ABSTRACT

Title of dissertation: DATA-DRIVEN STUDIES OF TRANSIENT EVENTS AND APERIODIC MOTIONS

Rui Wang
Doctor of Philosophy, 2019

Dissertation directed by: Professor Balakumar Balachandran
Department of Mechanical Engineering

The era of big data, high-performance computing, and machine learning has witnessed a paradigm shift from physics-based modeling to data-driven modeling across many scientific fields. In this dissertation work, transient events and aperiodic motions of complex nonlinear dynamical system are studied with the aid of a data-driven modeling approach. The goal of the work has been to further the ability for future behavior prediction, state estimation, and control of related behaviors.

It is shown that data on extreme waves can be used to carry out stability analysis and ascertain the nature of the transient phenomenon. In addition, it is demonstrated that a low number of soliton elements can be used to realize a rogue wave on the basis of nonlinear interactions amongst the basic elements. The proposed nonlinear phase interference model provides an appealing explanation for the formation of ocean extreme wave and related statistics, and a superior reconstruction of the Draupner wave event than that obtained on the basis of linear superposition.

Chaotic data, another manifestation of aperiodic motions, which are obtained

from prototypical ordinary differential and partial differential systems are considered and a neural machine is realized to predict the corresponding responses based on a limited training set as well to forecast the system behavior. A specific neural architecture, called the inhibitor mechanism, has been designed to enable chaotic time series forecasting. Without this mechanism, even the short-term predictions would be intractable. Both autonomous and non-autonomous dynamical systems have been studied to demonstrate the long-term forecasting possibilities with the developed neural machine. For each dynamical system considered in this dissertation, a long forecasting horizon is achieved with a short historical data set. Furthermore, with the developed neural machine, one can relax the requirement of continuous historical data measurements, thus, providing for a more pragmatic approach than the previous approaches available in the literature.

It is expected that the efforts of this dissertation work will lead to a better understanding of the underlying mechanism of transient and aperiodic events in complex systems and useful techniques for forecasting their future occurrences.

DATA-DRIVEN STUDIES OF TRANSIENT EVENTS AND APERIODIC MOTIONS

by

Rui Wang

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2019

Advisory Committee:
Professor Balakumar Balachandran, Chair & Advisor, Mechanical Engineering
Associate Professor Peter Chung, Mechanical Engineering
Professor James Duncan, Mechanical Engineering
Professor Eugenia Kalnay, Dean's Representative, Atmospheric and Oceanic Science
Associate Professor N. Johan Larsson, Mechanical Engineering

I dedicate this thesis to

my family, my friends, and my

colleagues for their persistent support

and unconditional love.

I love you all dearly.

Acknowledgements

I would like to thank my advisor, Professor Balakumar Balachandran, for his invaluable support and guidance, be it in the academic research or doctoral life. He is a truly remarkable character with talent and patience. I learned from him that integrity and ingenuity are two cornerstones for any kind of enterprise. He is enthusiastic about the research and full of insightful advice. It was unquestionably enjoyable and pleasant to work with him.

There is a saying which goes "you need a support ecosystem in order to survive and stay sane in the grad school". I am lucky to have a group of friends with excellent minds and outstanding characters. They are indispensable assets to me. We are like a "buddy" system where one can undoubtedly rely on each other. We had routine workout schedule in the gym. We learned Japanese Kendo every Sunday. We held interdisciplinary PhD seminars every Friday afternoon to discuss interesting research topics. We went to beaches during summer and discussed research methodologies in an ocean-front cabin. We played table board and poker games. We spent time cooking meals and tasting wines. Their encouragement and advice have carried me over various stages in my doctoral life.

I thank the continuous support from my colleagues in the Dynamics and Vibrations Group. The diverse backgrounds in my group really broaden the horizon and make me realize how small and delicate my research is. It makes me humble. Besides, they have provided intellectual support for some of the problems I faced during the PhD training period. I also collaborated with some of them to investigate

the interdisciplinary research topics. It was a interesting and rewarding experience.

I also thank the people who made me realize that life is full of challenges and surprises. It made me appreciate all the beauty and love which I have in life. It helped me to build resilience throughout the hardships. No pain, no gain.

I will forever be indebted to my parents, Fengying Hao and Baoguo Wang, for their personal sacrifice and unconditional love, without which none of these work would have been possible.

Lastly, I am beholden to all the committee members for their valuable feedbacks and comments on my dissertation work. Especially, I am extremely grateful to Professor Eugenia Kalnay for her sincere encouragement and insightful comments on my research work. Support received for this research through NSF Grant No. CMMI-1125285 and Maryland Advanced Research Computing Center (MARCC) is gratefully acknowledged.

Table of Contents

List of Figures

List of Abbreviations

AI	Artificial Intelligence
BD	Big Data
BPTT	Back-propagation through Time
DS	Dynamical System
ETDRK4	Exponential Time Differencing Runge-Kutta 4th method
FCM	Fourier Collocation Method
FD	Finite Difference
FP	Fixed Point
HPC	High Performance Computing
IST	Inverse Scattering Transform
JONSWAP	Joint North Sea Wave Project
KdV	Korteweg-de Vries
KL	Kullback-Leibler
KS	Kuramoto-Sivashinsky
l.h.s	left-hand side
LSTM	Long-Short Term Memory
MC	Monte Carlo
MI	Modulational Instability
NLSE	Nonlinear Schrödinger Equation
NMF	Neural Machine Forecasting
NWF	Numerical Weather Forecasting
ODE	Ordinary Differential Equation
r.h.s	right-hand side
RL	Reinforcement Learning
RNN	Recurrent Neural Network

