets basically the same as for the correlation sum (31), the only difference is that (31) still depends on the radius r. Build up from the introduced measures, RATIO defined as

$$\mathrm{RATIO} := \frac{\mathrm{DET}}{\mathrm{RR}} \tag{43}$$

is said to be useful for dynamical state transitions [53]. With the relative frequency of diagonal line lengths, hence the normalized histogram

$$p(\lambda) = \frac{H_D(\lambda)}{\sum_{\lambda=\lambda_{min}}^{N} H_D(\lambda)}$$

the *Shannon entropy* (ENTR) applied on the patterns in the recurrence plot can be calculated as

$$\mathrm{ENTR} := -\sum_{\lambda=\lambda_{min}}^{N} p(\lambda)\ln p(\lambda) \tag{44}$$

which represents the "complexity of the deterministic structure in the system" [53]. It is called in the pyunicorn python package "diagonal line entropy" as one among other entropy methods.

From that point on the performance measures for the reservoir prediction can be calculated with (27) as

$$\mathrm{NRMSE}_{\mathrm{RR}} = \mathrm{NRMSE}\left(\mathrm{RR}_{\mathrm{prediction}}, \mathrm{RR}_{\mathrm{data}}\right) \tag{45}$$

$$\mathrm{NRMSE}_{\mathrm{DET}} = \mathrm{NRMSE}\left(\mathrm{DET}_{\mathrm{prediction}}, \mathrm{DET}_{\mathrm{data}}\right) \tag{46}$$

$$\mathrm{NRMSE}_{\mathrm{LAM}} = \mathrm{NRMSE}\left(\mathrm{LAM}_{\mathrm{prediction}}, \mathrm{LAM}_{\mathrm{data}}\right) \tag{47}$$

$$\mathrm{NRMSE}_{\mathrm{L}} = \mathrm{NRMSE}\left(\mathrm{L}_{\mathrm{prediction}}, \mathrm{L}_{\mathrm{data}}\right) \tag{48}$$

$$\mathrm{NRMSE}_{\mathrm{RATIO}} = \mathrm{NRMSE}\left(\mathrm{RATIO}_{\mathrm{prediction}}, \mathrm{RATIO}_{\mathrm{data}}\right) \tag{49}$$

$$\mathrm{NRMSE}_{\mathrm{ENTR}} = \mathrm{NRMSE}\left(\mathrm{ENTR}_{\mathrm{prediction}}, \mathrm{ENTR}_{\mathrm{data}}\right)\ . \tag{50}$$

In our case, those measures do not need to yield exact information about the system, but should provide a comparability between both, data and prediction. For the implementation it is used the pyunicorn package [55], where the matrix R and all quantities, except the NRMSE, can be calculated. The subtlety is that false information is coupled to the RQA measures since the radius r, especially for noisy real data, can capture several trajectory sections and time steps within them, which leads to thicker lines in the recurrence plot. Within those thicker lines fit additional vertical and diagonal lines of short length, which are not intended from the geometrical idea. This *recurrent threshold* r is set to be 0.5 times the standard deviation for all three regions respectively [56], since it shows reasonable structures in the plot and nontrivial quantities, i.e. an average length of lines is significantly larger than 2 pixels. Despite the broad variation in phase space diameter for the combustion data, all plots looked equally good.

3.3.5 Amplitute Distribution based Measure

Another way to measure the geometrical behavior of the time series is to consider the amplitude distribution of one of its coordinates. For this a histogram is a proper method to gain statistical insights into a finite discrete amount of samples. For a histogram the amplitude range from the maximal appearing value $v_{\max}$ and the minimal one $v_{\min}$ of a predicted one-dimensional time series $Y_{\mu=1}$ and its ground truth $\hat{Y}_{\mu=1}$

$$v_{\max} = \max\left(\{Y^t_{\mu=1}\}_t \cup \{\hat{Y}^t_{\mu=1}\}_t\right)$$
$$v_{\min} = \min\left(\{Y^t_{\mu=1}\}_t \cup \{\hat{Y}^t_{\mu=1}\}_t\right)$$

gets discretized in bins. The number of bins chosen in this work is $k_{\max} = 128$. For both time series in comparison the number of observations will be counted in bins of the same fixed width and value assignment

$$n_k = \left|\{y \in \{Y^t_{\mu=1}\}_t | (k-1)\frac{v_{\max}-v_{\min}}{k_{\max}} + v_{\min} < y < k\frac{v_{\max}-v_{\min}}{k_{\max}} + v_{\min}\}\right|_{\text{set}}$$
$$\hat{n}_k = \left|\{y \in \{\hat{Y}^t_{\mu=1}\}_t | (k-1)\frac{v_{\max}-v_{\min}}{k_{\max}} + v_{\min} < y < k\frac{v_{\max}-v_{\min}}{k_{\max}} + v_{\min}\}\right|_{\text{set}}$$

where $|\cdot|_{\text{set}}$ is the cardinality of sets. So for both data of time series length T

$$T = \sum_{k=1}^{k_{\max}} n_k \quad \text{and} \quad T = \sum_{k=1}^{k_{\max}} \hat{n}_k$$

holds. From this point the NRMSE from (26)

$$\text{NRMSE}_{\text{distribution}} = \text{NRMSE}_k\left(\{n_k\}_k, \{\hat{n}_k\}\right) \tag{51}$$

can be used to get a quantitative representation of the quality for a prediction. The choice of the number of bins is the ambiguity of this measure since it could lose its continuity for too many bins but could also hide a possible multimodality and its highly dependent on T. Since this method does not have any sensitivity for temporal information, it is interesting to also consider a totally temporal specified measure as the frequency based measure in the next section.

3.3.6 Frequency based Measure

Since the pressure data is oscillating, it raises the question if there are any fixed frequencies within the data. The linear properties of data can be measured by the frequencies ν contained in the oscillations. Therefore the one-dimensional discrete Fourier Transformation with the Fast Fourier Algorithm (FFT) from the work of J. W. Cooley and J. W. Tukey in 1965 [57] was used through `np.fft.fft()` from the python library Numpy [58] was used to calculate the data's amplitudes $\mathcal{F}[Y_{\mu=1}](\nu)$ within the frequency space. From those amplitudes the normalized absolute value of the positive frequencies were taken

$$f_+[Y_{\mu=1}](\nu) := \left|\frac{\mathcal{F}[Y_{\mu=1}](\nu)}{\sum\limits_{\nu,\nu>0} |\mathcal{F}[Y_{\mu=1}](\nu)|}\right| , \quad \text{where } \nu > 0 \ . \tag{52}$$

From that it was calcuated the NRMSE from (26)

$$\mathrm{NRMSE}_{\mathrm{Fourier}} = \mathrm{NRMSE}_{\nu}(f_{+}[Y_{\mu=1}], f_{+}[\hat{Y}_{\mu=1}]) \tag{53}$$

as a further prediction measure, where it should be mentioned that it was only calculated for the first coordinate $\mu = 1$ of the test $\hat{Y}$ and prediction data Y. Since the generated discrete Fourier amplitudes are sharing the same disctrete frequency array, the NRMSE is well defined. But since the NRMSE over the frequency space is extremely sensitive to the slightest shifts of sharp frequency spectra, since geometrically each frequency corresponds to a different spatial dimension in the geometrical interpretation of the NRMSE, there is the question of a more robust measure that forgives such small inaccuracies in the spectra. That is where the moment based approach of the next section is brought up.

3.3.7 Moments

To get a more smooth transition from bad to very precise matching between prediction data Y and its ground truth $\hat{Y}$, one can consider the moments of a value distribution. It ignores a spiky behavior if just the first moments are used. In this section the amplitude distribution based measure from 3.3.5 and the frequency based measure from 3.3.6 are reconsidered. For the amplitude based measure it has the advantage to get rid of the free parameter for the bin number. The l-th moment m_l is calculated for the amplitude distribution as

$$m_l = \mathbb{E}_t\left[\left\{\left(Y^t{}_{\mu=1}\right)^l\right\}_t\right] = \frac{1}{T}\sum_{t=1}^{T}\left(Y^t{}_{\mu=1}\right)^l$$
$$\hat{m}_l = \mathbb{E}_t\left[\left\{\left(\hat{Y}^t{}_{\mu=1}\right)^l\right\}_t\right] = \frac{1}{T}\sum_{t=1}^{T}\left(\hat{Y}^t{}_{\mu=1}\right)^l$$

for the data of T time steps wich allows to calculate a NRMSE

$$\mathrm{NRMSE}_{\text{Distrib. Moments}} = \mathrm{NRMSE}_{l}\left(\{m_l\}_l, \{\hat{m}_l\}_l\right) \tag{54}$$

with formula (26). This measure though comes also with a degree of freedom, induced through the choice of the maximal moment $l_{\max}$ to be used for the NRMSE. For this work the first 4 moments, i.e. $l_{\max} = 4$ are used for the comparison of the two time series, as well for the amplitude distribution as also for the frequency density distribution. The 0th moment, which is just 1, is not used. For the frequency spectra equation (52) is used to calculate the moments

$$m_l = \mathbb{E}_{\nu}[\{\nu^l\}_{\mathbb{I}}] = \sum_{\nu\in\mathbb{I}} \nu^l \cdot f_{+}[Y_{\mu=1}](\nu)$$
$$\hat{m}_l = \mathbb{E}_{\nu}[\{\nu^l\}_{\mathbb{I}}] = \sum_{\nu\in\mathbb{I}} \nu^l \cdot f_{+}[\hat{Y}_{\mu=1}](\nu)$$

where $\mathbb{I}$ is the set of frequencies used by the FFT algorithm as in section 3.3.6 described. With the resulting NRMSE

$$\mathrm{NRMSE}_{\text{Fourier Moments}} = \mathrm{NRMSE}_{i}\left(\{m_l\}_l, \{\hat{m}_l\}_l\right) \tag{55}$$

it is hoped that the frequencies could now be compared in such a manner, that when dominant frequency peaks are near by, do already result in good prediction quality values. As the measure of 3.3.6 just allows a smooth transition for the prediction quality when the frequencies are already fitting and just the strength has to be improved.

Results

This chapter is splitted into 4 topics. In 4.1 is the data from 2 further analyzed to build a foundation for the machine learning application afterwards. Therefore are three special snippets, called *regions*, out of one experimental run are such selected, that they represent three different dynamical states. The admired target state and one example for each of the two instability types. Those three regions will serve further for the complete RC application. In this section is also the delay embedding from 3.1.1 prepared for the RC data preprocessing. In part 4.2 are shown explicit results for the application of the RC on the data, such that an impression is made, how close a RC based prediction gets to the original data and how does the ESN fail if the wrong hyperparameters are chosen. The analytical discussion on which hyperparameter are actually the best for this task and how reliable the introduced measures from section 3.3 actually represents the performance of the prediction is topic of section 4.3. In the last part, section 4.4, deals with the subject of time evolving RC adaption and quantity shifts with a sliding window method.

4.1 Data Related Results

This section reveals information that is found on the data itself. Three chosen regions from one experimental time series, which are used for the whole results section, form the basis of these studies. Within the experimental process, the three regions are located in and around the instability as shown in figure 17. The first region, the yellow one, refers to the *target state* which extends over a long period of time before the data plot eventually starts. While the underlying distinction between the states unstable, instability of type 1 and instability of type 2 is done by thresholds on the amplitude of the oscillation, one can easily see by figure 18, that there is a lot more difference between the instances, even ignoring the amplitude. These differences at different levels are an exciting basis for the machine learning-based investigations in the following sections. The color code for the 3 regions will be kept during the following plots, to enable quick recognition. In the following subsection, the noise-signal relation will be considered, while in the following subsections the temporal correlation and the spatial dimensions in the phase space are considered.

4.1.1 Signal or Noise

The maybe overall most important question for data collected from a detector, is whether the signal is just stochastic noise or actual dynamics. The way this question is approached here, is due to the fourier transform as given by equation 53. Since the frequency spectrum of noise is also just noise, as seen in figure 19, one has a first clue of a signal to noise ratio by determining if there are any significant spectral peaks within the frequency spectrum. In figure 19 is a demonstration of gaussian noise which is mapped in the same normalization (53) and diagram range as the data is in equation (20). For the 3 used snippets of the data one can see that the frequency spectrum has for each region a characteristic distribution of contained frequencies. This means

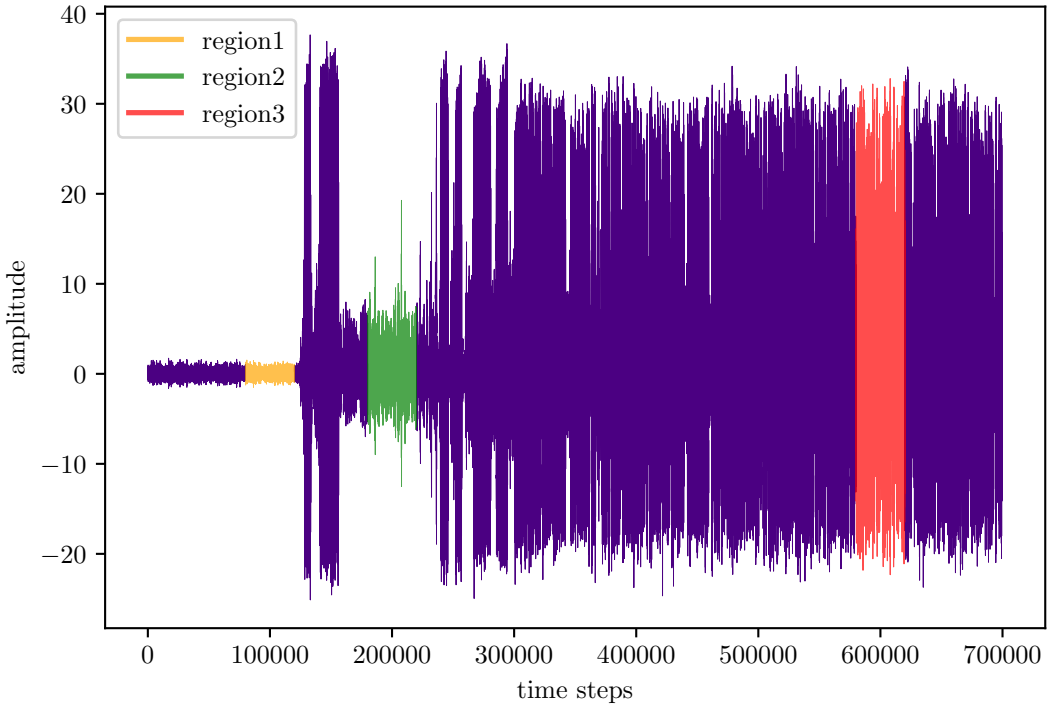


Figure 17: *Data from one pressure sensor of the combustion chamber, with the three data sets labeled, called regions, which are used for the machine learning tests in this Thesis. The stable state is marked in yellow, type 1 instability is green and type 2 instability is red.*

that the signal originates from a physical system and it makes sense to apply machine learning to this measurement of the combustion process in the first place. The known characteristic frequency of 10 kHz [16] within the data also perfectly fits to the appearing peaks found. It is interesting though, that the dominant frequencies do differ, explicitly the frequency of region 3, which has an about 1.16 times higher frequency. But also the higher modes are different from each other. The green power spectra, of region 2, shows a perfect higher mode peak at twice the frequency of the main peak. A third small bump for region 2 is additionally visible at 1.5 ± 0.1 of the main frequency. Characteristic of region 2 is that the frequencies are very sharp. As all frequenyc density distributions have a normed sum to 1, the sharpness is so high that the main frequency reaches up to 0.02, far above the plotted range. So the overall "noise" like frequencies are really low, relative to the amplitude in the original pressure space. Also the red region 3 has a clearly visible peak for double frequency but also has two peaks at about 1.6 ± 0.1 and 2.6 ± 0.1 the main frequency. Only Region 1 has no silhouette at the doubled frequency. But this region does have beside the main peak at 10 kHz a second very sharp peak at 1.47 ± 0.05. So it seems not being related with the secondary peaks of region 3, but is close to small peak of region 2. As a time animated frequency scan has shown though, is the thin yellow peak, just sometimes appearing but then can even tower above the first peak. While region 1 appears to have the highest noise level, has region 3 a very comparatively fuzzy spectra. That might be because of a very unstable

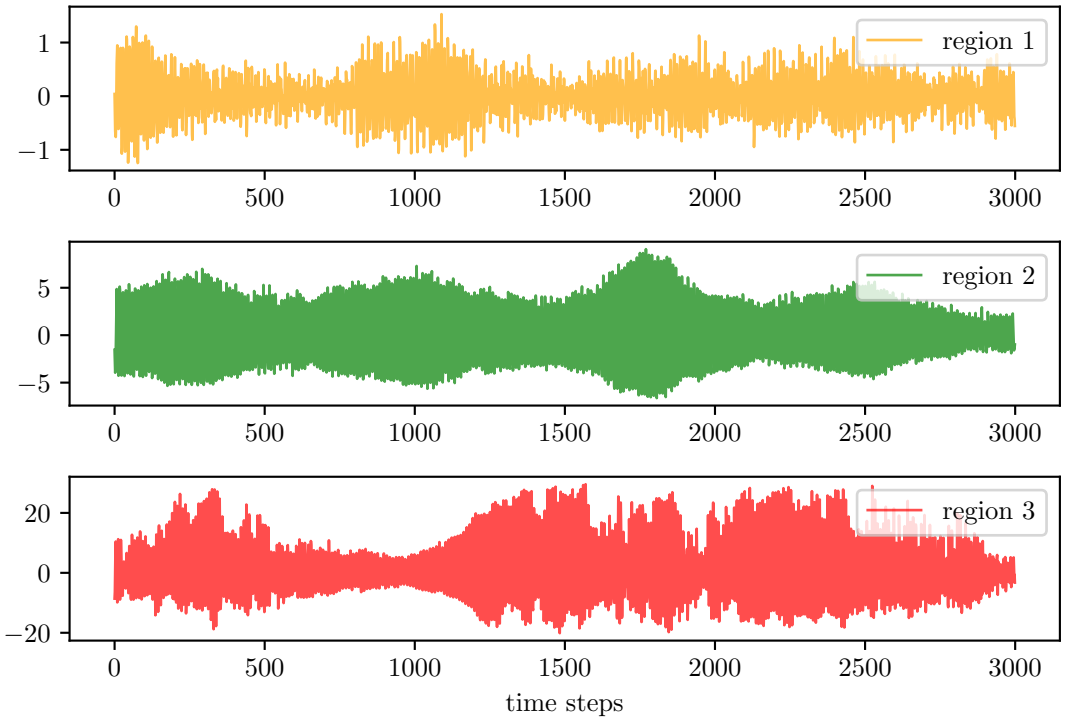


Figure 18: *Those are parts of the three marked regions in figure 17. In this picture, one can see different patterns for the dynamical states also apart from the amplitude range. It should be mentioned that the amplitude scales are different on the y-axis for the three plots.*

dynamical state for the type 2 instability with a lot of small transitions. That goes also with the time animated frequency spectrum, where the frequencies during states similar to region 3 do often shift their main frequencies between the distribution of the green and red spectrum in the figure 20. This is also indicated in the left lower area of the red main peak. The transitions during the type 2 instabilities are also visible in 18 as the oscillation shape changes over time. This effect will blur the quantitative measure later on and must therefore be taken into account. In the next section it will be discussed again a frequency based question, namely the autocorrelation function.

4.1.2 Autocorrelation and Time Delay

In this section the autocorrelation function from 3.1.2 will be applied on the 3 regions of the combustion data in figure 17 to get a suitable time interval for the time delay embedding. For the calculation of the autocorrelation, in each of the three regions where five 3000 time steps long samples were taken, such that it is also possible to draw a standard deviation into plot 21. In this plot one can see again an oscillation pattern. With the strong frequency spectra of the previous section, this pattern validates the causality in the data, because of the statement from the methods chapter, that any sine shaped function will yield a cosine shaped autocorrelation. Also the differences of the three regions are reflected by their autocorrelation. While region 2, with its

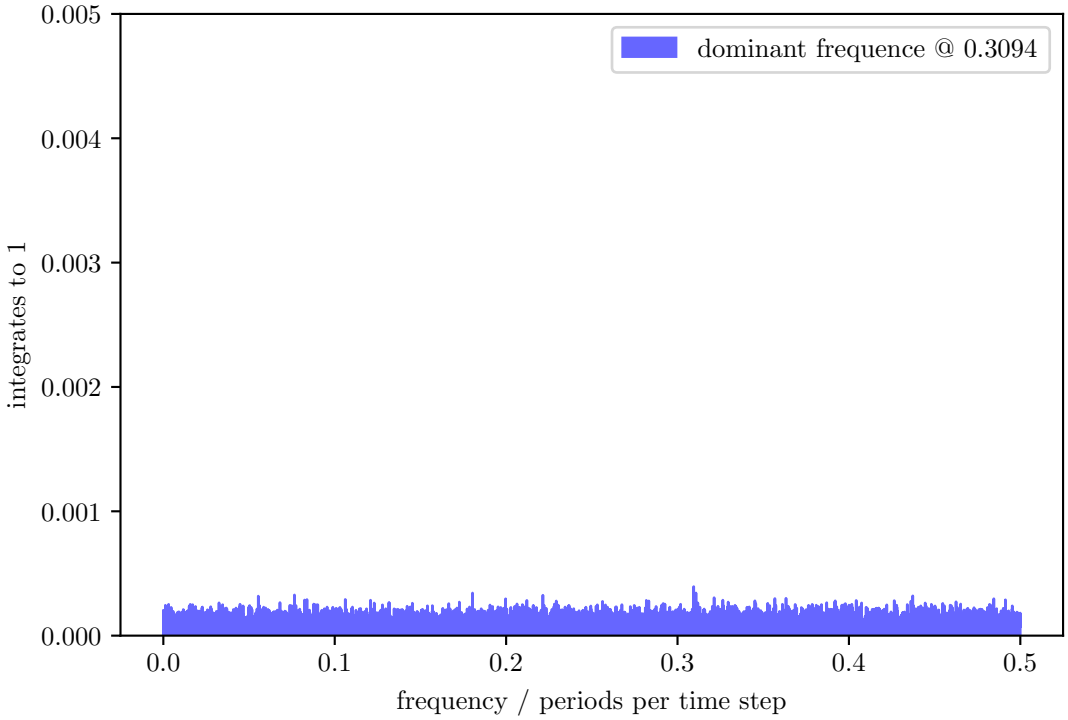


Figure 19: *This is a fourier spectrum of Gaussian noise. The figure is comparable to figure 20, due to the normalization, axis scaling and the usage of the same data length. The y axis values are such, that the frequency distribution integrates to 1. The data length of the noise time series was 20,000 time steps.*

sharp frequency spectra, has a nearly perfect cosine, the correlation of region 1 and 3 are damped over time. It seems that region 3 has a more continuous damping, which would fit the spreaded frequency peaks. On the other hand it appears, that the amplitude maxima for the autocorrelation of region 1 are oscillating by itself, what is a sign for more complex dynamics within the target state. The second region, that means the green graph, has a very thin range for the standard deviation, which again shows the high linear structure for this instability.