PDE	Partial Differential Equation
SGD	Stochastic Gradient Descent
SL	Supervised Learning
SSL	Semi-supervised Learning
SVD	Singular Value Decomposition
SVM	Support Vector Machine
w.r.t	with respect to
UL	Unsupervised Learning
UTC	Universal Time Coordinated
ZS	Zakharov-Shabat

Chapter 1: Introduction

Science pursuits have been gradually moving from hypothesis-based exploration to data-driven discoveries. In contrast to the *reductionism* approach, wherein a scientist builds up knowledge from *first principles*, more and more knowledge is revealed through the inverse approach. With a data-driven method, scientists and engineers are able to extract useful information from massive amount of data. The main driving force during this transition is the advent of big data era. High-fidelity sensors, large-scale numerical simulations, and globally deployed instruments have enabled the generation of tremendous amount of data-streams at this moment. They provide unparalleled opportunities for researchers to analyze and discover new phenomena, in a manner which has not been done before. However, another aspect should also be mentioned, which is the awareness of the abundance of complex systems around us. The complex interactions amongst agents within such systems prohibit the traditional research methods based on first principles. A closer look obtained by breaking a large system into small components is not helping simplify the problem further, for example, fractals [4]. Therefore, the mathematical tools and analysis alone are always sufficient to reach conclusions. On the other hand, with a data-driven approach, one can study the complex problem from a reverse way

where the behaviors are analyzed in an integrated manner. The whole properties of the considered system will be reflected in the system's data representation, given sufficient data collection in time and assuming a relatively low noise environment. From this perspective, a large data set is expected to expedite the discovery process of physics, enabling new paradigms in science development. Moreover, the discovery will not be limited to the pool of hypothesis established by the scientists in advance.

Data-driven studies are a convergence of computer science, statistics, and physical and life sciences. Extensive data sets are produced by dynamical systems across disciplines in science and engineering, such as fluid dynamics [5], material science [6], molecular dynamics [7], and geophysics [8]. A great challenge is how to exploit the large amount of data, which is being gathered by measurements from sensors in experiments and outputs from numerical models to advance the current understanding of physics and reveal the predictability of behavior. Many of the complex systems, although appearing as high-dimensional and exhibiting rich multi-scale phenomena, often evolve to a low-dimensional subspace that can be characterized as spatiotemporal coherent structures. Therefore, successful extraction of these coherent structures is crucial to system identification and scientific discoveries. This process requires good learning algorithms that enable one to translate the superficially convoluted data set into meaningful perceptions.

Transient events can be momentary bursts of energy in a system caused by either an internal state change or an external driving force. The time scales of different transient events range from nanoseconds to years. The growth of transient events is the result of asymmetrical interactions amongst components. Asymme-

try is imperative in the sense that energy can be interchanged amongst different eigenmodes. If the energy is concentrated in a certain mode, the system displays a short duration of "extreme" behavior, or transient event. The energy cannot be increased ad infinitum. After a certain stage, the dissipation mechanism prevails and the system settles down to a steady state, before the next cycle of bursts occurs. Understanding the bursts of energy and building a precursor for its occurrence can potentially minimize the harmful consequences.

Aperiodic dynamics is ubiquitous in nature and human society. Traditionally, a good model representation of a certain system can help in predicting this system's future behavior. However, for a complex system, a physics-based model may not be easy to construct given the complexity of a system, in particular, those that exhibit chaotic behavior. Furthermore, due to the aperiodic nature of the motion and finite precision, a model based prediction may only have relatively high accuracy over a short time horizon, before significant growth of error occurs in the prediction.

An overall goal of this dissertation work is to help build the necessary theory and tools in order to use data-driven methods for studies of transient events and aperiodic motions. Specifically, the author has proposed and used stochastic phase interference based on data-driven modeling as an enabler for the formation of rogue waves, an extreme event in oceanic dynamics. Second, a new deep learning architecture is created to enable long-term forecasting of aperiodic motions of different systems. In the next section, background knowledge about complex systems and related prerequisites to understand the rest of the dissertation are briefly discussed.

The organization of this dissertation is discussed at the end of this chapter.

1.1 Complex system

Science has been used to understand the complexity in nature, as opposed to a traditional focus on unveiling the fundamental simplicity of system behaviors. The field of complexity science holds that the dynamics of various complex system builds upon universal principles, which can be used to explain a wide range of topics from plasma to ecology. It is hoped that knowledge and methodology learned from one field will cross-fertilize with important findings in other disparate systems. In this dissertation, both transient events and aperiodic motions can be categorized into complex dynamics. There is a lack of the universal definition of complex systems in the science community. Moreover, scientists with different backgrounds, from physicists to biologists, tend to have diverging definitions. But generally, a complex system consists of many interacting parts whose individual effects contribute to the global behavior, in short term or long term, in explicit or implicit format. The number of components in a complex system should be medium scale, or *mesoscopic*. Its size is larger than what a human can normally comprehend. But there is a limit to the total size or dimension. If there are too many components, even if they are strongly related, the system can be efficiently studied by the traditional thermodynamics approach. Usually it is intricate to understand how the small-scale, local effects propagate through system and aggregate into large-scale behavior, especially, involving a large number of interacting agents. The term *complexity*

comes from this perspective. Although it is hard to give a comprehensive definition of complex system, one can understand it through a list of its generic properties.

- [**Nonlinearity**] Nonlinearity is a defining feature of complex systems simply because even large scale linear systems can be solved exactly, by one way or another. A precise understanding of the physics modeled by linear equations allows for high-fidelity prediction of their future behaviors. Established mathematical tools are available to uncover the behavior of linear systems, which can be done carried out. On the other hand, complex systems are nonlinear and often times only an approximate or a numerical solution can be found.

- [**Dependence**] The interacting agents within the complex systems can be modeled as a network, which is a graph represented with nodes and links. Nodes stand for the agents and links are the relationships amongst them. Any independent agent can be removed from this graph without affecting the rest since there is no way to propagate their effect to the remaining part. A non-complex system is usually a collection of weakly, if any, connected components. The total number of components can be large, but still they can not be categorized as a complex system, since it is the number of links that exists in the graph that determines the system's complex behavior.

- [**Multiscale**] The notion of scale is strongly related to the size of each agent. Consider a migrating herd of gnu in the Serengeti Nation Park in Africa. One can track a single gnu as a unit (small scale) to study its local interaction with the neighbours or a family of gnus, including the father, mother, and

the children as a unit (large scale). In the latter case, the total number of units within the herd is smaller and the interactions between units can be modeled more uniformly and independently. This is in contrast to the first case. One can image that the children don't have too much freedom to explore but to follow their parents during the whole trip. Therefore, the dependence is stronger within the unit, whereas the interactions between different units do not possess that much reliance. Interesting behaviors are observed through different levels of scales. Complex systems are known for their rich multi-scale dynamics.

- [**Emergence**] This feature is a result of the two preceding characteristics. Because of the dependence on each other, agents can collectively display multi-scale dynamics. Some behaviors can only be observed at a larger scale within complex systems, and they can not be foretold by the close examination of each individual. This phenomenon is called *emergence*. One great example of emergence is *cellular automata* [9]. The interaction rules are only prescribed locally to the agents. However, many interesting global behaviors can be observed during the evolution of such system. In social science, the emergence of impromptu order is called *spontaneous order*, such as the market crash and the V formation of a flock of geese. In natural science, this is more often called as *self-organization*. Examples include the emergence of ordered-structure in micron-sized $Nb_3O_7(OH)$ cubes during a hydro-thermal treatment at 200°C [10].

- [**Adaption**] Perhaps the most important and interesting complex system is the adaptive system. The agents learn from their experience and adapt to each other or collectively to the environment. Every living creature in the world is a complex adaptive system, so is the creature's societal system. This is an active research area and many questions are still waiting to be answered.

There are still several features, which might not be universal to all the complex systems but still important, like nestedness, positive feedback loops, and so on. See reference [11] for review. Statistical mechanics and stochastic dynamics are two analytical tools that can be used for studying complex dynamics. Computers also play a crucial role in simulating the evolution of a system and thus enhancing our understanding of how the system works.

1.2 Dynamical system

One approach to describe a complex system is to use differential equations. The temporal effect in complex system is explicitly modeled as derivatives in the equations. Spatial variables can also be incorporated as independent variables. A system whose configuration can change with time is called a dynamical system. The space of the describing variables, or the possible states of this system, is called the *state space*. The mathematical definition of a dynamical system can be given in the basic form

$$\frac{dx}{dt} = f(x), \tag{1.1}$$

which illustrates that a change in the state variable $x \in \mathbb{R}^n$ depends on the current state x itself [12]. The linear form of the right-hand side (r.h.s.) of (1.1) has been extensively studied and the theory is complete, whereas the nonlinear form is the more commonly observed case in nature and usually with interesting behaviors [13, 14].

Most dynamical systems evolve in a bounded state space as $t \to \infty$, without which any infinitesimal perturbation to the original state can lead to an intractable divergence. Such bounded region to which trajectories are attracted within the state space $\mathbb{R}^n$ is called an *attractor* [15]. One can readily call the attractor forward-invariant since the system stays on this attractor as time unfolds on the positive side.

A more clear understanding of system solutions involves studying the change rate of x, or $\frac{dx}{dt}$. If $\frac{dx}{dt} = 0$, then there are solutions $x^* \in \mathbb{R}^n$ which satisfy $f(x^*) = 0$. Such solutions are called *fixed points*. When the trajectory starts from fixed point x^*, it will stay in that point and never move away given the zero change rate. For a linear dynamical system, there can be only one global fixed point. However, for a nonlinear dynamical system, there can exist multiple fixed points. The property of each fixed point is determined by the signs of the change rate $\frac{dx}{dt}$ and also the state x. In Figures 1.1 and 1.2, some basic notions associated with fixed points are illustrated.

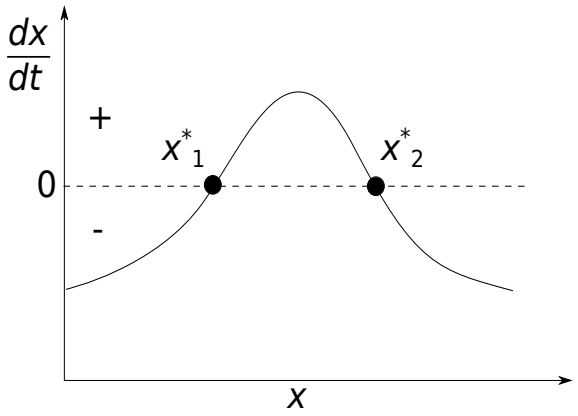

Figure 1.1: Diagram of stable and unstable fixed points based on the sign of $\frac{dx}{dt}$. Consider the 1-D case ($n = 1$ and $x \in \mathbb{R}$). If $x > x_2^*, \frac{dx}{dt} < 0$ and $x < x_2^*, \frac{dx}{dt} > 0$, then any perturbation to x_2^* will decay since $x \to x_2^*$ as $t \to \infty$. On the other hand, If $x < x_1^*, \frac{dx}{dt} < 0$ and $x > x_1^*, \frac{dx}{dt} > 0$, then any perturbation to x_1^* will push it away. Fixed point in the first scenario is called a *stable* fixed point and the second one is referred to as being *unstable*.