By a closer look on the ACF, one sees that the first zero crossing for region 1 is $\tau = 2.0$, for region 2 its $\tau = 2.4$ and for region 3 its $\tau = 2.5$. The figure 21 is kept with a larger range on the delays τ to be compatible with the other autocorrelations shown section 3.1.2 and appendix figure 49b. Since the delay embedding of section 3.1.1 can only be done with integers, the delay for the further work is chosen to be 2 for all regions. This is also consistent with the result in [12]. As practical experience has shown for reservoir computing experiments on the data, there also no big difference in the results for a delay of 2 or 3. And since a higher delay shortens the available data length, the shorter delay is always preferable. The next section will reveal for the delay embedding the necessary embedding dimension m and will discuss the spatial structure of the combustion data.

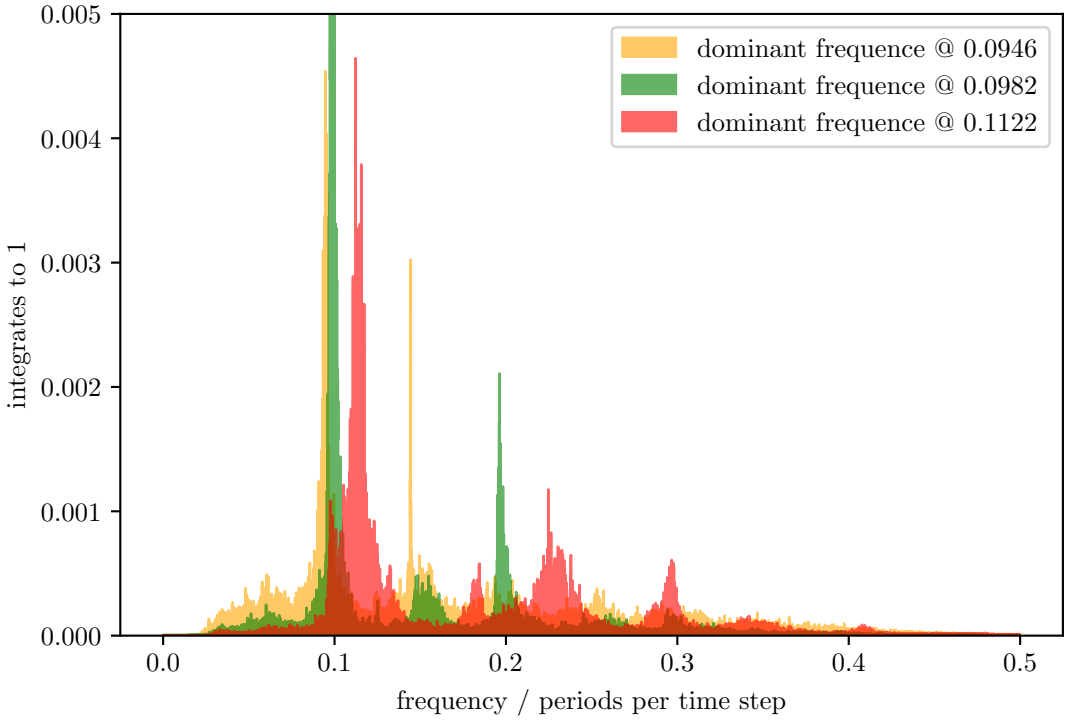


Figure 20: *Frequency density distributions of 3 different dynamic ranges of the combustion data. The y-axis is clipped at 0.005 as the green chart would peak as high as 0.02. The total data length for each data set is 20,000 time steps. The y-axis values are such that the frequency distribution integrates to 1.*

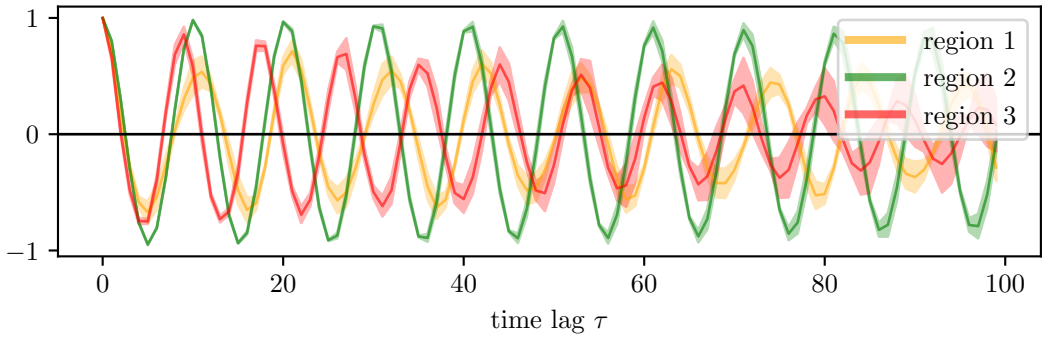


Figure 21: *Autocorrelation function for 3 dynamical states of the combustion data, by using 3,000 time steps long sets. For each of the 3 regions, the mean and standard deviation of 5 samples are plotted. The first zero crossing for region 1 is at $\tau = 2.0$, for region 2 at $\tau = 2.4$ and for region 3 at $\tau = 2.5$.*

4.1.3 Dimensionality

As dimensions are fundamental represents of the geometrical nature of spatial objects, we will analyze the correlation dimension from section 3.3.3, the proper embedding

dimension through the false nearest neighbors algorithm from section 3.1.3 as also the *plateau dimension* [23, 30], which is based on the correlation dimension. This should reveal information about the data and prepare the evaluation of the predictions due to the correlation dimension in the later chapters. For the calculations 3000 time steps are chosen, such that the results are close to the analyzed prediction data sets in the later chapters. All those tests were also performed on 2000 and 6000 time steps, but each result was within the standard deviation of the others. In addition, with real data, the integration volume cannot be increased at will, since different dynamic regimes can be combined unintentionally or the data is simply over. In order to get reliable results, for each region are 5 samples taken as it is also done in Section 4.1.2, in order to get some range of values.

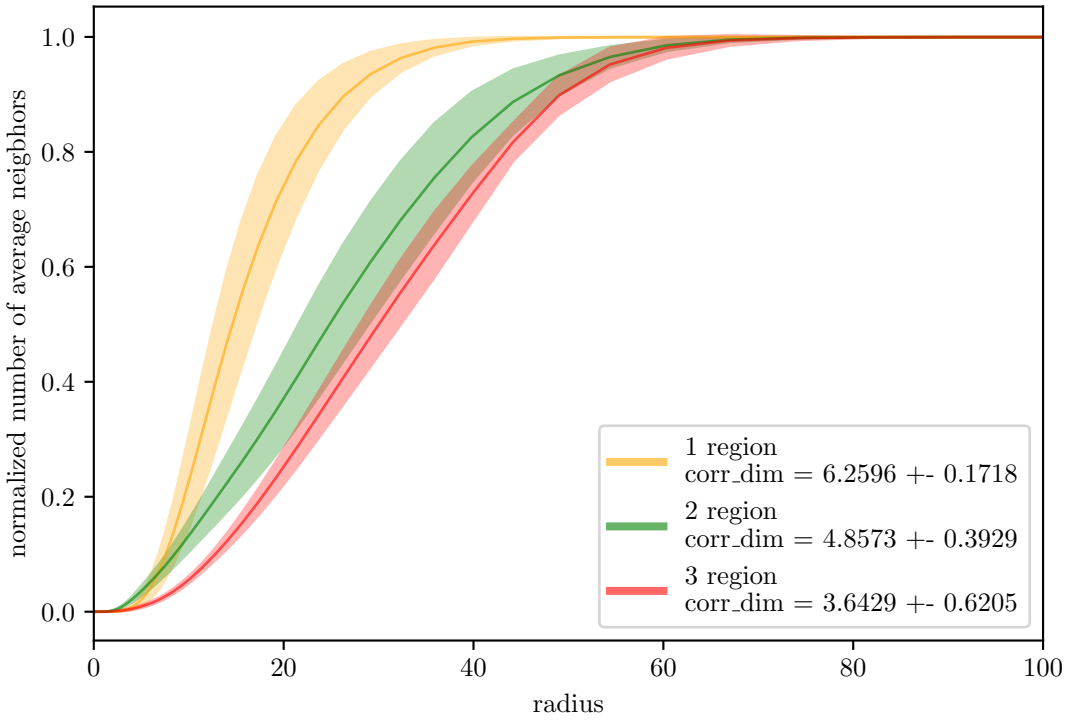
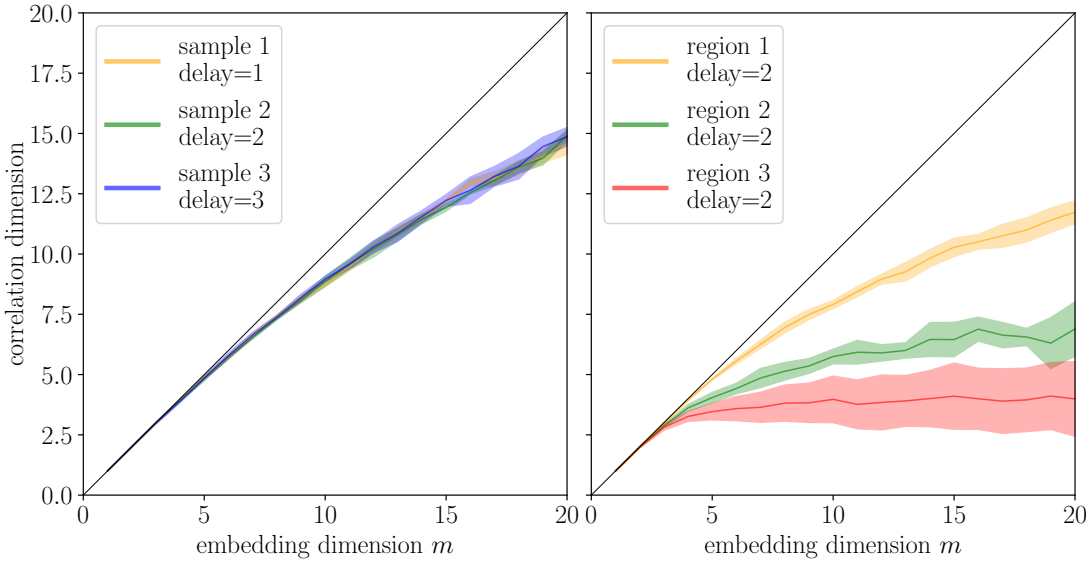


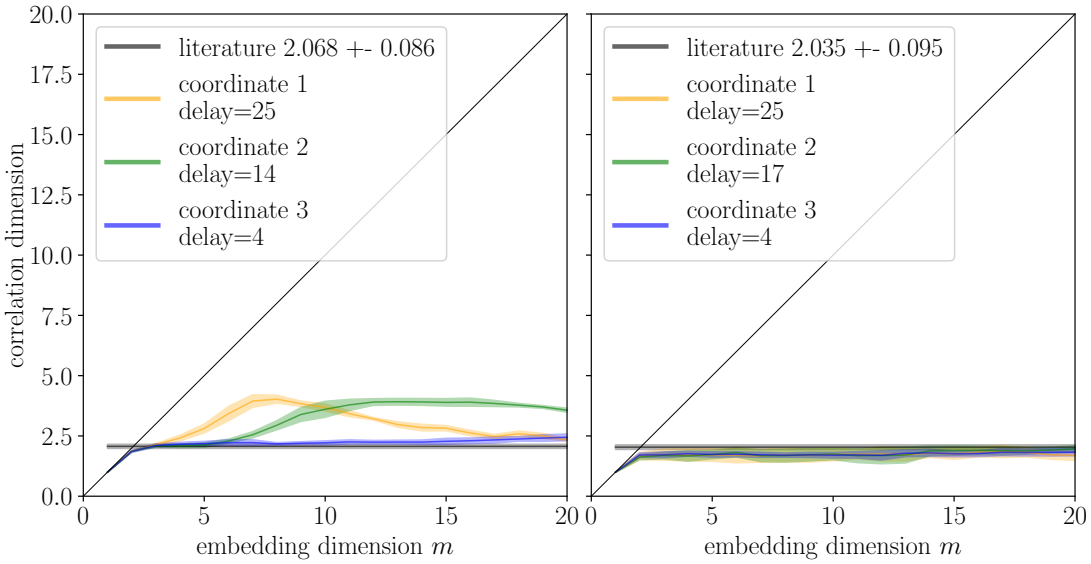
Figure 22: *The correlation sum was determined for 5 samples each with 3,000 time steps. A delay of 2 and an embedding dimension of 7 was used. All 3 regions have completely disjoint correlation dimensions.*

The plateau dimension is a method where some invariant of the attractor [30] or more specially a fractal dimension [23] like correlation dimension in this case, is computed multiple times by using a raising embedding dimension m. So this method also uses only one observational dimension of the data. When the computed invariant stagnates from a certain embedding dimension onwards, this embedding dimension represents a dimension in which the attractor is completely unfolded. In figure 23, the combustion data are compared with Gaussian noise, as it is assumed that the noise contribution in the pressure signal is Gaussian without further information. As the dependence on the embedding dimension is for the combustion data less steep then the one of the noise, it also shows that the signal has a certain geometrical shape in the

phase space, that is well distinguishable from pure noise. But, the region 1 curve is indeed quite close to the noise curve, beside the fourier analysis of figure 20 showed a clear signal within the data. An explanation to this seeming contradiction is that for the calculation of the fourier spectra the deterministic frequencies accumulate constructively, while the noise does not. This effect does not apply to the case of the attractor shape in the embedded space, where the noise contributes in the same way to the point cloud as the signal does. Therefore, a bigger sample size might reveal more information as it is the case for the fourier spectra. Another justification could also be that simply the complexity of region 1 is by far higher than those of region 2 and 3 which are more dominated by a linear structure. In general, an arbitrary raising of the number for the embedding dimension can yield a problem, if the sample size of the data set is not large enough. This is because the correlation integral of (31) refers to the number of neighbors for each data point, which is geometrical affected by the data point being an edge point of the set. For an increasing number of the embedding dimension, the number of edge points within the set rises. This can be understood by imagining the data points embedding in one dimension, i.e. arranged on a line, that only two points are edge points. For an embedding in two dimensions already all points on the border of the data set area are logically edge points and that continues in higher dimensions accordingly.
Ultimately, the results for the plateau dimensions are for region 1 about 12, for region 2 6.2, and for region 3 it is 4.3. Those numbers are subjective though [30], just like the result for the FNN we will talk about next.
The false nearest neightbors proportion in figure 24 does have the biggest decrease of until the embedding dimension of 3, while it further decreases for region 1 and 2 to 4, where the value for all regions is finally under 1%. The FNN algorithm is though sensitive as well as for the threshold introduced in section 3.1.3, as it is also for the threshold in the resulting graph, when one defines there is "no" false neighbors. Eventually the number for the embedding dimension used for the reservoir is 3, since most of the structure unfolded and it is no increase of performance for an embedding with 4 dimensions measurable. The correlation dimension in figure 22 is calculated with embedding dimension of 7, which results in a fractal dimension smaller than the plateau dimension. Nevertheless, the 3 dynamical regions are already very good distinguishable from each other, such that all three dynamical states do have an non-overlapping range in standard deviation, as there correlation dimensions are 6.2 ± 0.2, 4.9 ± 0.4 and 3.6 ± 0.7 for regions 1, 2, 3 calculated with an embedding dimension of 7. This can be seen in figure 22 including the correlation sum. That is the reason why the embedding dimension of 7 is used to calculate the correlation dimension measure also in the machine learning part of the results section, to determine the quality of the prediction. While now some ideas of the complexity of the dynamical system are gained, will the next section look at the development of frequencies over time.



(a) *Gaussian noise* ***(b)*** *combustion data*



(c) *Lorenz63 attractor* ***(d)*** *WINDMI attractor*

Figure 23: *Estimation of the plateau dimension through the dependence of the correlation dimension on the embedding dimension used. The algorithm is applied to Gaussian noise, the combustion data, the Lorenz63 attractor and the WINDMI attractor. For each system and each separate dimension the time delay used is given. The WINDMI attractor applies to all embedding values in the range of the literature value. For the Lorenz63 system this only applies to the third coordinate. The dependence on the embedding dimension for combustion region 1 (orange) is quite close to that of the pure Gaussian noise signal, indicating a weaker signal-to-noise relation than for regions 2 and 3, which converge to a fixed intrinsic correlation dimension much faster. This means that the excited states of the combustion process are less complex. For each curve are 5 samples used, with 3,000 time steps each.*

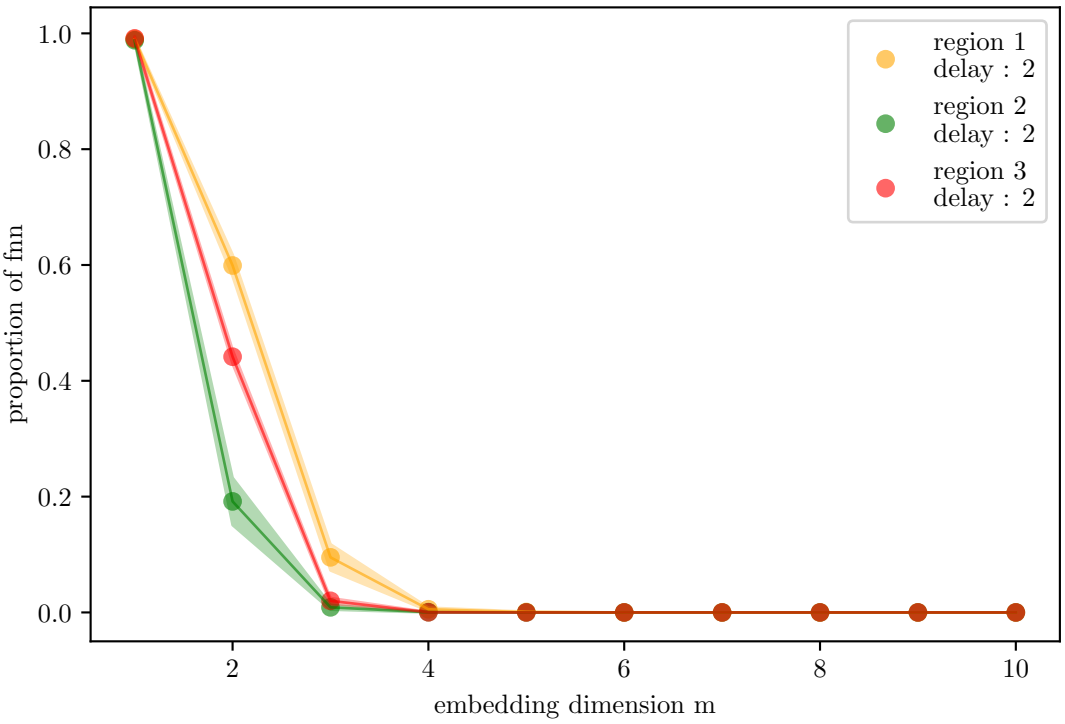


Figure 24: *The result of the FNN algorithm of the 3 individual regions of the combustion data. A threshold of 15 was used and 5 samples each with 3,000 data points.*

4.1.4 Instability Frequency Shift

Exceptionally, several experimental runs of the BKD rocket thrust chamber are considered in this section. Each of them contains a classified type 2 instability such that a sliding window approach for the frequency spectrum, as defined in equation (52), can be applied to see if there is any tendency in the linear part of the system dynamics towards them. The used window length, to apply the FFT algorithm on, is 2000 time steps of the first coordinate of the respective data X. The window was re-evaluated every 200 time steps for the total of 100k time steps. From the received frequency spectrum it is calculated the mean value

$$\text{frequency mean} := \sum_{\nu \in \mathbb{I}} \nu \cdot f_+[X_{\mu=1}](\nu) \tag{56}$$

of the complete frequency interval $\mathbb{I}$ like in figure 20 shown and the maximal value

$$\text{frequency max} := \underset{\nu}{\text{argmax}}(\{f_+[X_{\mu=1}](\nu)\}_{\mathbb{I}}) \tag{57}$$

as the frequency at which the Fourier spectrum has the highest peak. In Figure 25

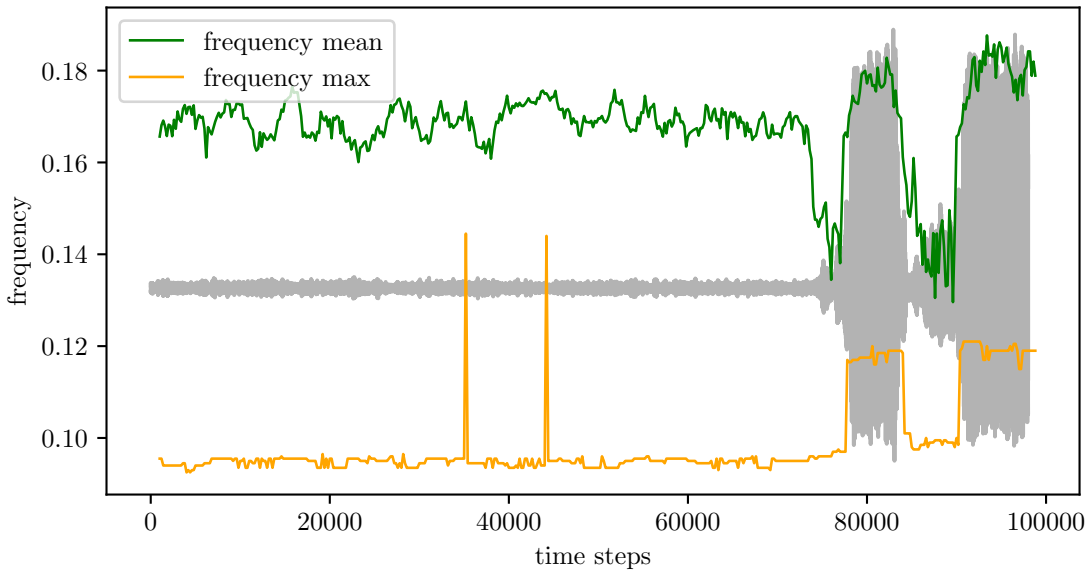


Figure 25: *Combustion pressure signal containing a type 2 instability. The scaling of the gray print signal is the same for this and the following three images, so that the data can be compared one-to-one. For every 200 steps, an FFT is applied to a window with 2000 data points. The mean value of the frequency distribution in Fourier space is plotted in green. The dominant frequency for each time slot is shown in yellow.*

are the default data used in this thesis and as already found in section 4.1.1 has the type 2 instability in region 3 a shifted main frequency compared to the target mode exactly as long as the instability takes place. This can be perfectly seen in this figure by the yellow frequency max graph. Interesting is though, that the type 2 instabilities can be further distinguished in the data. In figure 25 and figure 26 the instabilities do have a large shift in the main frequency as in region 3, while figure 27 and figure 28