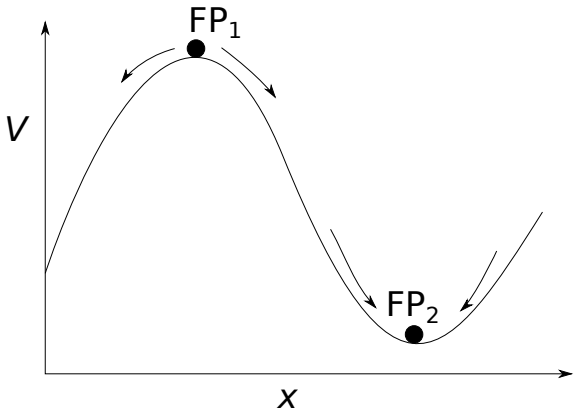

Figure 1.2: Diagram of potential valley of V associated with (1.1) where $\frac{dV}{dx} = -f(x)$. FP_1 can roll down from the top of the hill and any perturbation will destabilize the equilibrium; FP_2 lies on the valley and the state x will come back to FP_2 regardless of any local perturbation.

One may expect the transition of the state x from fixed point 1 (FP_1) to fixed

point 2 (FP_2) involves some overshoot of FP_2 to the right due to inertia. However, this is not the case, since there is no introduction of acceleration, $\frac{d^2x}{dt^2}$. The second-order derivative with respect to the state x can be regarded as assigning a new state variable $y = \frac{dx}{dt}$, thus transforming the original 1-D case into 2-D. Consequently, the system exhibits more familiar Newtonian dynamics with the inertia involved. The transition from FP_1 to FP_2 enables the possibility of oscillations around the stable fixed point. If no damping, or energy loss is considered, the oscillation in the state space is on a closed orbit and the total system energy is preserved.

1.3 Chaos

In the twentieth-century, quantum mechanics and relativity theory could be said to have started the physics revolution, which was all about simplicity and consistency, despite the quantum jumps. The primary tool was calculus and the final expression was field theory [16]. Chaos has revolutionized and ignited the twentieth-first century. It is all about complexity and a major tool for understanding this behavior has been super-computers. The final expression remains to be found, although artificial intelligence (AI) appears to be promising. Chaos can manifest itself both in space and time. In space, a chaotic object is called fractal if its geometric figure does not become simpler when one zooms it in a finer-scale, which simply implies that it is not smooth, such as a Cantor set [17] and a Sierpinski triangle [18]. Fractal not only exists in mathematics, but also in nature. A mountain range, a coastline, a human body, a fern leaf, a earthquake fault, even the cosmos

itself, can be considered as fractals. One is living in the world of fractals, which surprisingly occupies the whole universe.

Chaos in time is more often studied and time domain behavior is where the name is derived from [19]. A salient feature of temporal chaos is the *sensitivity to initial conditions*. This means that if a chaotic system is initiated from two extremely close starting points in the state space, then the two initiated trajectories will eventually diverge from each other as time goes on due to the existence of chaos. Edward Lorenz [20], who discovered the sensitivity to initial conditions, described temporal chaos as the "butterfly effect". A butterfly flapping its wings in Brazil can eventually lead to a tornado in Texas a month later on. The concept of "temporal chaos" is opposite in notion to *integrability* in classical mechanics. An integrable system is at most multi-periodic whose variables are changing periodically in time, although the motions can be at different frequencies. Most systems in classical mechanics textbooks are considered as integrable, such as the Kepler system and harmonic oscillators. However, starting from the late 1960s, scientists and engineers started to realize the prevalence of chaotic systems around us.

Chaos mitigates the dominant role of reductionism in science, since a finer scale of examination is not sufficient to identify principles and predict future behaviors of chaotic systems. The determination of a fine scale requires a even finer scale. This process goes on ad infinitum. The assumption that systems can be understood well by dividing its parts to a small scale and conquering them separately collapses. Indeed, any minuscule uncertainty in a chaotic system would eventually lead one to lose all useful knowledge about the system. *While a precise knowledge of the present*

can determine the future precisely, an approximate knowledge of the present cannot determine the future approximately, according to Edward Lorenz.

The connection between spatial features and temporal chaos can be seen with a chaotic dynamical system. Consider a closed region in the state space, consisting of an infinite number of initial conditions. Now integrate this system according to the governing laws, or equations, in time for a long period. During the evolution, all the initial points would have moved to other places in the state space. Due to chaos, the initial closed region gets transformed to a fractal in the state space after a long time.

Simple dynamical systems can display chaotic behaviors, which is contrary to the mundane thought that simple questions must have simple answers. An essential ingredient for the generation mechanism of chaos is nonlinearity. Most linear equations are truly "simple" systems, meaning that there exists a general method that can be used to solve them exactly with ease. If one knows a phenomenon can be described by linear equations, it is expected that their future behaviors can be predicted precisely. On the other hand, only a fraction of nonlinear dynamical systems can only be solved exactly, or approximately. Given the abundance of nonlinear behaviors in nature, most systems can only be simulated through computers or solved in a simplified version under certain assumptions.

The manner in which nonlinearity leads to the spatial features associated with chaos has been interpreted in terms of *stretching and folding* actions [15]. From a geometric perspective, the operations of stretching in state space gives rise to the divergence of neighbouring points and folding leads to the mixing of distant points.

It can be easily seen through a vivid example of a ball of dough that is persistently rolled out and then folded when making pasta. At the rolling out stage(stretching), two close points get separated. Next, two distant points can be struck together during folding. Nonlinearity comes into the picture during the folding. Linear equations can used to describe the stretching in the state space, but not the folding. It is the nonlinearity that helps with the folding.

1.4 Information theory and entropy

A second method used to study a complex system is notion of probability within the domain of information theory. Probability is about how to draw a useful conclusion from empirical evidence given the incomplete knowledge of all details of a system. This coincides with the study of complex systems in that it is the global behavior and the collective property of all agents that arouse the interest in studying and predicting such systems, regardless of the details at most times. The goal to understand and incorporate all of the details will be quite ambitious indeed.

With information theory, one studies how to quantify, store, and communicate the information. It was first proposed by Claude E. Shannon [21]. It has played a vital role in modern information society, including unmanned lunar exploration, the invention of Internet, mobile communication, and countless other fields. This field is fundamental to many electrical engineering and computer science research areas. Many crucial concepts and ideas from information theory are used to specify distributions and differentiate one probability distribution from other probabilities.

For more details on information theory, the reader is referred to references [22, 23].

One important quantity in information theory is entropy, or Shannon entropy. It is used to specify how uncertain a random variable or process is. For example, a deterministic process has less entropy than a stochastic process since the outcome of the latter one can be more uncertain than the first one; it thus has larger entropy. Many measures are built upon the concept of entropy, such as mutual information, which can be used to reconstruct attractors based on Takens' embedding theorem [24], and Rényi entropy [25]. It is related to the second law in thermodynamics, according to which, the total entropy for an isolated system can never decrease, or stay the same at best for equilibrium. This denotes the arrow of time which points to the direction of increasing entropy irreversibly. From a dynamical system perspective, a reversible process can be represented by closed orbit in phase space. Whenever the orbit is not closed there is an increase in entropy.

There is the macroscopic definition of entropy, which is given by

$$dS = \frac{dQ}{T} \tag{1.2}$$

for a reversible process. $dS = dS_i + dS_e$ is the change in entropy, which is also equal to the sum of entropy change due to external source and internal processes; dQ is the change in heat; and T is the temperature of the system.

The second definition comes from the microscopic viewpoint,

$$S = k_B \ln \Omega, \tag{1.3}$$

where k_B is Boltzmann constant and Ω is the number of all possible states. For a system with few possible states, it is more likely to display order, whereas a system

with large number of possible states has a tendency to be more disordered, thus have a high entropy.

A generalized version of entropy for dynamical system was brought up by Alfréd Rényi in 1961 [25]. This version has the form

$$H_\alpha = \frac{1}{1-\alpha} \ln \sum_i p_i^\alpha, \tag{1.4}$$

where H_α is the Rényi entropy. p_i is the probability that the system is at state i. α can be used to adjust the relative importance of the less likely state in H_α overall. As α increases, those states will have less impact on H_α. When $\alpha = 0$, all states are treated equally.

The intuition behind entropy is that knowing an unlikely event can be more informative than a likely event. A event with high entropy simply means that the information contained in such event has more value than a low entropy event. Therefore, three major points can be concluded as follows:

- A guaranteed future event should have zero entropy, indicating that knowing the happening of this event can increase zero information.
- A rare or an extreme event should have large entropy and high information material.
- Entropy is additive for independent events. For example, tossing a dice twice with the same result 2 should have twice entropy than tossing a dice with 2 for a single time.

Mathematically, one can define the **self-information** of an event x=x to be

$$I(x) = -\log P(x), \tag{1.5}$$

where the log is in base e. The above definition satisfies the three points listed previously. The unit of $I(x)$ is commonly written as **nat**. The amount of information harnessed by observing one event happening with the probability of $\frac{1}{e}$ is called 1 nat. When one changes the base of log to 2, then the unit is canonically called *shannons*. Now, the information has been something quantify that can be measured based on the unit determined by the choice of the logarithm base.

If the event x follows a distribution $P(x)$, then the total amount of uncertainty can be defined as (**Shannon Entropy**)

$$H(x) = \mathbb{E}_{x\sim P} I(x) = -\mathbb{E}_{x\sim P} \log P(x), \tag{1.6}$$

where $\mathbb{E}_{x\sim P}$ is the expectation of x over the distribution $P(x)$. Conceptually, the Shannon entropy of a distribution denotes the expected amount of information in an event drawn from a certain probability distribution.

One can also use the information theory to study the differences between two different distributions. If one has two different distributions $P(x)$ and $Q(x)$ over the same random variable x, the **Kullback-Leibler(KL) divergence** can be used to calculate the "distance" between $P(x)$ and Q(x):

$$D_{KL}(P||Q) = \mathbb{E}_{x\sim P}\left[\log \frac{P(x)}{Q(x)}\right] = \mathbb{E}_{x\sim P}[\log P(x) - \log Q(x)]. \tag{1.7}$$

The KL divergence has several properties:

- It is non-negative.

- It is 0 if and only if P and Q are the same distributions for discrete variables.
- It is 0 if and only if P and Q are "almost everywhere" equal for continuous variables.
- It is asymmetrical; this is, $D_{KL}(P||Q) \neq D_{KL}(Q||P)$.

If one regards $P(x)$ as the true probability and $Q(x)$ is the one needed to be generated, then it is useful to use the following identity

$$H(P, Q) = H(P) + D_{KL}(P||Q) \tag{1.8}$$

to define the **cross-entropy** as

$$H(P, Q) = -\mathbb{E}_{x\sim P} \log(Q(x)). \tag{1.9}$$

Finding a distribution $Q(x)$ close to true probability $P(x)$ equals to minimizing the cross-entropy, or D_{KL}, since $H(P)$ will be fixed given $P(x)$ is true.