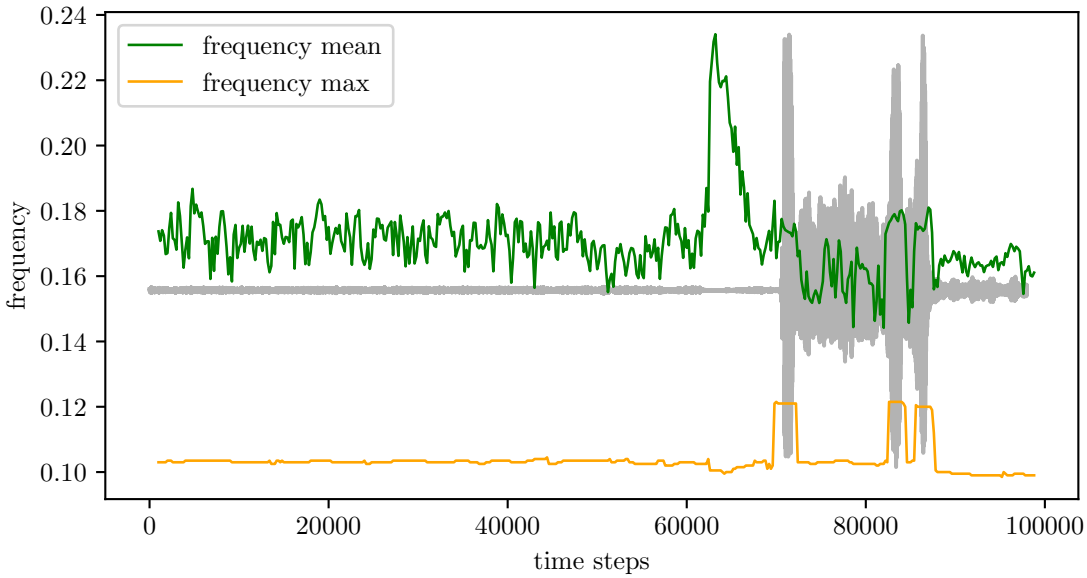


Figure 26: *The same content as for 25, but with different experimental data.*

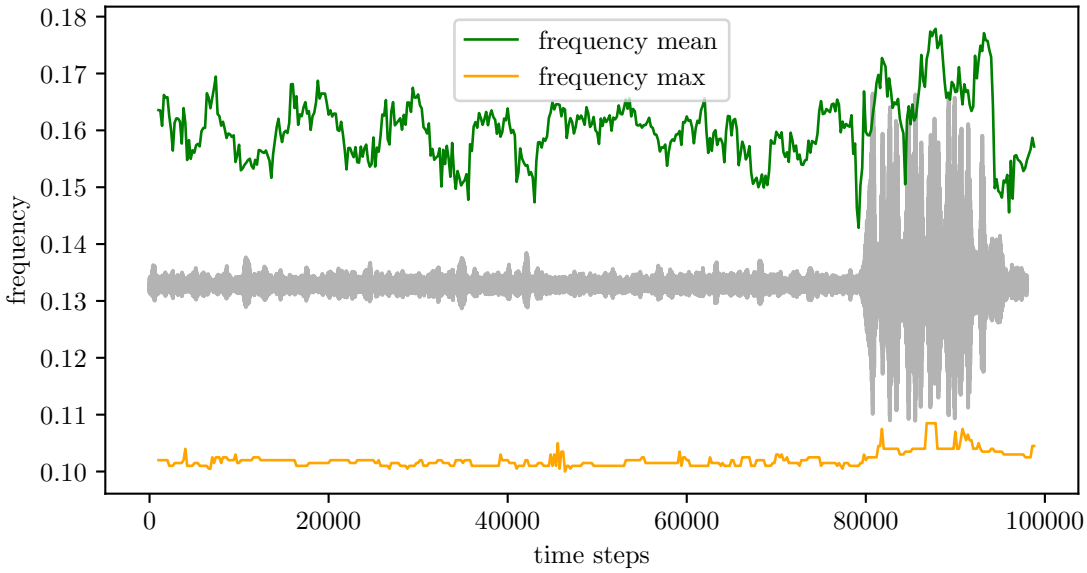


Figure 27: *The same content as for 25, but with different experimental data.*

do have at most a small frequency shift. Furthermore, the amplitudes reached larger for the first two than for the latter ones. At this point it should be mentioned that the gray data plots in the figures are comparable with each other regarding the amplitude. Instabilities of type 1 do not have such a significant difference in frequency than type 2 instabilities. With the mean frequency one has to be more careful in the interpretation. Like in figure 26 with the huge raise in the mean from time step 60,000 onwards. There the pressure oscillation gets a very small amplitude, which can yield a larger proportion of noise, such that it has a higher mean frequency independent from any

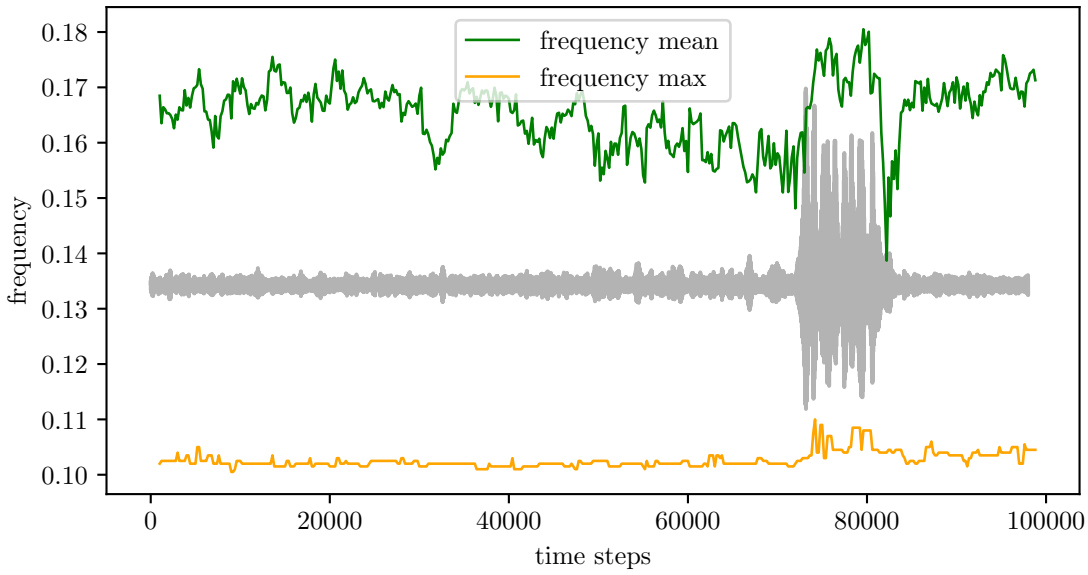


Figure 28: *The same content as for 25, but with different experimental data.*

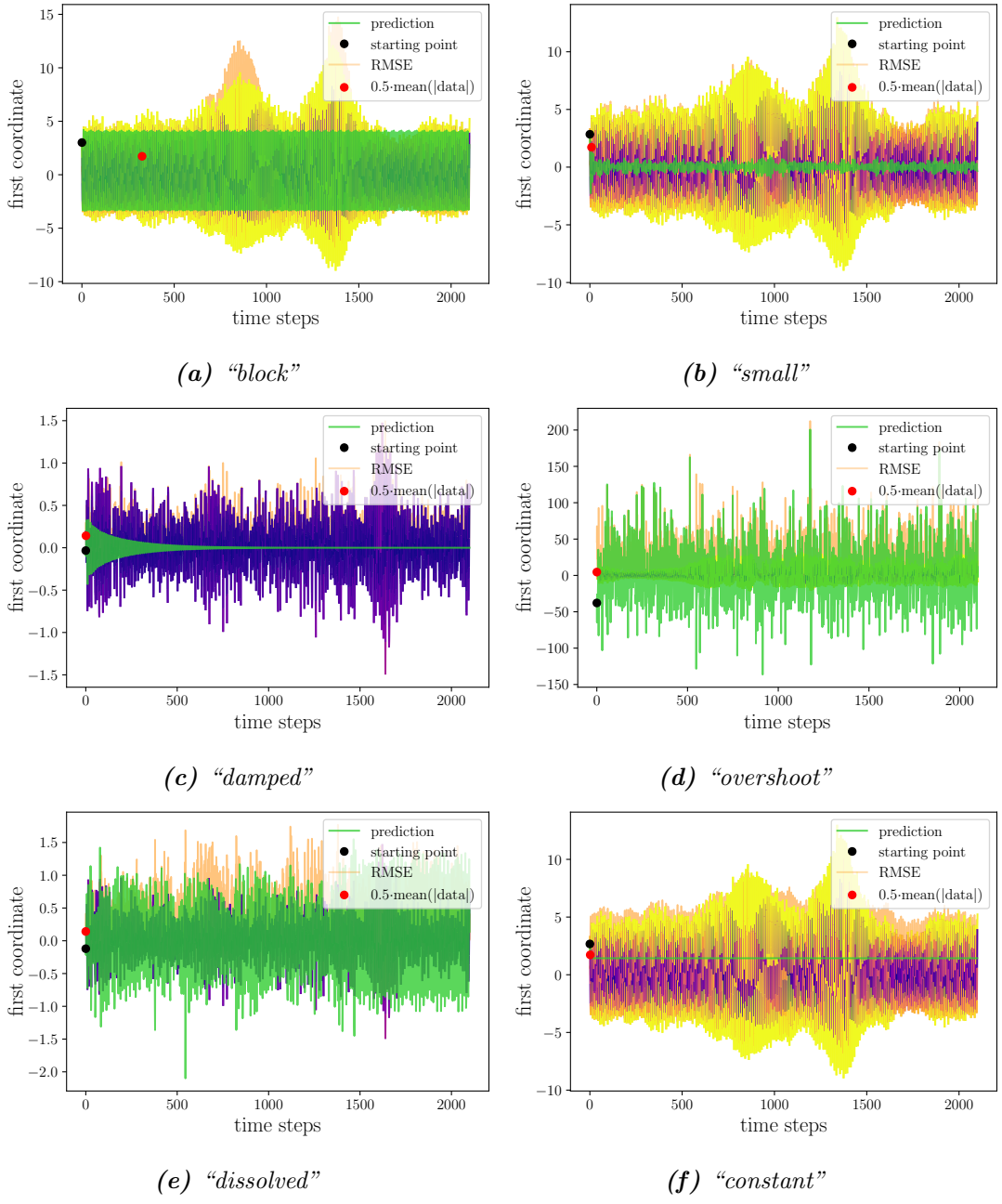
shifts of peaks, as evident in the figures 19 and 20. The short and high peaks in figure 25 for the frequency maximum are due to very short appearances of a second strong harmonic like in figure 20 for region 1. If the spikes are visible or not is dependent on the location and length of the sliding window and step size, since if the second peak is higher than the main one depends on the integration domain.

After analyzing basic properties of the time series data, we finally start in the next Chapter with the actual application of the machine learning method reservoir computing to reproduce the dynamical properties of the thermoacoustic oscillations.

4.2 RC Prediction and Failure on Combustion Data

After analyzing the data, we will now get an impression how good the reservoir computing learns the short period combustion pressure oscillation data. Therefore the three regions of figure 17 are used to state examples. The demonstration is done with chosen hyperparameter optima, which will be discussed in great detail in section 4.3. The explicit results are shown first, because it is considered to be helpful to keep it in mind for the analytical viewpoint. Each sample is a one shot for the chosen hyperparameter combination and is not further selected, such that it is independent of the author prior and not cherry picked. It will be discussed not only the results, where actually the prediction of the RC comes pretty close to the data, but it will also be shown which kind of failures appear during the optimization process for the hyperparameter, since those artifacts can contain valuable insights to the nature of the reservoir itself. Those not working measure optima can be understood as *false friends*, since they are promising but lead to a total fail behavior of the reservoir prediction. First of all, a complete overview for all kinds of failure, found in the training process, should be done, to introduce a vocabulary for the discussion. Therefore in figure 29 are examples for those

classes, which are, and that's all the more fascinating, generated by hyperparameter combinations according to global measure optima. This should also demonstrate, that



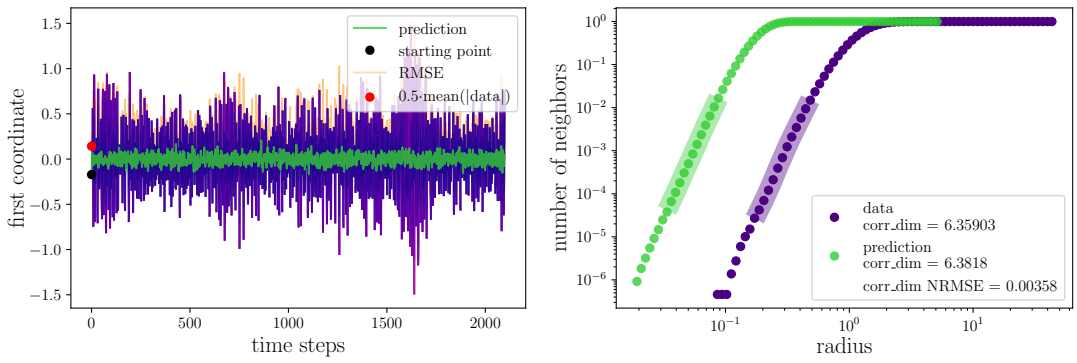
(a) *"block"* ***(b)*** *"small"*

(c) *"damped"* ***(d)*** *"overshoot"*

(e) *"dissolved"* ***(f)*** *"constant"*

Figure 29: *Classification of failures for RC based prediction on combustion data. (a) is an example for a mostly discrete power spectrum producing a static oszillation without further features. (b) has a complex prediction structure besides a very small amplitude. (c) has a fading signal strength. (d) has an amplitude maltiple times higher then the intended one. (e) starts with a prediction structure similar to the test data, but stops doing so at a certain time step. (f) jumps on the first steps to a finite value and remains there.*

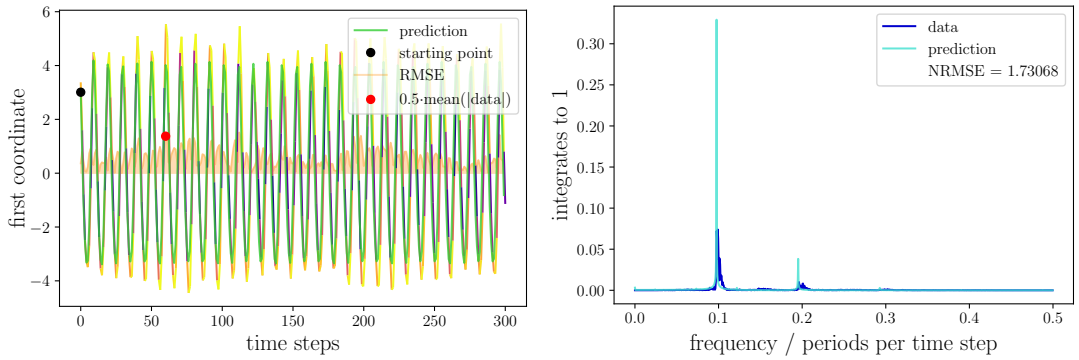
a single measure can be absolutely misleading. For example is the "damped" failure of figure 29c sampled from the optima of the ENTR measure on region 1. As the spectral radius is crucial for the propagation of the "echo" within the reservoir and is therefore responsible for preserving the signal strength, beside the W_{in} matrix, the "damped" error type would be a suspect for a too small value of the spectral radius. Indeed, this error type does arise with small spectral radii, but also the "block" error occurs on the minimal tested spectral radii of 0.3. The "small" error on the other hand does not occur for the tested examples for small spectral radii but rather on large values within a good operational range. What all found "small" situations got in common though, was a regression parameter larger than $50,000$. But the other way around does not hold, as that a large regression parameter leads to the "small" failure. As the "block" failure is by far the most appearing error in general, it also appears for large regression parameters. To get a closer insight how it is possible that one measure peaks at its optima, but all other measures and so the complete prediction is far from sensible, one can see in figure 30. It shows that it is possible that a bad prediction can have a perfect matching correlation dimension. This is the case for a prediction in region 1 where the correlation dimension of the data with 6.35 was contrasted with a prediction correlation dimension of 6.38, but all other measures failed completely. This is maybe due to the fact that the correlation dimension of region 1 is similar to that of gaussian noise. An example for the other way around is, that the linear features get adapted,



(a) *data and prediction* ***(b)*** *correlation sum*

Figure 30: *This is evidence that a good value for the correlation dimension can result from completely different correlation sums. The rapid saturation of the correlation sum for the prediction is a result of the much lower variance of the data in the prediction phase space.*

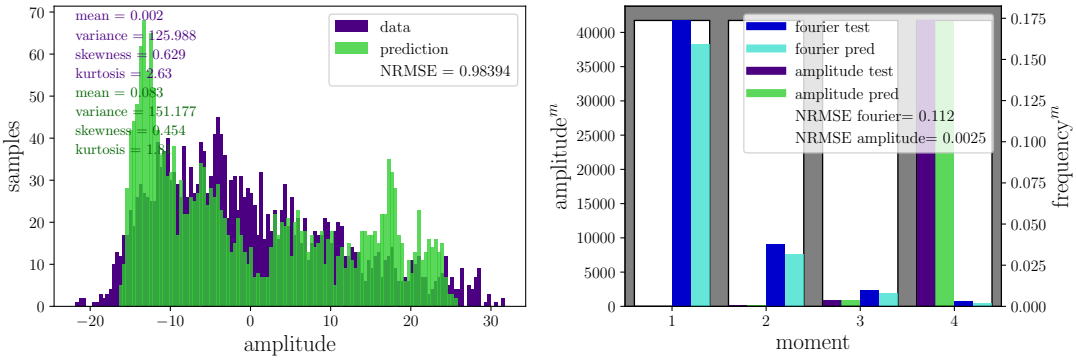
but the complexity of the dynamics isn't learned. That is also the case for figure 31, where a simple adaption to the major frequencies result in a "block" pattern that also results in a very high prediction length because of the stable phase in the pressure oscillation. But it has globally nothing captured from the nonlinearities and topology of the signal. Over all, was the prediction length no good indicator, for a good dynamical reproduction, but quite the opposite. It indicates that the nested amplitude modulation is totally neglected and was mostly prominent in regions of spectral radii smaller 1, which are considered as absolutely not functional. It also exists the case that even to amplitude distribution is very well matched, but with a very poor frequency spectrum.



(a) data and prediction *(b) power spectrum*

Figure 31: *Prediction length, as the most natural measure for forecasting time series, is an example of being a false friend here, as just being in phase with the original data leads to a huge forecast horizon. Since the region 2 data used here has such a strong linear characteristic that a very simple frequency adjustment is sufficient if the deviation threshold is as large as given in the scope of this work. Figure 29a, which shows the same result, shows that the overall quality of the prediction is very poor.*

The optima for RATIO, which is a spectral radius of 3.8 and a regression parameter of 5×10^5, showed very good results for the fourier distribution and for the amplitude distribution, but was at the same time bad on the correlation dimension with a discrepancy of 3.9 to 2.4. That shows that certain properties of the data can fit totally independent of others, as it seems to be the case for linear and non-linear features for example in the case of "block" errors, where it is most apparent. A further subtlety is of course, that the measures are not right adjusted in its sensitivity for the task. As in figure 32 the moments of the amplitude distribution are perfectly matched for the moment 0 to 4 but for the distribution itself its quite a difference recognizable. An interesting note at this point for the amplitude distributions and its moments is that all three regions do have actually very distinctive moment distributions as one can see in figure 50 in the appendix, even though there are all three oscillations with a strong linear background. As a first conclusion, a general dependency check for the occuring types of errors labeled above, could lead to an indicator grid, in which the direction for hyperparameter adaption is revealed. Combined with an identification automatism this could be a basis for a successor of the grid search algorithm for hyperparameter sweeps. However, this needs to be done in more detail, since as it appeared at least for the spectral radius or the regression parameter, different error types can happen for the single value considered alone. Now we start over with the actual successes for the RC predicting the data. As one can see in the figures 33, 34 and 35 that each dynamical state got its characteristics well distinguishable predicted by the RC. It is also good to see, that the RC is able to handle data which is off center and has an amplitude much larger than 1 as can be seen best for region 3. So the data has in a numerical viewpoint not be specially prepared or post processed, as the same RC setup could equally handle all three situations. That also means that the failures may not be attributed to the numerical range of the data. One can see in all three figures mentioned, the 3900 training time steps, the 1000 synchronization time steps and the final 2100 prediction



(a) *amplitude distribution*

(b) *moments of frequence and amplitude distribution*

Figure 32: *The moments $m = 1...4$ fit perfectly for this overall good prediction, while the corresponding amplitude distribution still shows significant differences, which is a sign of too few moments. This might also be useful if the testing phase should place less emphasis on special events and mimic the topology from a more global perspective.*

time steps where all the evaluation is done on. One challenge on real data is, especially for sequential data, that it cannot be guaranteed that all artifacts in the test data are also in the training data and vice versa. This could for example be the case for the second fourier second frequency missing in figure 33b, as this second frequency, which is not a harmonic of the first one, does just occasionally appear. This has revealed a time dependent analysis on the frequency spectrum.

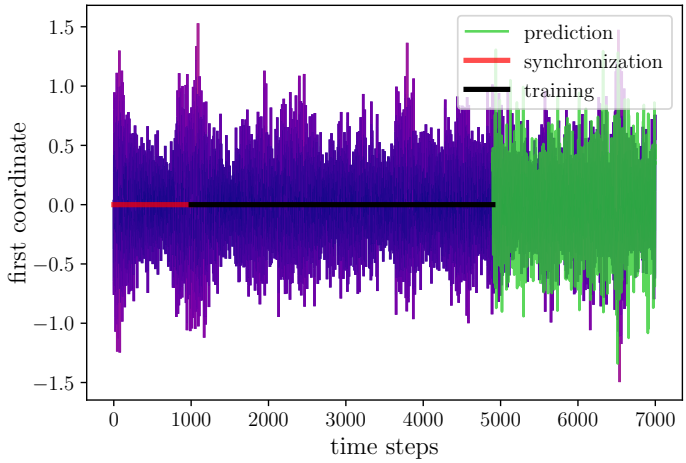
The distribution was was extremely well fitted for the first region. The distribution as also the respective momenta, seen in figure 50, reinforces the statement from section 4.1 that region 1 has a large proportion of noise hence the zero mean and high variance and it additionally indicates that the noise is gaussian. Additionally, has the example of figure 33 an even better fit for the correlation dimension as the mean optimum NRMSE for the correlation dimension, despite it being the optimum for the distribution measure. The spectral radius for the distribution measure optima is larger than the optimal region for the correlation dimension NRMSE suggests.

Regions 2 and 3 have both as the second prominent frequency the harmonic of the first one, which each time detected and learned by the reservoir. It should be said here, that it seems that the reservoir at all is very sensitive to multiples of frequencies, which needs to be further investigated. Also the very characteristic amplitude distributions of both regions got pretty good imitated by the prediction data. The smooth amplitude modulation of the combustion pressure in the instability seen in figure 34a, couldn't been reproduced yet, but it is actually also for the prediction data a modulation of a frequency much lower then the dynamics frequency visible, what is a great success, that the RC can actually learn features of very different time scales.

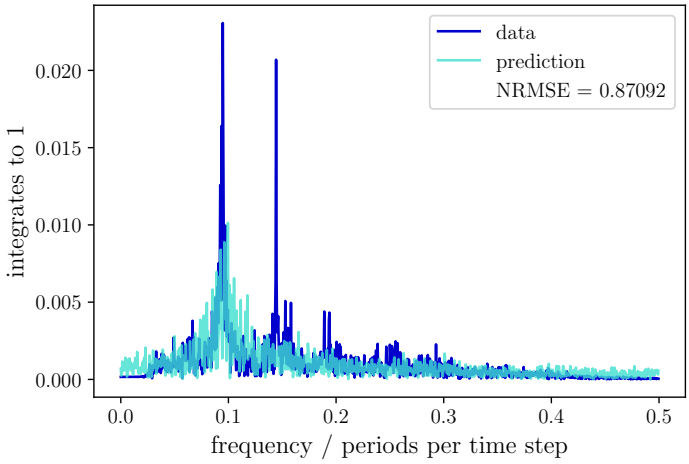
Also the recurrence plots for all 3 regions are very distinguishable and recognizable by the predicted version, as one can see in figure 37. In the case for region 3 there was at the beginning of the test phase at time step 5300 a waist of the amplitude modulation, as it also was on time step 700, which influenced the pattern of the recurrence plot. Despite this feature was in the training data, the reservoir prediction does not contained

a manifestation of it. Since the complete recurrence plot resolution is a 2000×2000 matrix, it is only a section of the data shown in figure 37, to keep the pattern visible. But one has to keep in mind that such unique features influence the numbers representing the global structure, when studying the pure performance numbers.

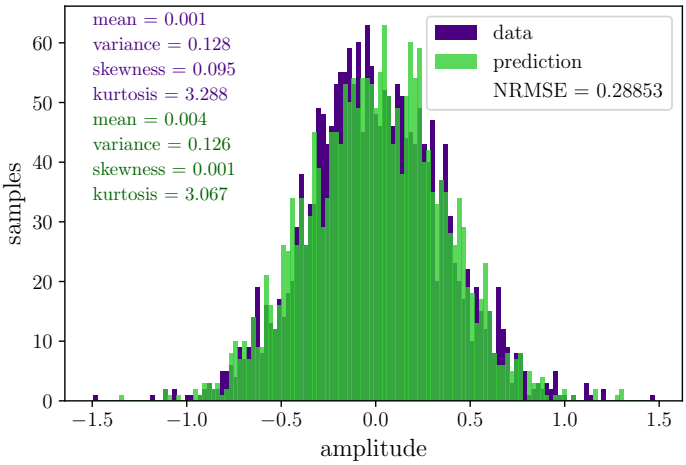
In figure 36 the 2 dimensional delay embedding, with the used delay of 2 is shown for the dynamics of all 3 regions in comparison with the predicted version. The next chapter will contain a more analytical approach to the discussion on the hyperparameter dependencies and will evaluate the utility of the measures itself.



(a) training and prediction data

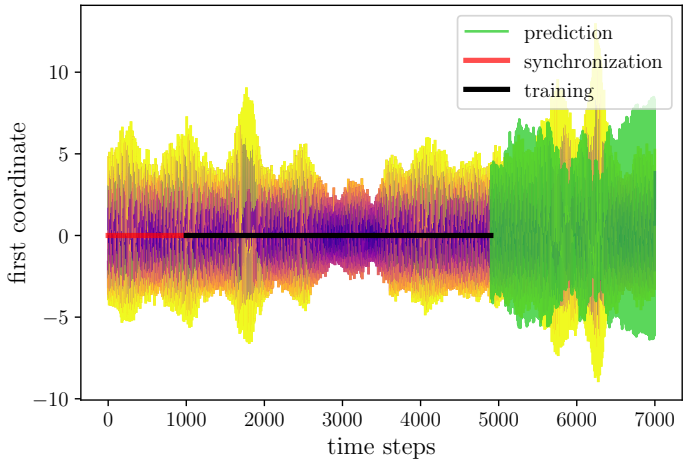


(b) fourier spectrum

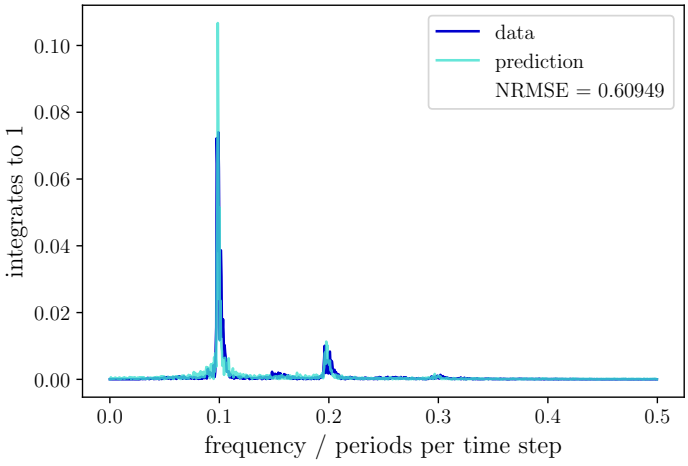


(c) amplitude distribution

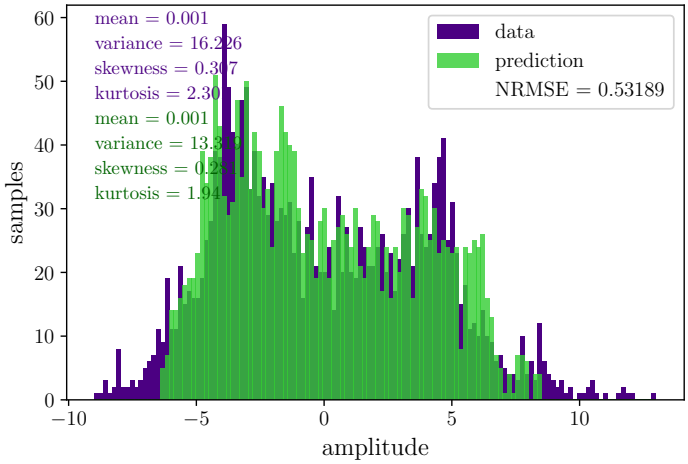
Figure 33: *Example prediction for region 1 based on distribution measure optima with spectral radius 3.5 and regression parameter 100.*



(a) training and prediction data

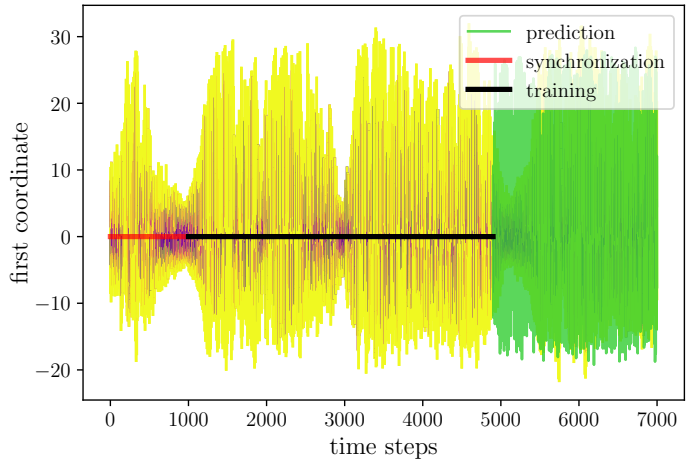


(b) fourier spectrum

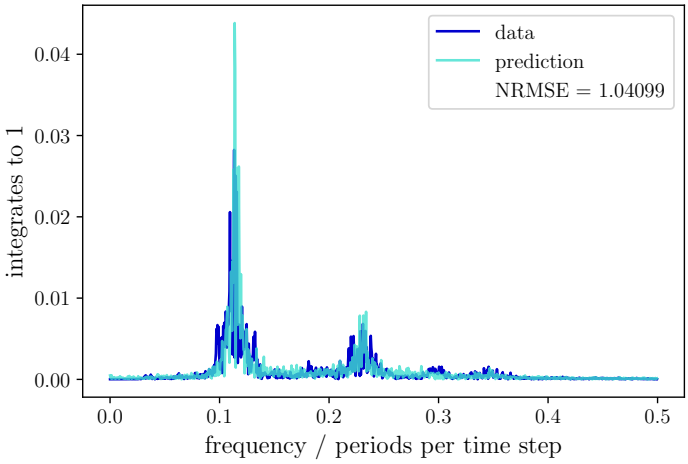


(c) amplitude distribution

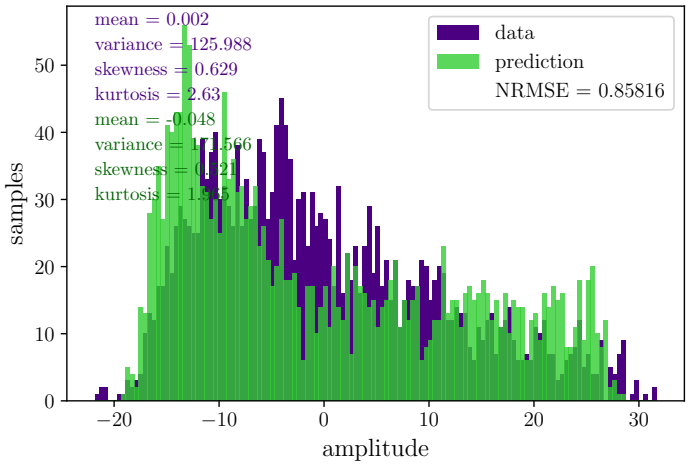
Figure 34: *Example prediction for region 2 based on amplitude distribution moments measure optima with spectral radius 2.6 and regression parameter 100.*



(a) training and prediction data

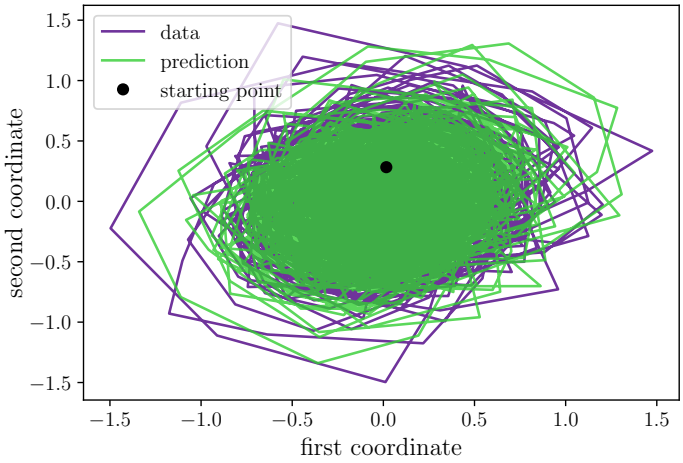


(b) fourier spectrum

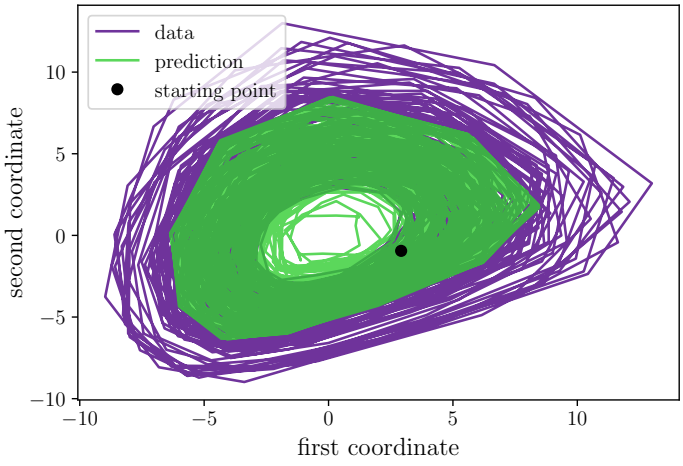


(c) amplitude distribution

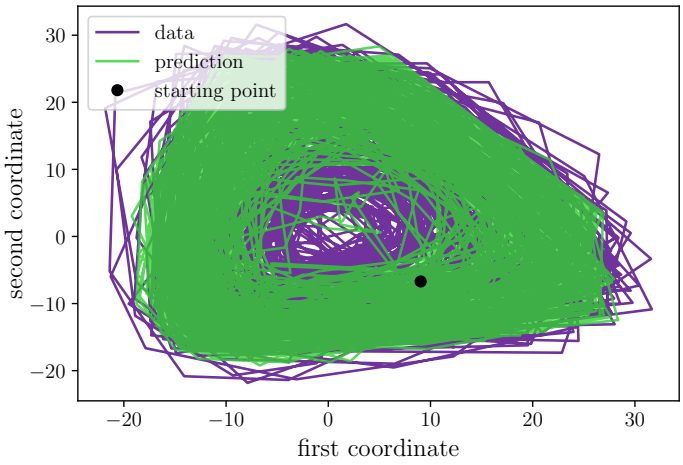
Figure 35: *Example prediction for region 3 based on fourier moments measure optima with spectral radius 3.7 and regression parameter 100.*



(a) *region 1*

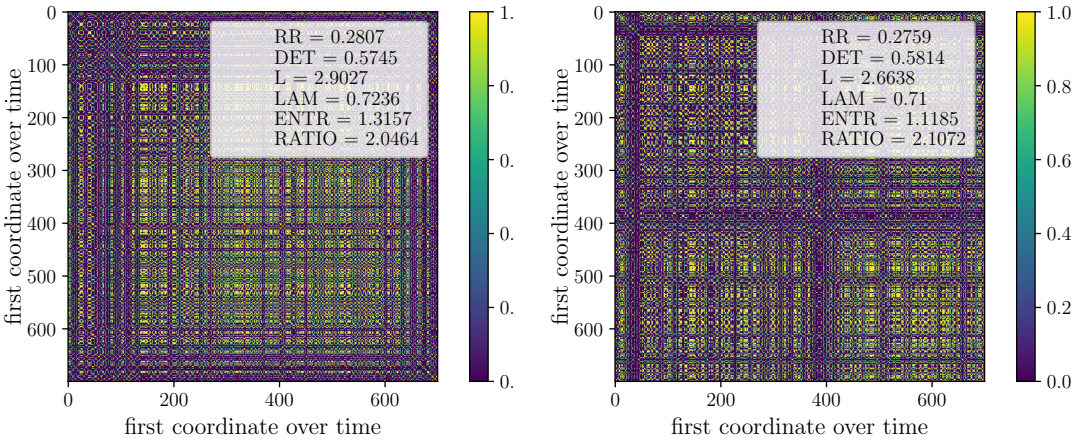


(b) *region 2*



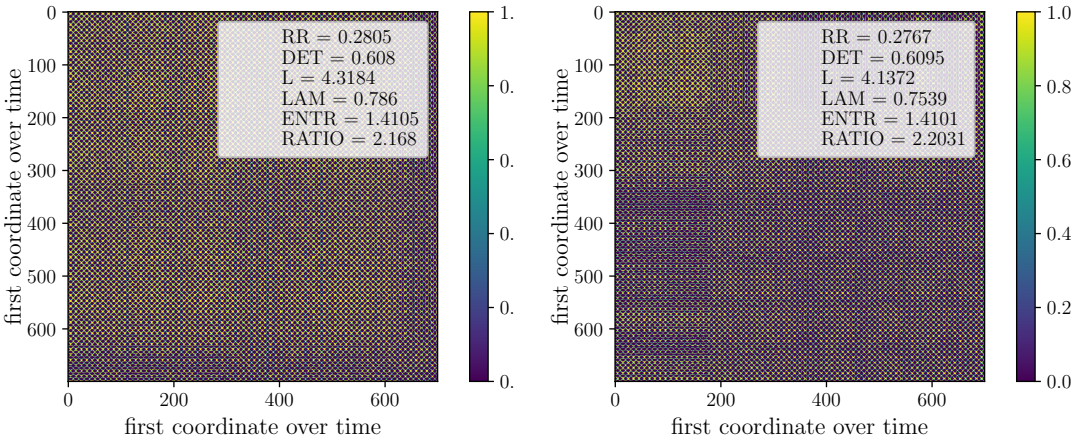
(c) *region 3*

Figure 36: *Two-dimensional delay embedding for the results of Figures 33, 34, and 35. The gap in the middle of regions 2 and 3 shows that the dynamics dominate any noise present in the data.*



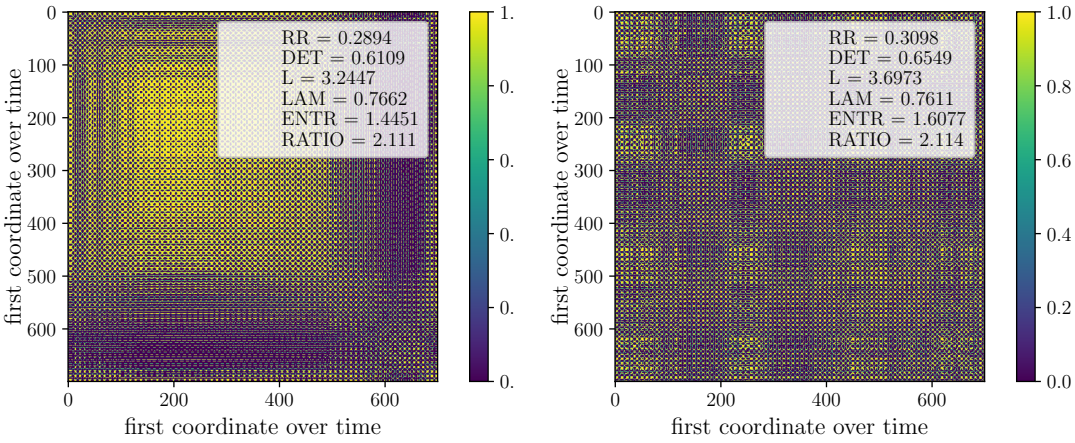
(a) *region 1 data*

(b) *region 1 prediction*



(c) *region 2 data*

(d) *region 2 prediction*



(e) *RP region 3 data*

(f) *region 3 prediction*

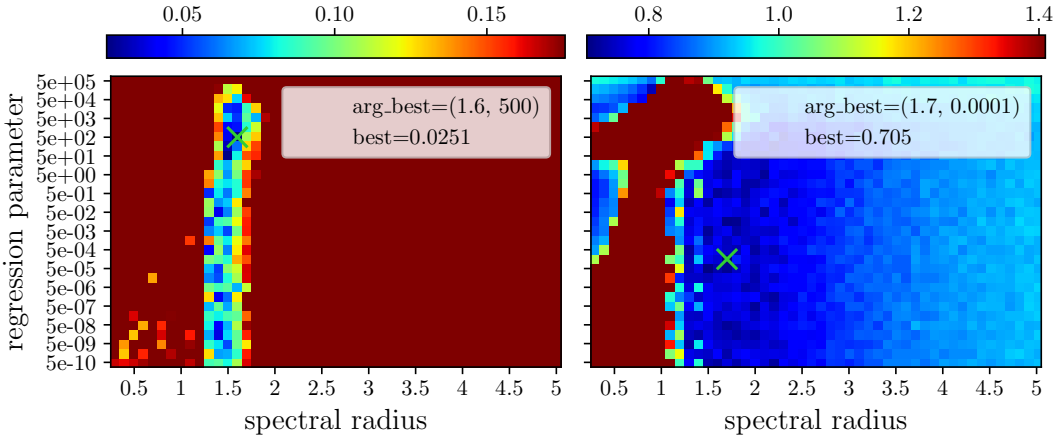
Figure 37: *Recurrence plots of the same examples as demonstrated in the figures 33, 34 and 35. Only the first 700 time steps of the 2000 time steps long predictions are shown. On the left hand side are the test data plots and on the right the related predictions.*

4.3 Measures and Hyperparameters

In this section it is aimed to reveal as well information about ESNs as it is to test measures on a real life example. As hyperparameter dependencies are maybe the most crucial topic for any machine learning system, a bruteforce grid search approach is used to give an explicit insight to what are the best magnitudes for a specific use case of the reservoir. The two chosen hyperparameters for this study are the regression parameter and the spectral radius of the reservoir recurrence matrix A. The spectral radius plays the central role for memory behavior of the reservoir topology [4, 35] since it affects the number of time steps a signal echoes back and contributes to the evolution of the reservoir states $r(t)$. The regression parameter on the other hand stabilizes through the regularization the trajectory generated by RC against noise and improves generalization by reduction of overfitting [46].

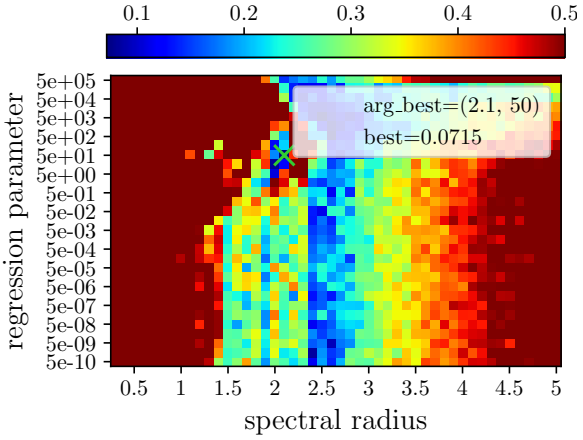
4.3.1 Spectral Radius vs Regression Parameter

It is used a grid-search algorithm to demonstrate the dependencies of the application reservoir computing plus combustion data on the spectral radius and the regression parameter. Therefore 31 logarithmic scaled values for the regression parameter, ranging from 5×10^{-10} to $5\times10^{+5}$ and 48 linear distributed values for the spectral radius, ranging from 0.3 to 5.0, are considered. The training is run on the three regions introduced in figure 17 where 3900 data points are used for the training, 1000 for the synchronization and eventually the evaluation is applied to 2100 prediction and test data time steps. For each pair of spectral radius and regression parameter combination, 7 different reservoirs were trained. That means that the random initialization processes for the input matrix W_{in} and the recurrence matrix A gets 7 different numerical seeds for the sampling process, resulting in 7 different reservoir-topologies, since changing the spectral radius is just a multiplication with a scalar and the change of the regression parameter does not change the reservoir at all. Given that, each pixel in the grid plots 38, 39 and 40 is the mean out of 7 runs. After that, the mean value for each hyperparameter pair out of the 7 realizations, is used in the following colored grid plots, where for all except for the prediction length red values mean a bad error-value and blue is desirable. In those figures it is not the whole numerical range for the achieved measure values mapped to the color range, but values multiple times higher then the optimal value are just uniquely assigned to dark red. The threshold is set to 7 times the optimal reached mean value, except for the NRMSE fourier, which has a factor 2, since this measure reacts more sensitive to hyperparameter change. To keep the notation legible, the number tuples mean (spectral radius, regression parameter) in this section. For figure 38 the NRMSE of the correlation dimension (35) and the NRMSE for the fourier spectrum (53) are compared for all three named data regions. This leads to a comparison of how good the RC is able to learn the nonlinear or linear properties of the data. The second figure, figure 39, contains the comparison of the amplitude distribution error (51) and the error of the RQA measure DET (46) such that it reflects reservoir adaptation to the phase space occupation alongside to the determinism of the phase space trajectory. The third comparison, in figure 40 uses the fourier spectrum moments (55) and the prediction length (30) to reveal the connection between the learning of linear features and the ability of the prediction to stay close to the signal for a long time.