1.5 Outline

The rest of the dissertation is organized in the following manner. Chapter 2 is about the data-driven study on rogue waves, one of the most interesting transient events in nature. In this chapter, the author mainly deals with how to extract coherent structures in the formation of rogue wave from field measurement data based on stochastic interference of wave groups. Literature has been surveyed to provide a brief review about the study of ocean rogue waves from theoretical, experimental, and computational perspectives. A modified solution to the nonlinear Schrödinger

equation (NLSE) has been introduced by incorporating the stochastic phase dynamics in both oscillation and modulation of wave groups. The Monte Carlo (MC) simulation results agree well with observational data points in the North Sea. Besides, the long-crested wave simulations from the modified wave solution reflects the true underlying wave height distributions both from the experiments and other high-order computational results.

In Chapter 3, the author prepares the reader with several key concepts of deep learning. This includes the fundamental definition of learning and its relationships with data sets. Then, several important learning algorithms are discussed. After introducing the learning basics, one of the most important neural networks in sequential modeling, recurrent neural network, is briefly mentioned.

Chapter 4 is about predicting chaotic dynamics based on deep learning. First, the definition of probabilistic dynamical system is introduced from an optimization point of view. Second, the relationship between maximum likelihood and Kullback-Leibler divergence is made in terms of predicting time series in dynamical system. Then, the author gives a detailed explanation of the neural network that has been used to predict different chaotic systems. Finally, results from these systems are shown to bring forth the network's superior ability in long-term forecasting.

Chapter 5 follows the similar vein of the previous chapter, but with a focus on non-autonomous systems. Here, the author illustrates the ability of the aforementioned neural network in generating long-term forecasting for a forced Duffing oscillator.

The contributions in this dissertation are summarized in the last chapter, along

with some thoughts for future work. Appendixes on additional technical details and references are provided at the end of this dissertation.

Chapter 2: Extreme Wave Formation in Unidirectional Sea due to Stochastic Wave Phase Dynamics

In this chapter[1], the author considers a stochastic model based on the interaction and phase coupling amongst wave components that are modified envelope soliton solutions to the nonlinear Schrödinger equation. A probabilistic study is carried out and the resulting findings are compared with ocean wave field observations and laboratory experimental results. The wave height probability distribution obtained from the model is found to match well with prior data in the large wave height region. From the eigenvalue spectrum obtained from the Inverse Scattering Transform, it is revealed that the deep-water wave groups move at a speed different from the linear group speed, which justifies the inclusion of phase correction to the envelope solitary wave components. It is determined that phase synchronization amongst elementary solitary wave components can be critical for the formation of extreme waves in unidirectional sea states.

[1] *This chapter is based on the work contained in the publication: Wang, R. & Balachandran, B. (2018). Extreme wave formation in unidirectional sea due to stochastic wave phase dynamics. Physics Letters A, 382(28), 1864-1872.*

2.1 Literature review

Rogue waves have been described as waves that appear from nowhere and leave without a trace [26]. These extreme energy concentrations pose severe threats to maritime voyages and offshore operations [27]. Considerable work has been done on modeling and predicting rogue waves [28, 29]. Related efforts include the analytical work based on modulational instability [30, 31], experiments and field measurements on wave statistical properties, such as kurtosis and skewness of the underlying probability density function [32], and numerical computations of different sea state parameters [33]. Broadly speaking, there are different mechanisms that can be used to explain the occurrence of extreme waves, including nonlinear focusing, dispersive focusing, atmospheric forcing and so on (e.g, the review papers by Dysthe *et al.* [26] and Kharif and Pelinovsky [27]). Until now, it is widely recognized that the unidirectional sea state often favors extreme wave statistics, as claimed in most of the studies [2, 3, 34, 35].

The modulational instability (MI) is a well-recognized mechanism for generating large waves due to energy transfer amongst different modes. A mathematical model for explaining MI has been developed by Shabat and Zakharov [31]. This model, known as the nonlinear Schrödinger equation (NLSE), has been used to study the interplay between nonlinearity and dispersion of water waves. NLSE is integrable in 1D+1 and can be solved by using the Inverse Scattering Transform (IST). Several analytical solutions, such as solitons and breathers, have been regarded as the prototypes of rogue waves. However, there is no broad agreement on

which solution is the best candidate for a rogue wave, when considering different spatial and temporal periodicities [36–39].

The existence of steep solitary wave groups has been confirmed in laboratories and examined under different numerical frameworks. When transverse effect is insignificant, weakly nonlinear wave groups do exhibit structural stability without noticeable distortion in the event of collisions and these groups can propagate a long distance. Whereas in the case of large wave steepness; that is, relatively steep solitary groups, dispersion outweighs the self-focusing effect along the propagation direction. However, it has been confirmed through experiments that the envelope soliton solution to NLSE provides a rather accurate approximation to the long-time evolution of steep intense solitary wave groups up to a wave steepness of 0.3 [40].

Although a single steep solitary wave group can create a freak wave event, interactions amongst multiple moderate solitary wave groups improves the likelihood of extreme waves significantly, leading to a heavy tail distribution in the wave height statistics. Soliton synchronization has been proved as an effective way to generate localized high-amplitude waves in the system governed by the NLSE [41] and the modified KdV framework [42]. In the former framework, it has been indicated with the Darboux transformation method that the solitons can be synchronized to form a peak at the focusing point with the magnitude equal to that of the sum of interacting solitons [43, 44].