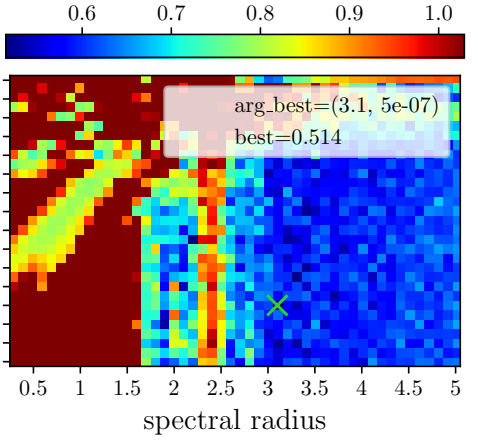


(a) NRMSE correlation dimension R1

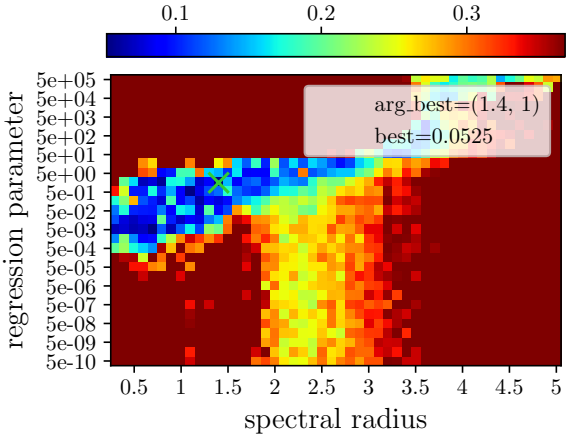
(b) NRMSE fourier spectrum R1



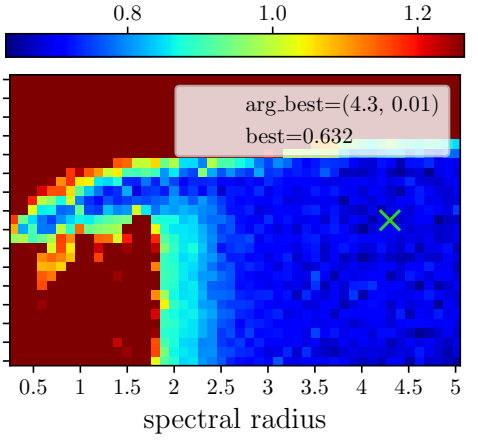
(c) NRMSE correlation dimension R2



(d) NRMSE fourier spectrum R2



(e) NRMSE correlation dimension R3



(f) NRMSE fourier spectrum R3

Figure 38: *Prediction Quality in relation to the correlation dimension and the fourier spectrum depending on the regression parameter and the spectral radius. Each pixel is the mean of 7 runs. The global optimum is given in each plot, with the associated (spectral radius, regression parameters). It is evaluated on the regions R1 to R3.*

A difference of the fourier moments measure to the pure fourier NRMSE is that the moments based measure does ignore the noise-shaped structure of the frequency spectrum as one can see in figure 20 or 33b, since it averages out the noise and weights more the main frequency peaks, as it just uses the $\mathbb{R}_+$ part of the fourier space. But this behavior can also lead to the "block" failure type of section 4.2, since it is prone to a discrete frequency spectrum shape. This on the other hand enables the prediction to stay in phase with the signal, like in figure 31, which results in the misleading optima for the prediction length on spectral radii smaller 1. These elongated areas are one to one reflected in the fourier moments NRMSE on the right hand side, which supports the assumption. It is also tested explicitly for the prediction length optima for region 2 as $(0.6,\ 5 \times 10^4)$ and region 3 with $(0.5,\ 0.1)$. In region 1, the prediction lengths with a maximum mean of 2.43 is assumed to be negligible. This leads to the point that the prediction length measure for the application on the thermoacoustic oscillation data is ironically an indicator for bad learning results. This might be generalizable to all high frequency data with respect to the discrete time steps. The nonlinear structures seem to need spectral radii well above 1.0 to appear in the prediction, which is anti-correlated to the prediction length, at least for the used reservoir setup.

By considering the dependencies on the three dynamical states of the regions by comparing the plots top to down, one sees in the correlation dimension a right shift of the optimal area from 1.5 in region 1 to about 2.5 in region 2. This shift is also reflected in the two fourier and the fourier moments NRMSE, as in the determinism DET. As the data for region 2 has a lower correlation dimension like shown in figure 23b, it could mean that a low spectral radii better matches with a high complexity and strong linear systems on the other hand with high spectral radii. This also holds for the third region, which has also in figure 38e, 38f, 39e and 39f a tendency to higher spectral radii. But for region 3 there is another peculiarity to note, that is, that the optimal region is more horizontally shaped in every measure. This leads to the assumption that for region 3 small regression parameters, less than 1×10^{-4} do not work. As for all measures true, the dependency structure in all plots below a regression parameter of 5×10^{-5} gets a simple vertical pattern, which means that the influence of the regression parameter seems to vanish. In contrast,the upper half of the grid plots shows a rich dynamic on hyperparameter dependency. This matches also the experimental observation, that a regression parameter in the magnitude of 100 does show the most stable results. The large value for the regularization could be explained with a rough calculation in order of magnitudes. Since the ridge estimator (19) consists of basically four different sized quantities, the reservoir output signal $\psi_e(R)$, short R, the target signal $\hat{Y}$, the penalty β and the transformation matrix W_{out}. From equation (15) one can simplify

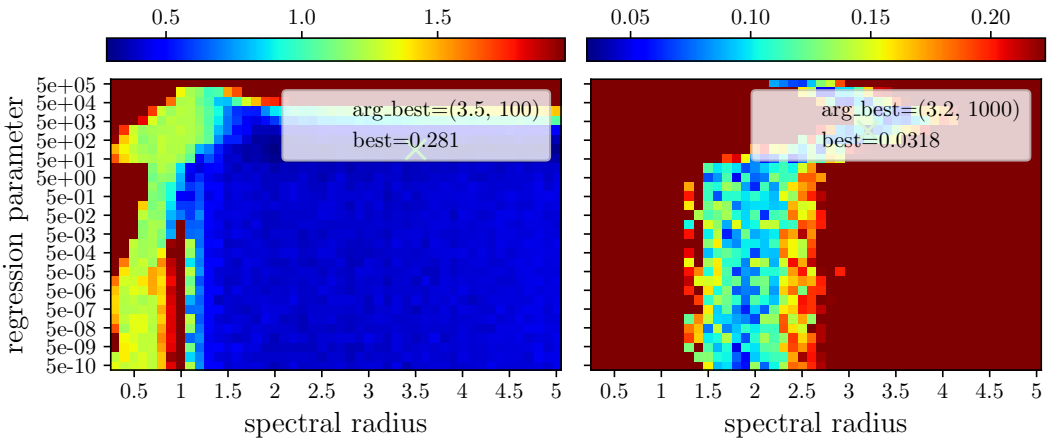
$$\hat{Y} = RW_{\text{out}} \tag{58}$$

such that the relation

$$W_{\text{out}} \overset{!}{\approx} \frac{\hat{Y}}{R}$$

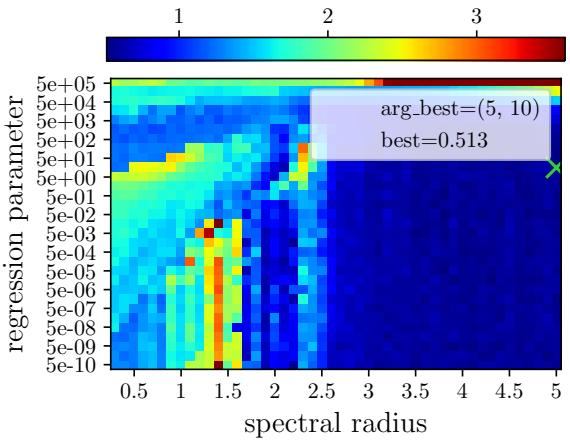
holds. By consideration of the ridge estimator (19) one gets the relation

$$W_{\text{out}} \approx \frac{RY}{R^2 + \beta} \approx \frac{\hat{Y}}{R + \beta/R}$$

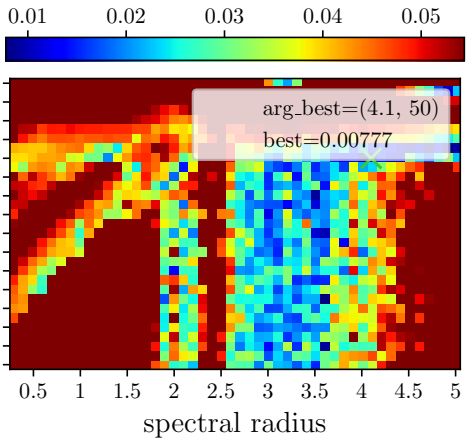


(a) NRMSE amplitude distribution R1

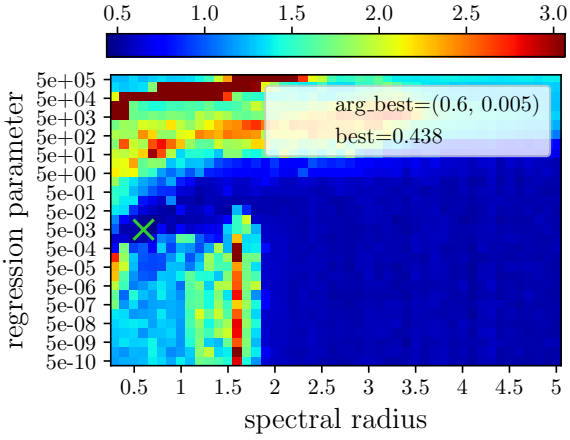
(b) NRMSE DET R1



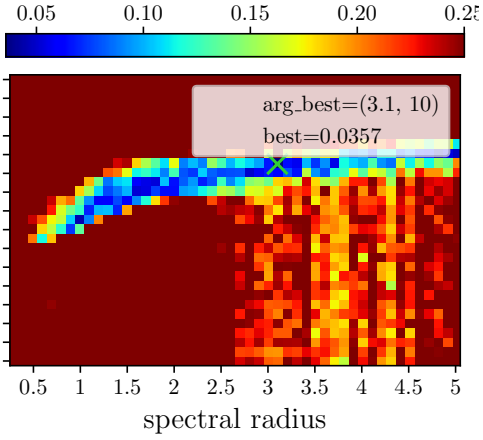
(c) NRMSE amplitude distribution R2



(d) NRMSE DET R2



(e) NRMSE amplitude distribution R3



(f) NRMSE DET R3

Figure 39: *Prediction Quality in relation to the amplitude distribution and the RQA measure DET depending on the regression parameter and the spectral radius. Each pixel is the mean of 7 runs. The global optimum is given in each plot, with the associated (spectral radius, regression parameters). It is evaluated on the regions R1 to R3.*

which shows that the regression parameter β is coupled to the size of R, which is a 2100×3900 dimensional matrix. Another approach is to take into account that $\hat{Y}$ is just a 2100×3 dimensional matrix, such that equation (58) needs a "small" W_{out} and as (19) tells that

$$W_{\text{out}} \propto \frac{1}{\beta}$$

such that the regression parameter β needs to be large for a fitting from a high dimensional space R to a low dimensional space $\hat{Y}$, whats is the well known risk for overfitting.

Another interesting artifact in the grid plots is, that for some measures, for example the fourier spectrum and the DET NRMSE as well for the amplitude distribution NRMSE there is on region 2 a negative maxima at a spectral parameter of 2.4 surrounded by two minima. This is also the case for the LAM NRMSE in the appendix. In contrast, this is exactly the position where the correlation dimension has its best values.

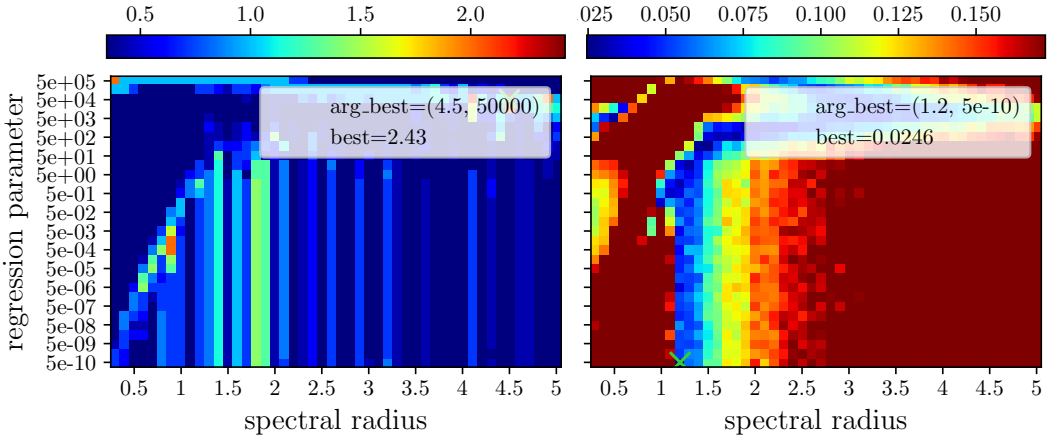
By analyzing the pattern it is clearly evident that the RQA measure DET shares a lot of the pattern with the two fourier measures. For region 2 and 3 the pure fourier NRMSE is more similar to DET while for region 1 the fourier moments measure closer to DET. This could be because of the higher noise proportion in the fourier spectrum for region 1, which gets suppressed by the moments method. Since the data is dominated by oscillations in the phase space (see figure 36), close trajectories stay very close to each other for well defined frequencies. This can be the reason, as discussed in section 3.3.4 that DET and Fourier do have for all 3 regions almost the same pattern.

LAM, ENTR, RR and of course the prediction length are those candidates to be most prone to having false friends as global optima. This can of course be sometimes explained with local artifacts, which are not part of the appropriate range for the hyperparameters, but it is also a sign that those quantities do have an numerical unstable behavior as a performance measure.

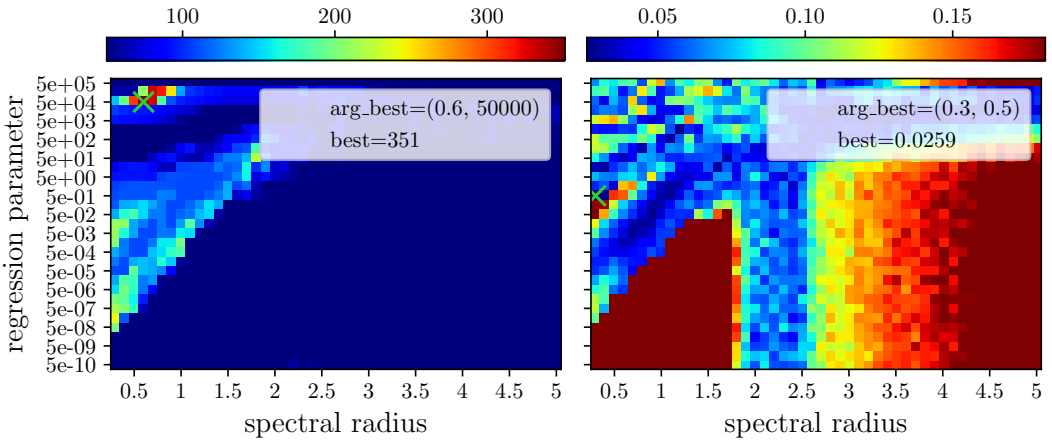
Furthermore, it is unclear yet, why the right bottom area with high spectral radii and low regularization, shows for several measures a low error. This is true for the fourier NRMSE and amplitude distribution NRMSE, as well for some RQA measures like the RATIO NRMSE. It seems that from a certain spectral radius there is no more regularization required. What can be confirmed is that those areas are no false friends but actually work.

However, the same cannot be said for the blue tails at the very top of some grid plots. There is no empirical evidence that these are valid settings. So for example (1.5, 1×10^4) from the correlation dimension of region 1 or (2.4, 5×10^5) from the DET measure on region 1, (2.4, 1×10^4) for region 2 and (3.9, 1×10^4) for the correlation dimensions, are no good hyperparameter in practice.

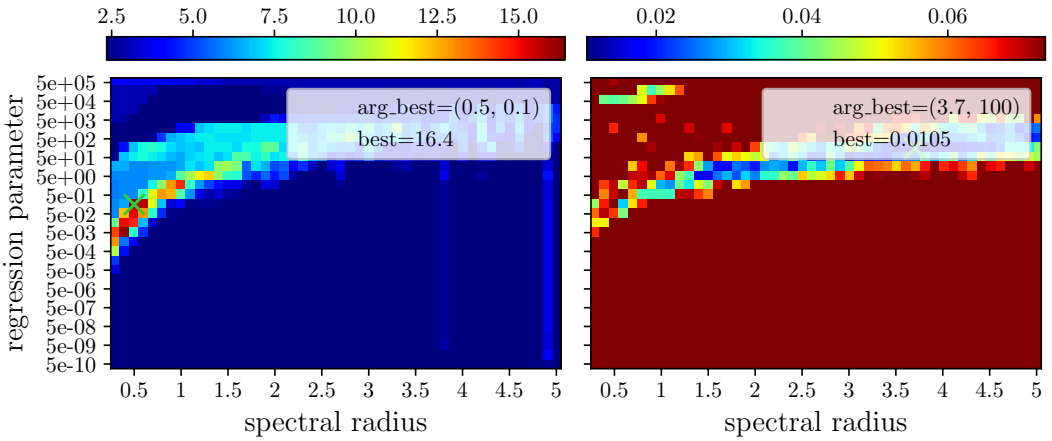
Summarizing for the hyperparameter, one can state that all spectral radii smaller one are shown to be false friends. In contrast, spectral radii up to 5.0 are possible and not as harmful to the performance as to choose a too small value. For the regression parameter, there is the limit on the side of high values. Namly going beyond 10×10^4 starts to drastically lower the performance. That means the left and the top region of the grid plots are bound by a sharp performance decrease. For the right and bottom there is not such a hard edge, but there is no need to test for spectral



(a) prediction length R1 *(b) NRMSE fourier spectrum moments R1*



(c) prediction length R2 *(d) NRMSE fourier spectrum moments R2*



(e) prediction length R3 *(f) NRMSE fourier spectrum moments R3*

Figure 40: *Prediction Quality in relation to the prediction length and the fourier spectrum moments depending on the regression parameter and the spectral radius. Each pixel is the mean of 7 runs. The global optimum is given in each plot, with the associated (spectral radius, regression parameters). It is evaluated on the regions R1 to R3.*

radii above 5.0 since the performance change is stagnant or slowly decreasing and the regression parameter has lower than 5×10^{-8} a vanishing influence. Overall the tuple (3, 100) shows a very stable performance for all three regions and is the best starting point for optimization. Both are unusually high values and should encourage to look for hyperparameter settings on a broader range for reservoir computing. Further comparisons of grid plots for the measures can be found in the appendix, while in the next section a quantitative comparison on the measures is made.