The effect of multiple soliton interactions strongly depends on the details of the collision process. Although an intersection of soliton trajectories is necessary but it is not sufficient for the efficient focusing. When approaching the focusing

point, the train of solitons should be positioned with descending group velocities; this allows farther solitons to overtake the nearer ones. In addition, they should have alternating phases [42]. By simply setting the positions and phases to be the same amongst soliton trains, one will not have amplitude synchronization since the nonlinear interaction process makes the trajectory of each soliton bend before reaching the focusing point [41]. Although the exact synchronization of amplitude requires further details, there are two essential ingredients for soliton synchronization, phase coherence during the synchronization and different group velocities for soliton collision [45, 46].

Sea waves are an example of inherently stochastic waves and they are often modeled as a combination of quasi-sinusoidal waves with independent random uniformly distributed phases, known as Gaussian sea, following earlier work [47]. Onorato *et al.* [2, 3] have performed three-dimensional random waves water basin experiments to study the free surface profile probability distributions based on the JONSWAP spectrum. Different degrees of directionality have been considered to study the effects of wave crest length. The results indicate that the probability distributions of the surface elevation of unidirectional waves deviate most from the Gaussian or near-Gaussian sea and the occurrence of rogue waves has increased significantly compared to short-crest sea. Gramstad and Trulsen [48] have claimed a similar finding that more rogue waves are generated in unidirectional seas.

Here, the author focuses on understanding how the introduction of phase interference and wave train modulation can enhance the possibility of extreme waves formations in unidirectional sea states. The rest of the chapter is organized as

follows. In the next section, the author describes the model construction as an extension of the envelope solitary wave solution to NLSE. Following that, in Section 2.3, the author presents the results obtained through the application of this model to North Sea Draupner events to demonstrate the validity of the described methodology. Statistical results obtained from large-scale simulations are also discussed in support of the proposed model.

2.2 Solitary wave model approximation

2.2.1 Nonlinear Schrödinger equation and fundamental solitary wave solution

The leading-order theory for the description of unidirectional gravity water wave nonlinear focusing is the classic cubic NLSE written for the complex wave envelope $A(x,t)$ as

$$A_t + c_g A_x + \frac{i}{4} c_g k_0^{-1} A_{xx} + \frac{i}{2} \omega_0 k_0^2 |A|^2 A = 0. \tag{2.1}$$

Here, ω_0 and k_0 are the dominant wave frequency and wavenumber, respectively, and $c_g = \omega_0/2k_0$ is the linear group velocity in deep water, with the dispersion relation

$$\omega_0 = \sqrt{g k_0}. \tag{2.2}$$

Both the surface elevation $\eta(x,t) = \mathrm{Re}\{A(x,t)e^{ik_0 x - iw_0 t}\}$ and velocity potential $\phi(x,z,t)$ are determined by the complex-valued function $A(x,t)$. The η and ϕ fields can be computed with high accuracy by including higher order nonlinear terms in

NLSE , such as the Dysthe equation. The fundamental envelope soliton, a solution to equation (2.1), is of the form [49]

$$A = a_0 \ \mathrm{sech}(\sqrt{2}a_0 k_0^2 (x - c_g t)) e^{-ia_0^2 k_0^2 \omega_0 t/4}, \tag{2.3}$$

where a_0 is the soliton amplitude. The envelope soliton given by equation (2.3) is propagated with the linear group velocity c_g. Different from transient wave groups, the envelope soliton consists of coherent wave harmonics that prevent the dispersion of the wave group. The Fourier spectrum of wave group (2.3) may be obtained as

$$\hat{A}(k,t) = \int_{-\infty}^{+\infty} A(x,t) e^{ikx} dx = F(k) e^{i\xi(t)}, \tag{2.4}$$

$$F(k) = \frac{\pi A_0}{\sqrt{2}k_0^2 a_0} \sinh(\frac{\pi k}{2\sqrt{(2)}k_0^2 a_0}), \tag{2.5}$$

$$\xi(t) = -kc_g t - \frac{k_0^2 a_0^2 \omega_0 t}{4}. \tag{2.6}$$

Hence, all Fourier modes have the same phases and the Fourier amplitudes $F(k)$ do not evolve in time for a single envelope soliton. However, within the framework of NLSE, envelope solitons (2.3) may interact amongst each other, and also with other quasi-linear waves. It is noted that equation (2.1) has high-order solutions such as the Peregrine soliton, Kuznetsov-Ma breather, and Akhmediev breather [50], which are the results of interactions involving envelope solitons (2.3) with background waves [51]. These high-order breathers have different characteristic group velocity than c_g and they are defined by the IST spectrum [46, 51]. Next, the author revisits the IST to examine the determination of the spectrum from the complex modulation amplitude based on NLSE.

2.2.2 Inverse scattering transform

Starting with (2.1), one can non-dimensionalize it by applying the following transformation

$$\psi = -\frac{k_0}{\sqrt{2}}A, \tau = \omega_0 t, X = -k_0 x - \frac{\omega_0}{2}t,$$

and obtain the following equation

$$i\psi_X + \psi_{\tau\tau} + 2|\psi|^2\psi = 0.$$

In order to reduce the number of symbols and keep the formula simple, one can still express the above equation based on the more traditional format as following

$$iA_x + A_{tt} + 2|A|^2A = 0. \tag{2.7}$$

This is the scaled, *time* nonlinear Schrödinger equation. It satisfies the compatibility condition of the following system of linear equations:

$$\mathbf{B}_t = \begin{pmatrix} -i\lambda & A \\ -A^* & i\lambda \end{pmatrix} \mathbf{B}, \tag{2.8}$$