4.3.2 Correlation of the Measures

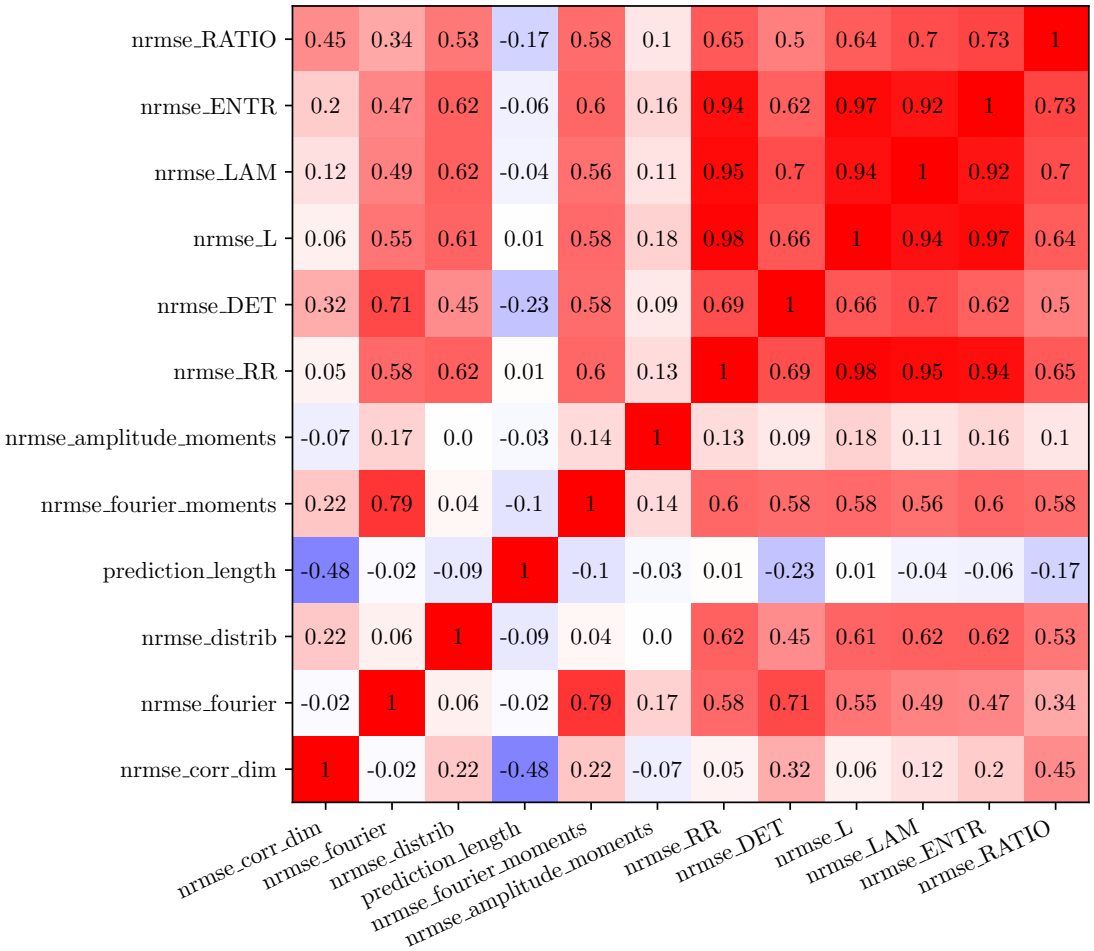


Figure 41: *Correlation of the two-dimensional grid search diagrams with each other. Red means high correlation, white means no correlation, and blue means anti-correlation. This is done for region 1.*

While in the last section already a visual evaluation of the hyperparameters was done for the different objectives, we will now look for the concrete correlations of the independent measures. The correlations between each of the measures is independently

plotted for the three regions in figures 41, 42 and 43. Where the correlation ranges from -1 to $+1$ as usual and is colored dark blue for a strong anticorrelated relation, white for no correlation and red if the dependencies of the measures on the two hyperparameters are correlated with each other. The correlation is defined as

$$Corr(X,Y) := \frac{Cov(X,Y)}{\sqrt{\mathbb{V}(X) \cdot \mathbb{V}(Y)}} \tag{59}$$

where the covariance $Cov(\cdot,\cdot)$ as the nonlinearity term for a group of random variables, e.g. necessary but not sufficient dependent variables, is defined as

$$\mathbb{V}(X+Y) = \mathbb{V}(X)+\mathbb{V}(Y)+2Cov(X,Y) = \mathbb{V}(X)+\mathbb{V}(Y)+2\mathbb{E}[X \cdot Y]-2\mathbb{E}[X]\cdot\mathbb{E}[Y] \tag{60}$$

such that in application to the measure grids from the last section 4.3.1 the result for the two hyperparameter p_1 and p_2 and the measures M_1 and M_2 are computed as

$$Corr(M_1, M_2) = \frac{\mathbb{E}_{(p_1,p_2)}[M_1 \cdot M_2] - \mathbb{E}_{(p_1,p_2)}[M_1] \cdot \mathbb{E}_{(p_1,p_2)}[M_2]}{\sqrt{\mathbb{V}_{(p_1,p_2)}[M_1] \cdot \mathbb{V}_{(p_1,p_2)}[M_2]}} \tag{61}$$

with the mean value

$$\mathbb{E}_{(p_1,p_2)}[M] = \frac{1}{\#(p_1,p_2)} \sum_{(p_1,p_2)\in grid}^{\#(p_1,p_2)} M(p_1,p_2) \tag{62}$$

such that the varianz on the measure M gets

$$\mathbb{V}_{(p_1,p_2)}[M] = \mathbb{E}_{(p_1,p_2)}[(M(p_1,p_2) - \mathbb{E}_{(p_1,p_2)}[M])^2] \tag{63}$$

where each (p_1, p_2) entry stands for the (spectral radius, regression parameter) tuple and $M(p_1, p_2)$ is the average out of 7 runs performed on this combination as described in section 4.3.1. As one can easily see in the bold red upper right quarter for the correlation figures, the RQA based measures do correlate with each other, it is plausible since all those measures are based on the same matrix and depend on the same recurrence threshold parameter. It is very interesting that the determinism measure DET in region 1 has the weakest correlation with the other RAQ measures. This reflects the fact that in region 1 is indeed the highest complexity and/or noise, such that the predictability and hence determinism is low. For region 2 and 3 the laminarity LAM has a vanishingly small correlation with the other RQA measures, despite the fact that all measures are sharing the same computational foundation, namely the recurrence plot. The reason for that is likely that the strong linear oscillations in those two regions prevent laminar states as described in section 3.3.4. Since region 1 has a weaker linear expression than the other regions, the LAM measure is also more correlated to the rest of the RQA measures, which goes with the overall more complex noise like dynamic in region 1. As already found in the pattern of section 4.3.1, is the correlation between the two fourier measures and the RQA measure stronger as the correlation between the correlation dimension and the RQAs. For that, the correlation dimension is the third least correlated measure out of the 12 measures and hence very independent. The two more uncorrelated measures are the prediction length and the amplitude distribution moments. It is astonishing that the amplitude distribution moments are so uncorrelated

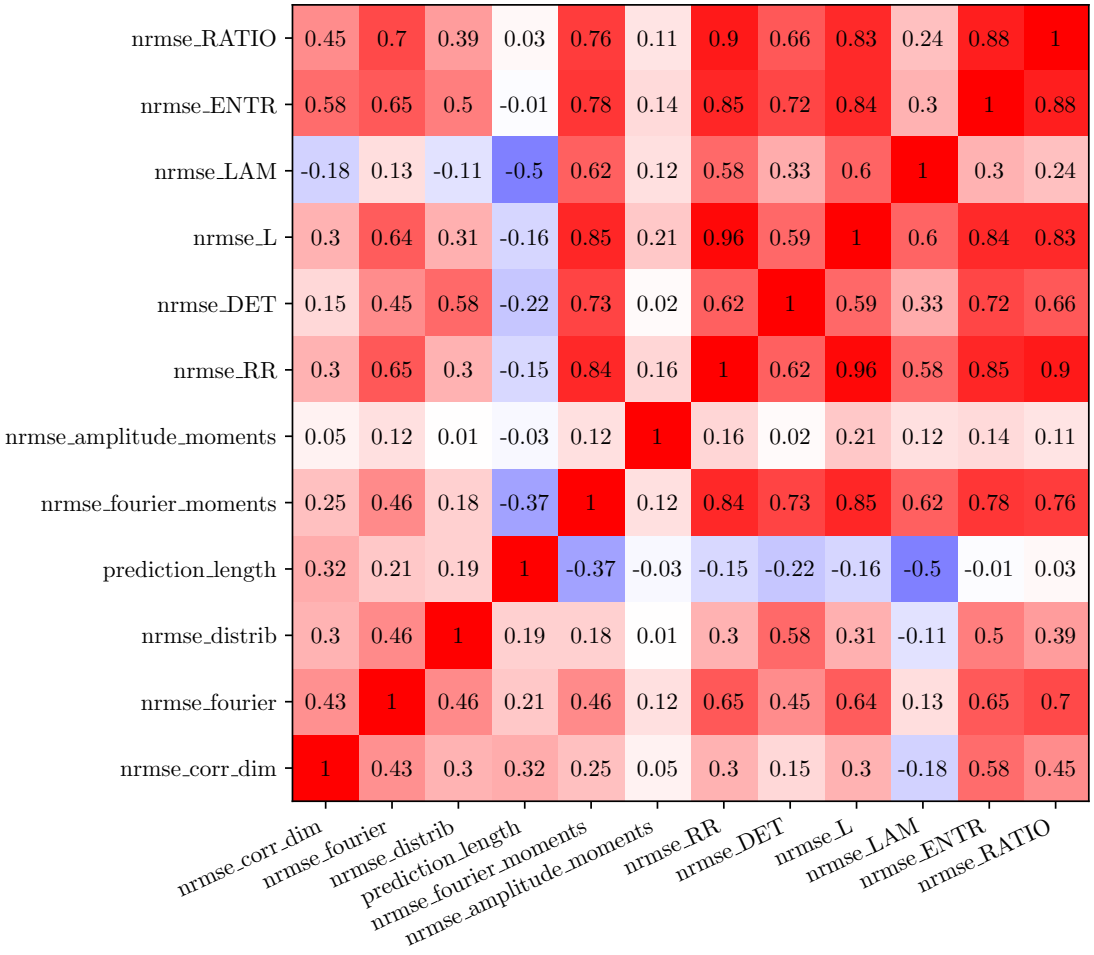


Figure 42: *Correlation of the two-dimensional grid search diagrams with each other. Red means high correlation, white means no correlation, and blue means anti-correlation. This is done for region 2.*

to the others, since it has actually very good optima and had the least false friends rate for all measures. Even more surprising is the fact that it is not even correlated with the amplitude distribution measure itself. The reason might be that the correlations are computed for the whole grid range, such that not only the location of optima but also the numerical behavior for the regions of large errors play into the overall correlation, which can lead to different results as for a visual evaluation. It should also be stated one more that the grid plots in section 4.3.1 do have a limited color range close to the optimal value. The prediction length, as it is also not or even anticorrelated with the other measures, was already discussed, as it is indeed not useful in any point of view for the hyperparameter optimization in the case of the thermoacoustic oscillation data. The relative low correlation between the correlation dimension, as measure for complexity in the dynamical structure and the fourier measures, can be reasoned with

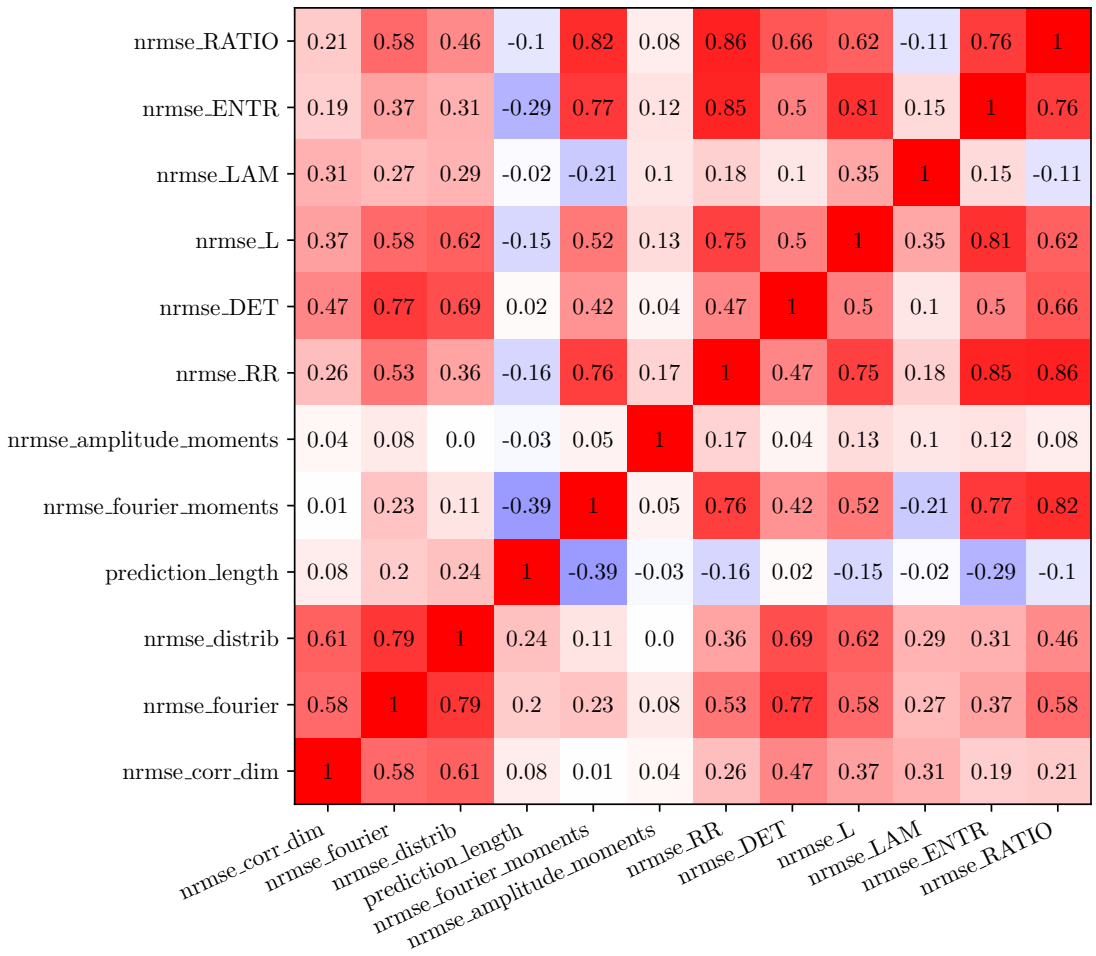


Figure 43: *Correlation of the two-dimensional grid search diagrams with each other. Red means high correlation, white means no correlation, and blue means anti-correlation. This is done for region 3.*

the fact that there are actually nonlinearities in the thermoacoustic oscillations as the difference between those two measures also shrinks in the two more linear regions 2 and 3, which are the two instability states. After analyzing the different measures itself and searching for a good hyperparameter setup, the next section 4.4 will use this knowledge to scan the pressure oscillation evolution of the combustion process over time.

4.4 Time Evolution of Data and Prediction

The research motivation to this section is to determine, if the RC can reveal any interesting information about the dynamical state of the combustion process, due to the oscillating pressure signal. Therefore, it is used a sliding window of the length of 2100 time steps, which allows to compare the quantities introduced in section 3.3,

between the data and the prediction data of the RC. As shown in figure 44, does the reservoir have a lead-time of 4900 time steps, to adapt to the system state, with synchronization and training. Furthermore, it should be introduced to the vocabulary *jump* and *step*. While a step refers to the usual time step t for the data propagation in the RC, jump means the shift of the whole framework of figure 44. As also shown in the figure, is that for the further figures in this chapter, the dynamical state is drawn in the middle of the evaluation window. The jump size could be chosen such that each prediction window starts at the end of the other, such that a 1:1 covering of the data is achieved. For the analysis here, though, the jump size is 300, and smaller than the prediction window of 2100 time steps, such that a more crisp resolution for the measure evolution is reached. That being said, there are two different methods used to run the RC alongside with the combustion data. One method, called *fixed-RC*,

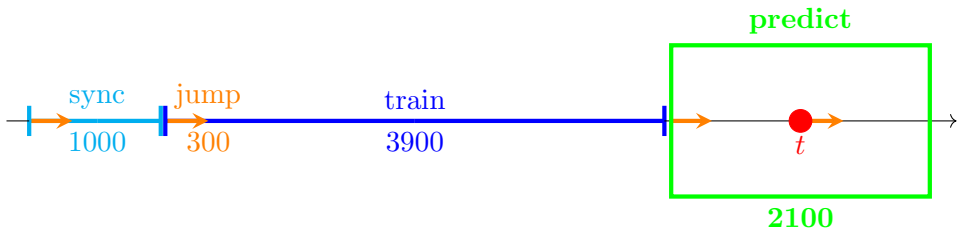


Figure 44: *The complete framework for the sliding window technique requires 7000 time steps. The first 1000 time steps are used to synchronize the reservoir, while the next 3900 time steps of the time series data are used to train the reservoir. The last 2100 time steps are the prediction parallel to the data. The finally calculated quantity on this green boxed region is assigned to the center time. This is important to know for the following evolution plots in this section. Each jump, or shift, of the framework takes 300 time steps. That means that there is a large overlap for each framework. The training phase does not apply to the fixed-RC method.*

is that the reservoir was trained once on a target state of the combustion dynamics and then applied over the complete time-series with fixed output-weights. Since the reservoir was trained with 5 several randomization seeds, its statistical blurriness with respect to the according quantity, which hints to the significance in the deviation to the same quantity of the data. The other method, called *flexible-RC*, the reservoir is trained on each step of the sliding window, and it is of interest how well the reservoir can adapt to the new situation, by keeping the same hyperparameter. The fixed-RC has some similarities with the RC at all. The RC as used, outputs at each time step the prediction Y^t, which afterwards serves as input X^{t+1} and is thus independent from the data for the number of prediction time steps. In contrast, the sliding window of the fixed-RC, synchronizes once in a while again with the original data. But as it turns out, the RC output does not vary more than its intrinsic standard deviation, despite the new synchronized dynamical states of the data. Short, the fixed-RC is totally independent from the data and will always stay in its trained dynamics. The flexible-RC, on the other hand, tries to learn each new dynamical state. This provides a natural threshold for the boundaries of a certain quantity, if the reservoir is train

several times with each time different random networks. It has the advantage, that it would also take into account for environmental changes of the physical system, and serves as a real digital twin, that evolves dynamically side by side. That is the reason why in this thesis the flexible type is chosen. However, it is also interesting, that had shown, that a once trained reservoir is almost unaffected by its dynamics, when it is synchronized on extraneous data.

One further important design choice is the type of hyperparameter, which is used all over the dynamical evolution. One can use a hyperparameter selection, that suits best for the target state, the instability or one, that is nowhere the optimum, but the best compromise over all types of dynamics. For here, the latter was chosen, with an regression parameter of 100 and a spectral radius of 3.0. A combination that showed decent results on all 3 regions of the previous sections. The usage for a combination that works best on the target state, at least for the frequency reproduction, a spectral radius of 1.5 and a regression parameter of 1, showed very similar results for the transient states. An adaptation for the hyperparameters during the slide is not trivial, since for the implementation without further information like in this thesis, one had to know the upcoming dynamical changes to accordingly adapt the hyperparameter, which is in contradiction to an early warning system.

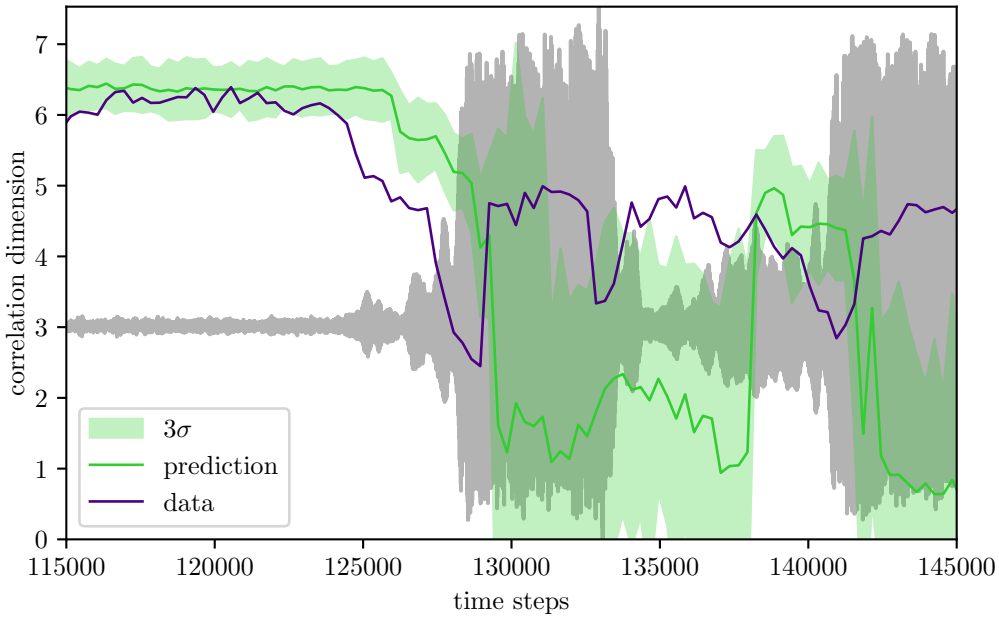


Figure 45: *In this evolution plot each 300 time steps the correlation dimension is calculated. One time for the original data, and for 30 realizations of the RC. For that 3 times the standard deviation is calculated and the mean. In the background one can see the current situation in the time series in gray.*

As one can see in the figures 45, 46 and 47, there is a delay between the green and the purple line. This means that the data properties are changing before the reservoir prediction experiences state changes in the dynamics. This delay takes about 2400 time steps, meaning 24 ms. Since the time distance between the middle of the training time and the middle of the prediction time window is 3000 time steps, the delay may result

from the difference between the dynamics climate of the training data set and the test data set, as marked in figure 44.

The basic idea is, for a warning of a state transition, that the uncertainty margin of the adapted reservoir gets crossed by the change of the actual system, and can thus raise a warning signal for a possible control intervention. The margin is set here to 3 times the standard deviation σ out of 30 realizations for the reservoir network at each jump.

From the 12 measures tested in the previous section, 4 measures are chosen as representatives. The correlation dimension, as measure of complexity, had shown a good distinction in figure 23b for the different dynamical states, and is thus an interesting candidate. It shows in figure 45 a smooth transition of the complexity, towards smaller values right at the transient situations. As well as it enters the thermoacoustic oscillation, as well as it leaves the state. The reservoir mimics that behavior, which is visible when considering the time offset of 2400 time steps, but in an but in an exaggerated magnitude. The correlation dimension might be cut off in the beginning of the plot, since it was only embedded in 7 dimensions for the calculation of the correlation dimension. This is done because of the limited number of data points for each calculation, as discussed in section 4.1.3.

Another good choice of monitoring measure is the RQA quantity LAM. It is extreme sensitivity to the dynamical states, and was the measure that had a really strong and early slope, for the transition from the stable state towards the instability. It has the strongest contrast of values between the stable states and the instabilities, as it is noticeable in figure 46. Moreover, it is, like for all RQA measures, with the used algorithms significantly faster to compute than the correlation dimension. Also unlike the correlation dimension, the differences between the dynamical states are more faint for the prediction, as for the true value. But the prediction values are more stable all over the time series, where the correlation dimension gots very vague for certain regions, like for time step 130,000. The laminarity LAM had for each dynamic state a clear 2σ difference, which is an advantage in terms of information content and usability. The third used measure is the recurrence rate RR. The strength of this measure is that it has a very narrow corridor for the values at the target state, and the system values are almost all within the margins of the predicted state. Also the edge of the signal is very sharp, and hence good for a warning trigger. The rise of recurrence towards the instabilities can be explained with the theory of [15], that towards the instability, the acoustic oscillations become periodic. That also fits the findings of the prediction length evolution along the time series in figure 48. At the beginnings of the instabilities, the prediction length rises from near zero up to a few hundred time steps, which is also discussed in section 4.2. This high predictability might also be explained by the shift from aperiodicity towards periodicity [15] by occurring excited modes in thermoacoustic oscillation. That also explains the observation in section 4.2, that for region 2 the strong linear predictions could stay in phase. All in all, every quantity, which has an local stable and global significant reacting behavior, with respect to the system changes, shows at around the same moment a change, and there isn't known yet a true precursor for a few hundred milliseconds, by the results of this work. But, the selection of measures shows a significant change of dynamics roundabout 5000 time steps, or 50 ms, before the high amplitude oscillation starts.

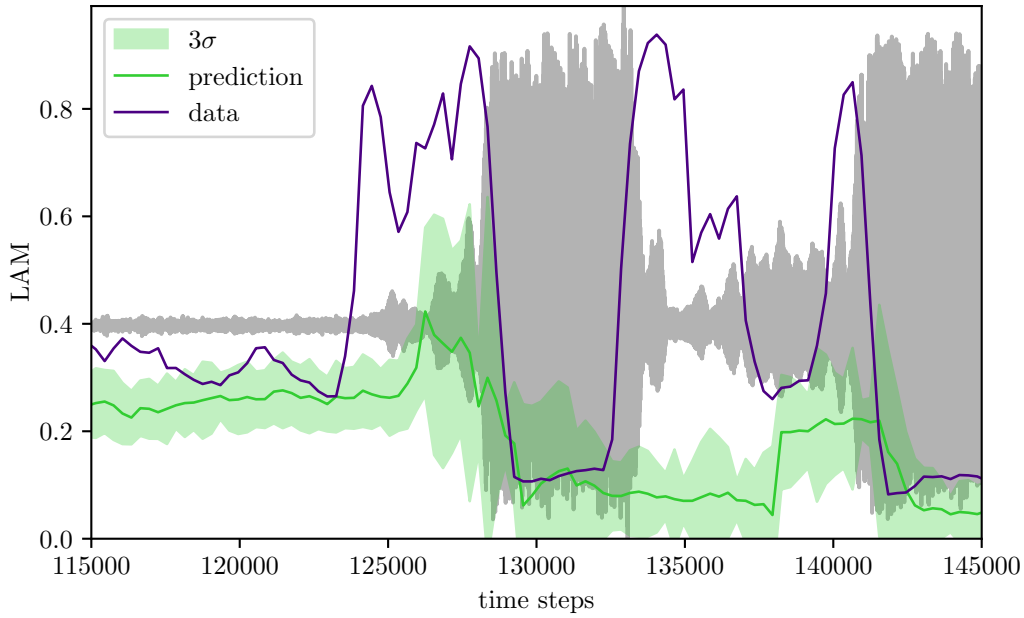


Figure 46: *In this evolution plot each 300 time steps the laminarity (LAM) is calculated. One time for the original data, and for 30 realizations of the RC. For that 3 times the standard deviation is calculated and the mean. In the background one can see the current situation in the time series in gray.*

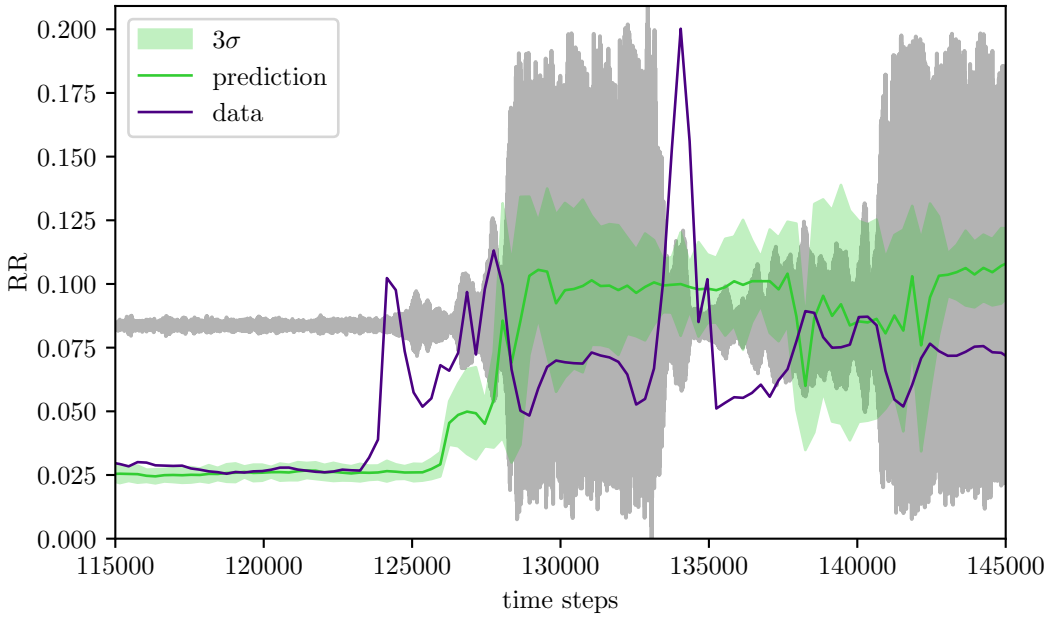


Figure 47: *In this evolution plot each 300 time steps the recurrence rate (RR) is calculated. One time for the original data, and for 30 realizations of the RC. For that 3 times the standard deviation is calculated and the mean. In the background one can see the current situation in the time series in gray.*

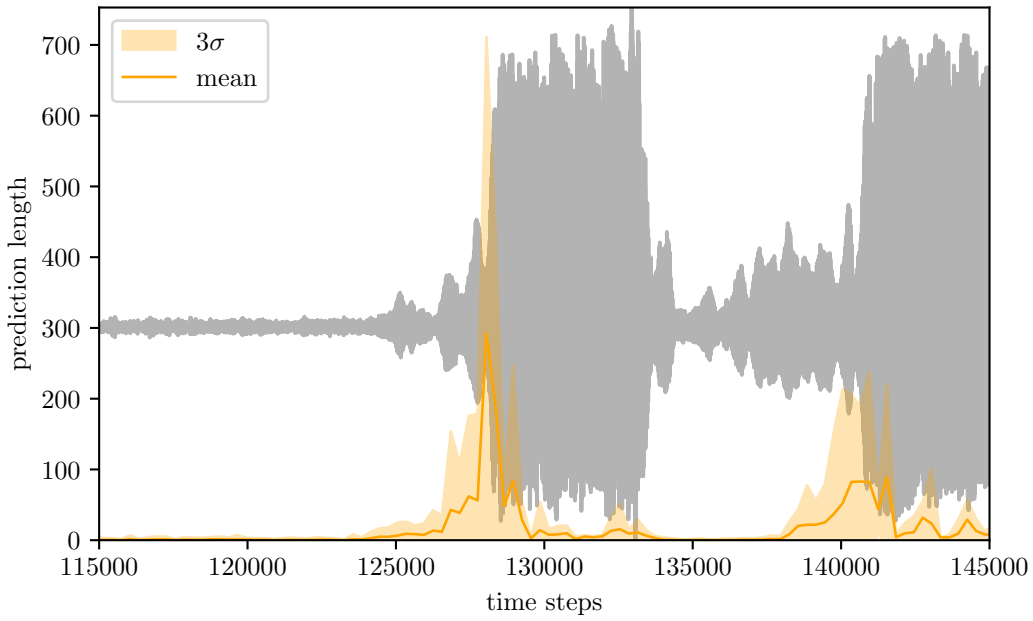


Figure 48: *In this evolution plot each 300 time steps the prediction length is shown. Each time 30 different reservoirs were trained. For that 3 times the standard deviation is calculated and the mean. In the background one can see the current situation in the time series in gray.*

Conclusion and Outlook

Over all, it is shown that the reservoir could reproduce the typical long term characteristics of different dynamical states in the thermoacoustic pressure data. In the following, we will summarize for the various aspects of this work the most important findings and make suggestions on how they can be further developed.

Data Our work on the relative pressure oscillation signal could confirm the statement "The pressure signals in the unstable (limit-cycle) regime possess a deterministic periodic nature, while the stable regime is distinguished by a noisy or chaotic nature" in [12]. The target state shows indeed a more complex structure than the instabilities, which are dominated by a sharper frequency spectrum and lower correlation dimension. This applies in particular to the transition region, called region 2 in this thesis. But it also turned out that the different events of instability type 2 within the data set differ further in their properties. This shows section 4.1.4, where some instabilities have a significant shift in the dominant frequency, others do not. And also the amplitude modulation has a different characteristic. This could be evidence that the acoustic energy is stored in different modes. For the time-delay phase space reconstruction, an optimal delay of 2 time steps and an embedding dimension of 3 was determined.

Reservoir Computing With the RC it is actually possible to reproduce both the linear frequency spectra and the phase space complexity for different dynamical states. But furthermore, it is made for the first time a classification for numerically artifacts outside the optimal region for the learning task, to provide an approach for a deeper understanding of RC. It is discussed which types of these failures lead to "false friends" during the hyperparameter optimization. A very promising result is that it is possible for RC to learn dynamical patterns on different time scales, as shown in 34a, where a pattern is visible with a time scale of about 1000 time steps alongside the original 10 time steps periodicity of the underlying dynamics. Even if it still needs to be improved. This improvement might be achieved either by a larger training interval or by using a higher sampling rate during the measurement and thus more information about each signal period, with the former being preferable. For the tested hyperparameters, spectral radius and the regression parameter, it turned out that unusually high values give good results, in contrast to values that are often taken into account. This might be the case, because it is used a large and sparse ESN with 3000 nodes, and thus a large average network path length. In addition, the results indicate that a higher spectral radius is appropriate for data with a strong linear character than for data with a more complex structure, such as the target state of the combustor data. Furthermore, all three regions tested could be learned from the reservoir without further input normalization, despite large differences in amplitude values and even a non-zero mean.

Measures As already mentioned, nearly all measures tend to have "false friends", which are hyperparameter regions of small error, but bad predictions. This often appears to be the case for too small spectral radii or too large regression parameters. A

weighted summation of complementary measures could lead to the exclusion of false friends, and hence enable a powerful way for optimization without human supervision. The combination of the NRMSE on the amplitude distribution and on the fourier spectrum moments, could be a good and computational inexpensive candidate for this task. In general, further fine tuning, like the number of used moments or the bin size for distributions, can yield better results, since all measures except the fourier spectrum NRMSE do depend on freely adjustable parameters.

Sliding Window Evolution With the used sliding window setup, the first major changes to the system properties appeared at the earliest 50 ms before the maximum thermoacoustic oscillations were reached. This is a first guideline for how far in advance one can perceive an instability without any information other than the pressure signal. Furthermore, it turned out that a once-trained reservoir retains its trained dynamics, regardless of the synchronization with changing system properties. Therefore, a co-developing reservoir is used, which provides a warning of drastically changing quantities. It turned out that the RQA based laminarity is a good metric to monitor the system status. It showed the most significant changes in each dynamic state. For further optimization, one should optimize the reservoir with respect to the fastest measure, e.g. the laminarity, to get higher sensitivity and stricter thresholds. However, the computing cost of RC and the calculation of the measure must be taken into account for fast time series such as thermoacoustic oscillations. Nevertheless, it is an exciting new method as an adaptive monitoring tool for future control methods.

References

[1] P. Kasthuri, I. Pavithran, S. A. Pawar, R. Sujith, R. Gejji, and W. Anderson, "Dynamical systems approach to study thermoacoustic transitions in a liquid rocket combustor," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 29, no. 10, 2019.

[2] M. Grieves, "Digital twin: manufacturing excellence through virtual factory replication," *White paper*, vol. 1, no. 2014, pp. 1–7, 2014.

[3] A. Haluszczynski and C. Räth, "Controlling nonlinear dynamical systems into arbitrary states using machine learning," *Scientific reports*, vol. 11, no. 1, p. 12991, 2021.

[4] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note," *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, vol. 148, no. 34, p. 13, 2001.

[5] W. Maass and H. Markram, "On the computational power of circuits of spiking neurons," *Journal of computer and system sciences*, vol. 69, no. 4, pp. 593–616, 2004.

[6] M. D. Skowronski and J. G. Harris, "Automatic speech recognition using a predictive echo state network classifier," *Neural networks*, vol. 20, no. 3, pp. 414–423, 2007.

[7] A. Jalalvand, G. Van Wallendael, and R. Van de Walle, "Real-time reservoir computing network-based systems for detection tasks on visual contents," in *2015 7th International Conference on Computational Intelligence, Communication Systems and Networks*, pp. 146–151, IEEE, 2015.

[8] B. T. Nadiga, "Reservoir computing as a tool for climate predictability studies," *Journal of Advances in Modeling Earth Systems*, vol. 13, no. 4, p. e2020MS002290, 2021.

[9] F. X. Moreira Huhn, *Optimisation of chaotic thermoacoustics*. PhD thesis, University of Cambridge, 2021.

[10] F. Huhn and L. Magri, "Gradient-free optimization of chaotic acoustics with reservoir computing," *Physical Review Fluids*, vol. 7, no. 1, p. 014402, 2022.

[11] L.-W. Kong, Y. Weng, B. Glaz, M. Haile, and Y.-C. Lai, "Reservoir computing as digital twins for nonlinear dynamical systems," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 33, no. 3, 2023.

[12] G. Waxenegger-Wilfing, U. Sengupta, J. Martin, W. Armbruster, J. Hardi, M. Juniper, and M. Oschwald, "Early detection of thermoacoustic instabilities in a cryogenic rocket thrust chamber using combustion noise features and machine learning," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 31, no. 6, 2021.

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

[14] L. Rayleigh, "The explanation of certain acoustical phenomena," *Roy. Inst. Proc.*, vol. 8, pp. 536–542, 1878.

[15] M. P. Juniper and R. I. Sujith, "Sensitivity and nonlinearity of thermoacoustic oscillations," *Annual Review of Fluid Mechanics*, vol. 50, pp. 661–689, 2018.

[16] S. Gröning, *Untersuchung selbsterregter Verbrennungsinstabilitäten in einer Raketenbrennkammer*. PhD thesis, RWTH Aachen, 2017.

[17] T. Fiala, *Radiation from high pressure hydrogen-oxygen flames and its use in assessing rocket combustion instability*. PhD thesis, Technische Universität München, 2015.

[18] S. Gröning, J. S. Hardi, D. Suslov, and M. Oschwald, "Injector-driven combustion instabilities in a hydrogen/oxygen rocket combustor," *Journal of Propulsion and Power*, vol. 32, no. 3, pp. 560–573, 2016.

[19] J. S. Hardi, T. Traudt, C. Bombardieri, M. Börner, S. K. Beinke, W. Armbruster, P. N. Blanco, F. Tonti, D. Suslov, B. Dally, *et al.*, "Combustion dynamics in cryogenic rocket engines: Research programme at dlr lampoldshausen," *Acta Astronautica*, vol. 147, pp. 251–258, 2018.

[20] M. Nakahara, *Geometry, topology and physics*. CRC press, 2018.

[21] H. Whitney, "Differentiable manifolds," *Annals of Mathematics*, vol. 37, no. 3, pp. 645–680, 1936.

[22] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980: proceedings of a symposium held at the University of Warwick 1979/80*, pp. 366–381, Springer, 2006.

[23] T. Sauer, J. A. Yorke, and M. Casdagli, "Embedology," *Journal of statistical Physics*, vol. 65, pp. 579–616, 1991.

[24] J. C. Sprott and J. C. Sprott, *Chaos and time-series analysis*, vol. 69. Oxford university press Oxford, 2003.

[25] S. H. Strogatz, "Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering," 1994.

[26] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[27] N. Mars and G. Van Arragon, "Time delay estimation in non-linear systems using average amount of mutual information analysis," *Signal processing*, vol. 4, no. 2-3, pp. 139–153, 1982.

[28] M. B. Kennel, R. Brown, and H. Abarbanel, "Determining embedding dimension for phase space reconstruction using the method of false nearest neighbors," *Phys. Rev. A*, vol. 45, no. 6, pp. 3403–3411, 1992.

[29] C. Rhodes and M. Morari, "The false nearest neighbors algorithm: An overview," *Computers & Chemical Engineering*, vol. 21, pp. S1149–S1154, 1997.

[30] L. Cao, "Practical method for determining the minimum embedding dimension of a scalar time series," *Physica D: Nonlinear Phenomena*, vol. 110, no. 1-2, pp. 43–50, 1997.

[31] R. Hegger and H. Kantz, "Improved false nearest neighbor method to detect determinism in time series data," *Physical Review E*, vol. 60, no. 4, p. 4970, 1999.

[32] S.-I. Amari, "Learning patterns and pattern sequences by self-organizing nets of threshold elements," *IEEE Transactions on computers*, vol. 100, no. 11, pp. 1197–1206, 1972.

[33] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, "Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach," *Physical review letters*, vol. 120, no. 2, p. 024102, 2018.

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

[35] H. Jaeger, "Short term memory in echo state networks," 2001.

[36] A. Amidi and S. Amidi, "Stanford University lecture CS 230 - Deep Learning by Andrew Ng." `https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks`. Accessed: 2023-08-23.

[37] C. C. Aggarwal, *Optimization in Computational Graphs*, pp. 447–482. Cham: Springer International Publishing, 2020.

[38] H. Ma, D. Prosperino, and C. Räth, "A novel approach to minimal reservoir computing," *Scientific Reports*, vol. 13, no. 1, p. 12970, 2023.

[39] P. Erdős, A. Rényi, *et al.*, "On the evolution of random graphs," *Publ. math. inst. hung. acad. sci*, vol. 5, no. 1, pp. 17–60, 1960.

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

[41] S. Basterrech and G. Rubino, "Evolutionary echo state network: evolving reservoirs in the fourier space," *arXiv preprint arXiv:2206.04951*, 2022.

[42] T. L. Carroll and L. M. Pecora, "Network structure effects in reservoir computers," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 29, no. 8, 2019.

[43] A. Haluszczynski and C. Räth, "Good and bad predictions: Assessing and improving the replication of chaotic attractors by means of reservoir computing," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 29, no. 10, 2019.

[44] Y. Kawai, J. Park, and M. Asada, "A small-world topology enhances the echo state property and signal propagation in reservoir computing," *Neural Networks*, vol. 112, pp. 15–23, 2019.

[45] H. Jaeger, M. Lukoševičius, D. Popovici, and U. Siewert, "Optimization and applications of echo state networks with leaky-integrator neurons," *Neural networks*, vol. 20, no. 3, pp. 335–352, 2007.

[46] F. Wyffels, B. Schrauwen, and D. Stroobandt, "Stable output feedback in reservoir computing using ridge regression," in *International conference on artificial neural networks*, pp. 808–817, Springer, 2008.

[47] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[48] J. Herteux and C. Räth, "Breaking symmetries of the reservoir equations in echo state networks," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 30, no. 12, 2020.

[49] Z. Lu, J. Pathak, B. Hunt, M. Girvan, R. Brockett, and E. Ott, "Reservoir observers: Model-free inference of unmeasured variables in chaotic systems," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 27, no. 4, 2017.

[50] C. Gallicchio, "Chasing the echo state property," *arXiv preprint arXiv:1811.10892*, 2018.

[51] A. Hart, J. Hook, and J. Dawes, "Embedding and approximation theorems for echo state networks," *Neural Networks*, vol. 128, pp. 234–247, 2020.

[52] P. Grassberger and I. Procaccia, "Measuring the strangeness of strange attractors," *Physica D: nonlinear phenomena*, vol. 9, no. 1-2, pp. 189–208, 1983.

[53] C. L. Webber and N. Marwan, "Recurrence quantification analysis," *Theory and Best Practices*, p. 426, 2015.

[54] N. Marwan, J. Kurths, and P. Saparin, "Generalised recurrence plot analysis for spatial data," *Physics Letters A*, vol. 360, no. 4-5, pp. 545–551, 2007.

[55] J. F. Donges, J. Heitzig, B. Beronov, M. Wiedermann, J. Runge, Q. Y. Feng, L. Tupikina, V. Stolbova, R. V. Donner, N. Marwan, *et al.*, "Unified functional network and nonlinear time series analysis for complex systems science: The pyunicorn package," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 25, no. 11, 2015.

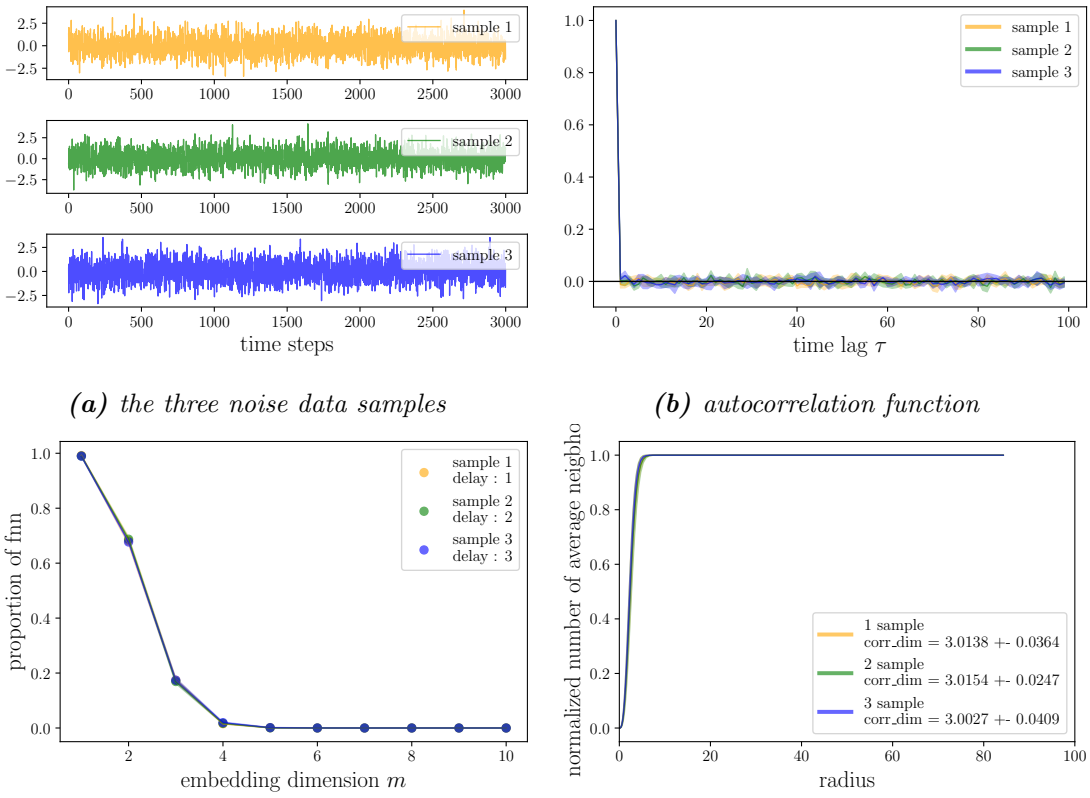
[56] S. Schinkel, O. Dimigen, and N. Marwan, "Selection of recurrence threshold for signal detection," *The european physical journal special topics*, vol. 164, no. 1, pp. 45–53, 2008.

[57] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.

[58] "NumPy documentation np.fft.fft." `https://numpy.org/doc/stable/reference/generated/numpy.fft.fft.html`. Accessed: 2023-07-16.

Appendix

Methods Test on Noise



(a) *the three noise data samples*

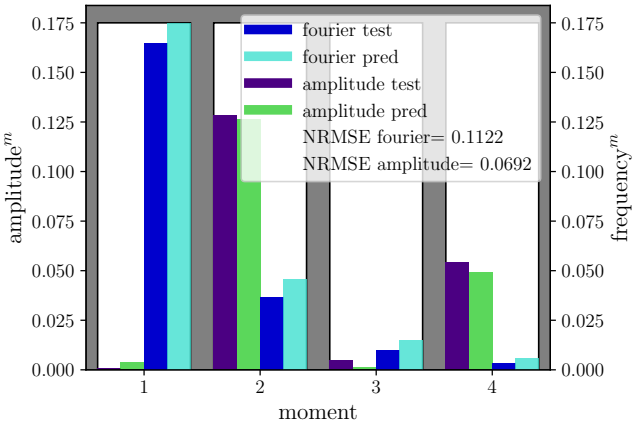
(b) *autocorrelation function*

(c) *FNN*

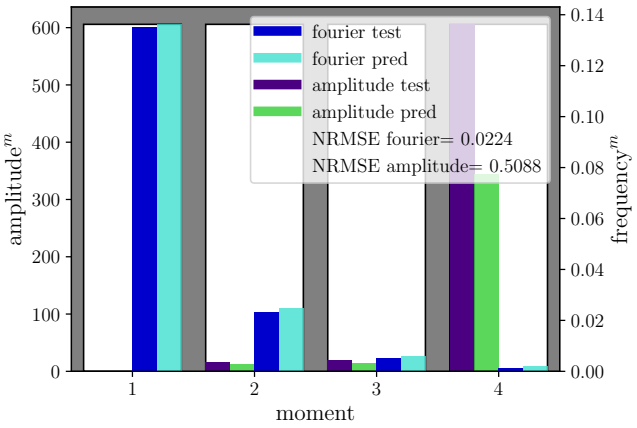
(d) *correlation sum and dimension*

Figure 49: *As comparison, five times 3000 data points of a noise time series, sampled from a gaussian normal distribution. There are three completely equally sampled "coordinates" of the time serie, embedded with a delay of 1, 2 and 3 such that it is also shown that there is no effect and hence no proper embedding for a completely stochastic time series. The plot includes both the mean and the standard deviation.*

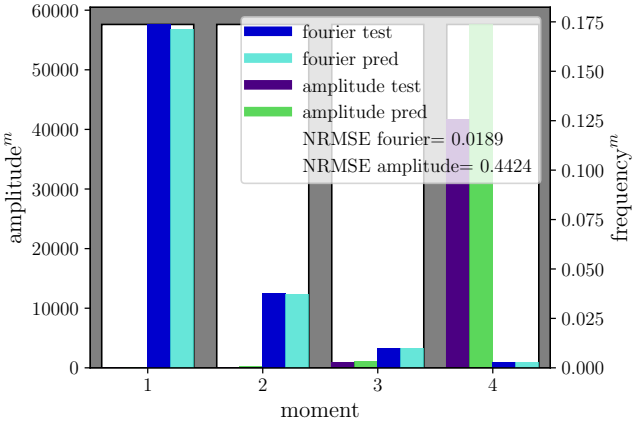
Appendix to section 4.2 and 4.3



(a) moments region 1

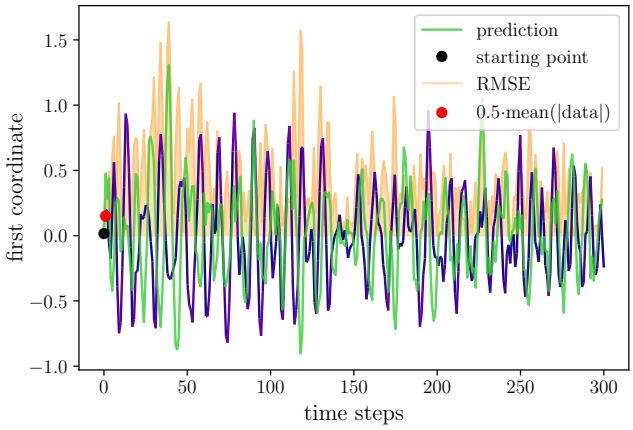


(b) moments region 2

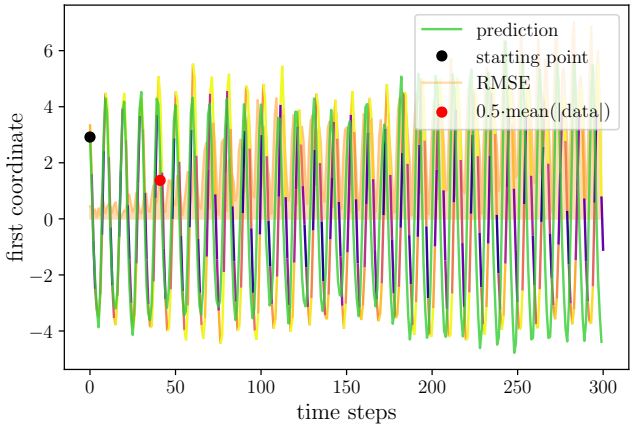


(c) moments region 3

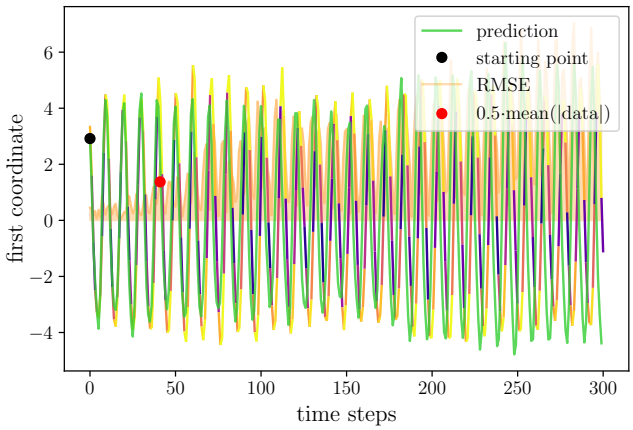
Figure 50: *This are the used moments of the power spectral density and the amplitude distribution for the two moments measures. Top to down, the moment distributions for region 1 to 3 with the comparison to the respective prediction values for the results of the figures 33, 34 and 35.*



(a) region 1

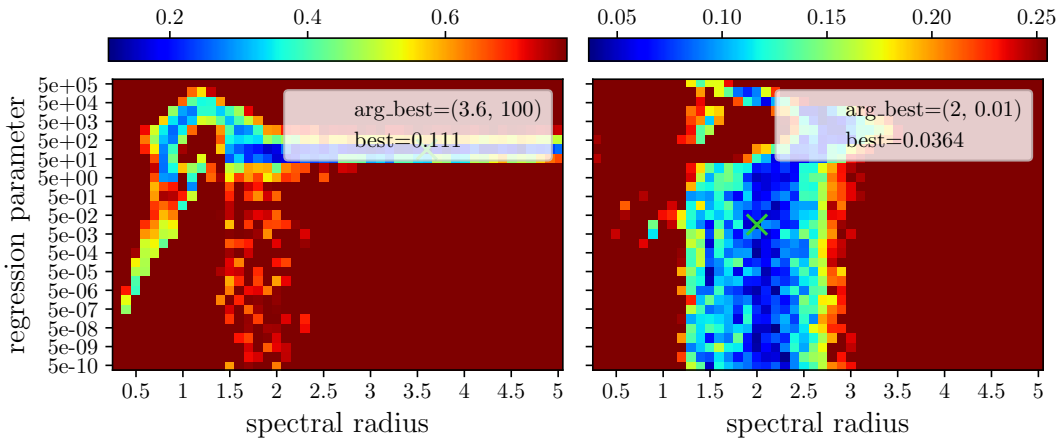


(b) region 2

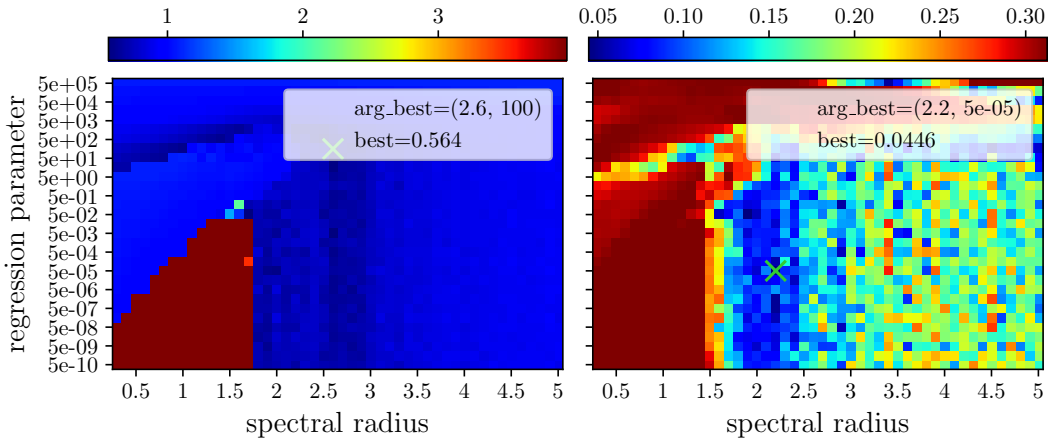


(c) region 3

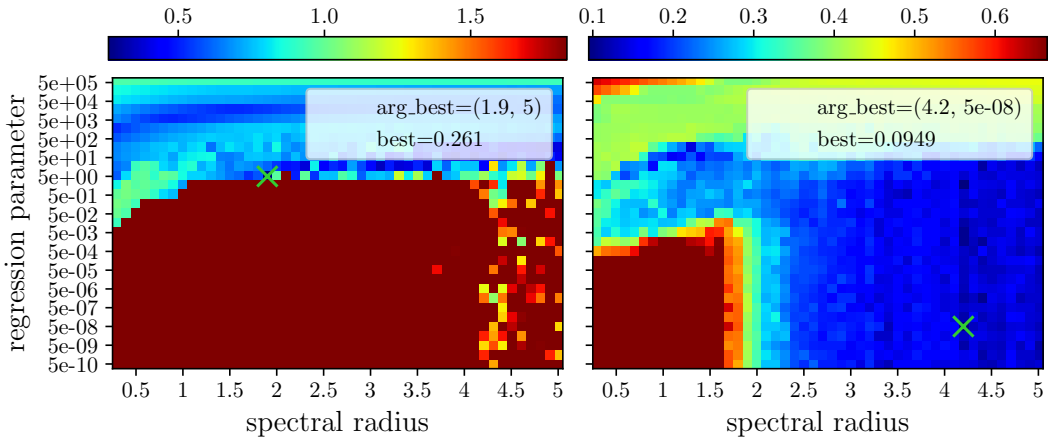
Figure 51: *One can see the first 300 hundred time steps of the predictions for each of the results of the figures 33, 34 and 35. So the individual periods of the oszillation are good visible, such that it is obvious that an exact short term prediction isn't possible or reasonable for the actual setup.*



(a) *NRMSE amplitude moments R1* ***(b)*** *NRMSE RATIO R1*

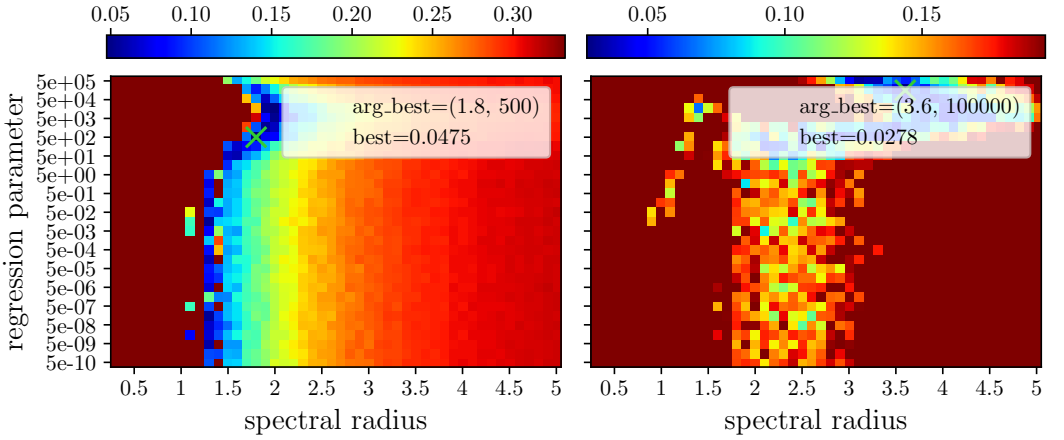


(c) *NRMSE amplitude moments R2* ***(d)*** *NRMSE RATIO R2*

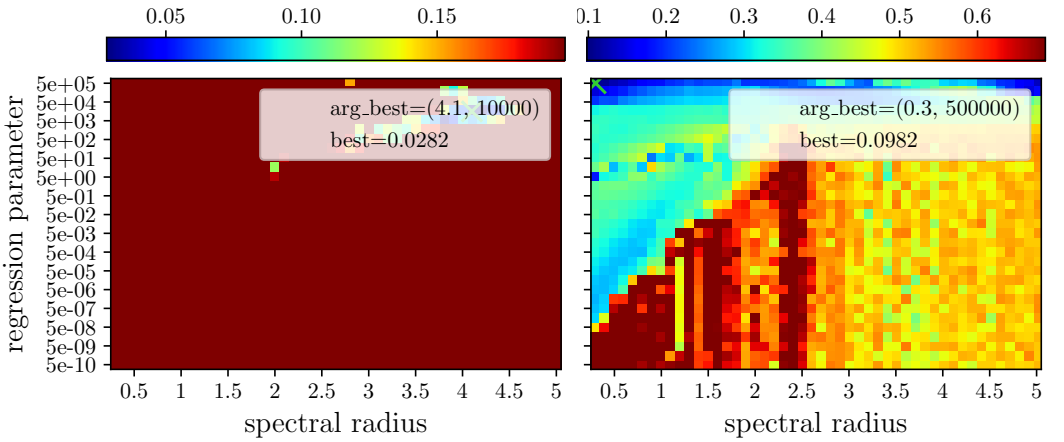


(e) *NRMSE amplitude moments R3* ***(f)*** *NRMSE RATIO R3*

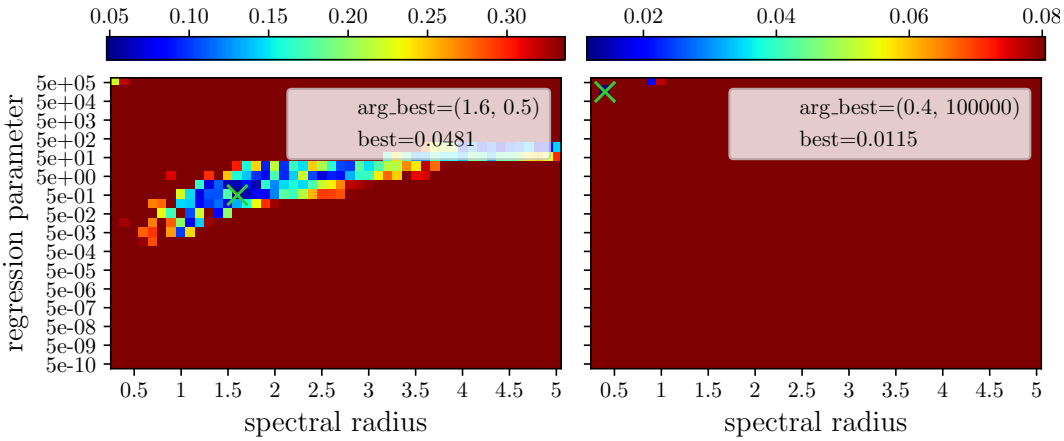
Figure 52: *Comparison of the hyperparameter-dependencies for phase space occupation and determinism via the amplitude distribution moments and the RQA measure RATIO. Each pixel is the mean out of 7 runs.*



(a) NRMSE L R1 *(b) NRMSE LAM R1*

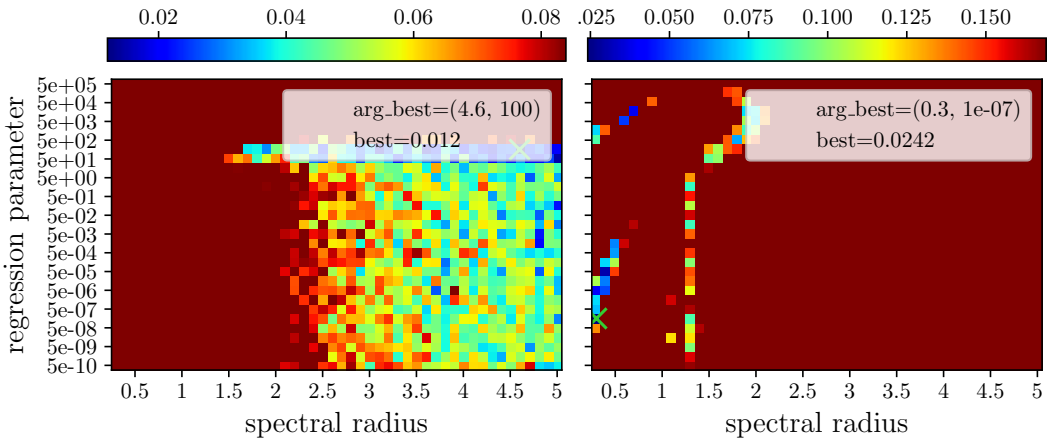


(c) NRMSE L R2 *(d) NRMSE LAM R2*



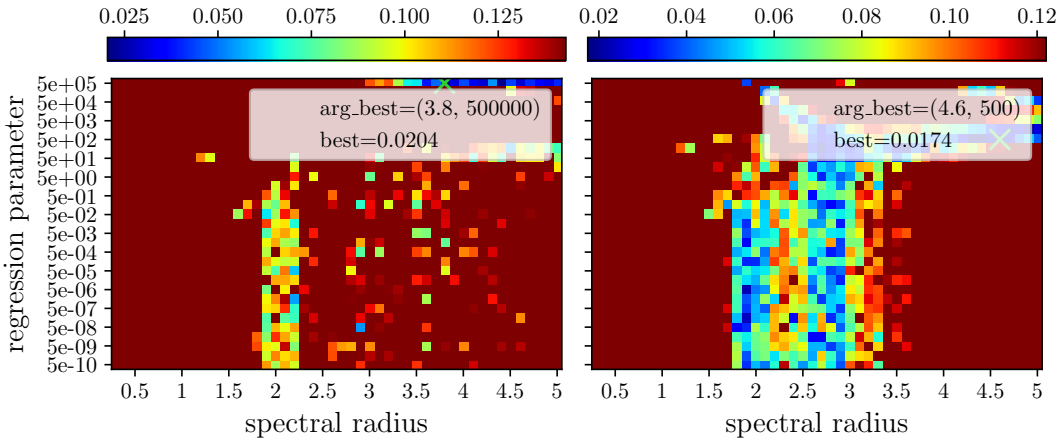
(e) NRMSE L R3 *(f) NRMSE LAM R3*

Figure 53: *Comparison of the hyperparameter-dependencies for the diagonal line structure and the vertical line structure in the recurrence plot, due to L and LAM. Each pixel is the mean out of 7 runs.*



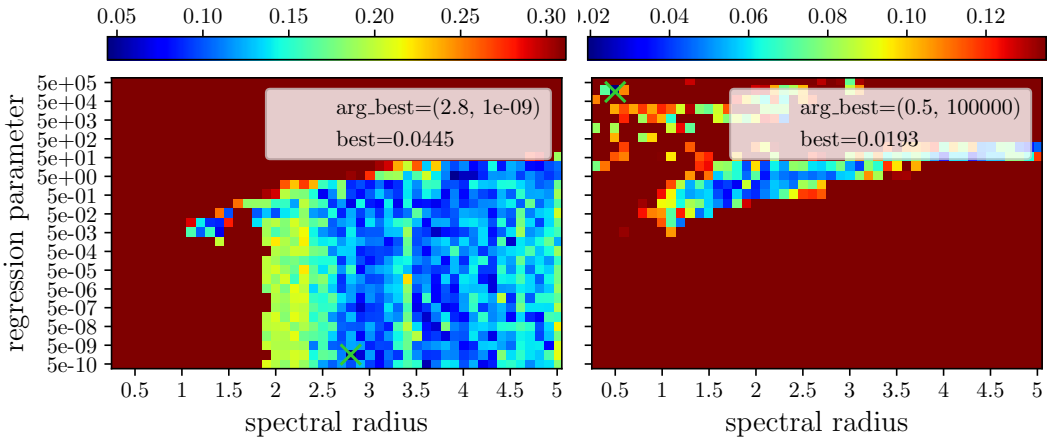
(a) NRMSE RR R1

(b) NRMSE ENTR R1



(c) NRMSE RR R2

(d) NRMSE ENTR R2



(e) NRMSE RR R3

(f) NRMSE ENTR R3

Figure 54: *Comparison of the hyperparameter-dependencies for amount of events in the recurrence plot and the shannon entropy of the diagonal lines due to the recurrence measure RR and the entropy measure ENTR. Each pixel is the mean out of 7 runs.*

List of Parameters

ESN	
number of nodes	3,000
spectral radius	0.3, ..., 5.0
average degree	280
regression parameter	5×10^{-10}, ..., $5\times10^{-+5}$
training steps	3900
synchronization steps	1000
evaluation steps	2100
W_{in} scale	0.8
Measures	
RQA threshold	0.5σ
used moments	1,2,3,4
embedding for correlation dimension	7
number bins for amplitude distribution	128
forecast threshold	half the mean absolute value

Time-Delay Embedding vs. Multiple Pressure Sensors

To determine whether there is more information in the other pressure sensors, the same setup was done as in section 4.3, and then the optima were compared to see whether 3 time-delay coordinates or 3 pressure sensors are more effective.

measure	region 1	region 2	region 3
correlation dimension	D[13%]	S[82%]	D[17%]
fourier	S[1%]	D[10%]	D[7%]
prediction length	S[39%]	D[6%]	D[16%]
fourier moments	S[15%]	D[16%]	D[38%]
amp. distrib. moments	S[3%]	S[76%]	S[87%]
RR	D[63%]	D[39%]	D[21%]
DET	D[2%]	D[68%]	D[54%]
L	S[73%]	D[26%]	S[23%]
LAM	S[21%]	S[88%]	D[70%]
ENTR	D[57%]	S[71%]	D[66%]
RATIO	S[46%]	S[2%]	S[94%]

Table 1: *For each measure the optimal value for regression parameter and spectral radius was compared. The 3-dimensional delay embedding method "D" was better in 18 cases than the usage of three sensors S, which was better in 15 cases. Each optimum is the mean out of 7 reservoir initializations. In [·] the error reduction in % by choosing the better method. This means values from 0% to 100% are possible and a higher number equals more performance gain.*

Acknowledgement

I would like to express my sincere gratitude to Christoph for the great entry into the field of machine learning and AI, as well as for the advice, discussion, and guidance throughout my path into computational based research. I would also like to thank Günther Waxenegger-Wilfing for the kind professional support and the German Aerospace Center DLR for the opportunity to work on this topic. Furthermore, I would like to thank Sebastian for his patience and nice way of helping me with organizational hurdles and for always being a reliable contact person for IT and research topics.

Selbständigkeitserklärung

Hiermit erkläre ich, die vorliegende Arbeit selbständig verfasst zu haben und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel benutzt zu haben.

München, 01. September 2023