$$\mathbf{B}_x = \begin{pmatrix} -2i\lambda^2 + i|A|^2 & iA_t + 2\lambda A \\ -iA_t^* - 2\lambda A^* & 2i\lambda^2 - i|A|^2 \end{pmatrix} \mathbf{B}, \tag{2.9}$$

where λ is a spectral parameter, $\mathbf{B}(x,t,\lambda)$ is a vector or matrix function, and A^* represents the complex conjugate of A. In fact, if one differentiates equations (2.8) and (2.9) with respect to x and t respectively, one can find that in order to force the right hand side to be equal to each other, the complex envelope function $A(x,t)$ must satisfy equation (2.7). In other words, equation (2.8) and (2.9) are compatible

with each other on the equation condition (2.7). The matrix operators in the above linear systems are called the *Lax pair* of equation (2.7) and these operators were first studied by Zakharov and Shabat [31]. Equation (2.8) is called the *Zakharov-Shabat* (ZS) *scattering problem*. The parameter λ, which lies in the complex plane, is such that $\lambda = \lambda_R + i\lambda_I$. Then, the λ_I can be interpreted as having the information about the amplitude of the unstable mode and λ_R can be interpreted as referring to the group velocity relative to the linear group velocity, which corresponds to λ located on the imaginary axis.

In most cases, the parameter λ can only be obtained through numerical computation. Equation (2.8) can be rewritten as the linear eigenvalue problem:

$$\begin{pmatrix} -\partial_t & A \\ A^* & \partial_t \end{pmatrix} \mathbf{B} = i\lambda \mathbf{B}. \tag{2.10}$$

Typically, one can discretize the matrix coefficients on the left-hand side (l.h.s) of the above equation by using the finite-difference (FD) scheme. It involves first truncating the temporal domain into a finite length and then assigning grid points evenly across the whole domain. After this, one can approximate the temporal derivatives ∂_t by using a specific finite difference such as the central differencing scheme. With this, (2.10) can be transformed into a matrix eigenvalue problem. One can use various types of algorithms to solve for the eigenvalue, such as the Arnoldi algorithm [52]. However the accuracy is bounded by the order of the FD method. Moreover, FD can generate spurious eigenfunctions even if the eigenvalues are approximately correct. In contrast, the Fourier collocation method (FCM) allows for a more reliable and accurate computation of eigenvalues and eigenvectors of

the above linear system compared to the finite difference methods (FDMs) [53]. Instead of approximating the ∂_t by finite differencing, FCM transforms the temporal derivatives into the Fourier space. So is the complex wave envelope function. The first step is also to confine the temporal domain to $[0, L]$, where L is the total length of the considered time interval. On this interval, one can express the eigenfunction $\mathbf{B} = (b_1, b_2)^T$ and the complex envelope function $A(x = 0, t)$ by Fourier series with $2N + 1$ modes

$$b_1(t) = \sum_{n=-N}^{N} a_{1,n} e^{ink_0 t}, \tag{2.11a}$$

$$b_2(t) = \sum_{n=-N}^{N} a_{2,n} e^{ink_0 t}, \tag{2.11b}$$

$$A(x = 0, t) = \sum_{n=-N}^{N} c_n e^{ink_0 t}, \tag{2.11c}$$

where $k_0 = 2\pi/L$. Putting the above expressions into (2.10), one gets

$$\begin{bmatrix} -\mathcal{N} & \mathcal{C} \\ \mathcal{C}^\dagger & \mathcal{N} \end{bmatrix} \begin{pmatrix} \mathcal{A}_1 \\ \mathcal{A}_2 \end{pmatrix} = [i\lambda] \begin{pmatrix} \mathcal{A}_1 \\ \mathcal{A}_2 \end{pmatrix}, \tag{2.12}$$

where

$$\mathcal{N} = ik_0 \mathrm{diag}(-N, -N + 1, \ldots, N - 1, N),$$

$$\mathcal{C} = \begin{pmatrix} c_0 & c_{-1} & \dots & c_{-N} & & & \\ c_1 & c_0 & c_{-1} & \ddots & \ddots & & \\ \vdots & c_1 & c_0 & \ddots & \ddots & \ddots & \\ c_N & \vdots & \ddots & \ddots & \ddots & \ddots & c_{-N} \\ & c_N & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & \ddots & \ddots & \ddots & \ddots & c_{-1} \\ & & & c_N & \dots & c_1 & c_0 \end{pmatrix},$$

$$\mathcal{A}_1 = (a_{1,-N}, a_{1,-N+1}, \dots, a_{1,N})^T,$$

and

$$\mathcal{A}_2 = (a_{2,-N}, a_{2,-N+1}, \dots, a_{2,N})^T.$$

By using FCM, the author expands the eigenfunction $\mathbf{B}$ into a Fourier series and convert the ZS system (2.10) into a discretized matrix eigenvalue problem for the Fourier coefficients of the eigenfunctions. Later, the author applies FCM to study the eigenvalues of the ZS system for the Draupner events in Section 2.3.1 and finds that the large unstable modes that have large λ_I deviate from each other in the group velocity, which is represented by different λ_R values in the spectra (For envelope soliton solution(2.3), λ is located on imaginary axis in the spectrum.). These unstable modes whose eigenvalues have non-zero real parts correspond to high-order breather structures [54]. Hence, simply from the envelope soliton solution, one cannot explain why they move at a speed different from the linear group speed. Therefore, the author proposes the following phase interference model to address this discrepancy.

2.2.3 Stochastic phase interference model

In order to explain and represent the high-order breather solutions to NLSE, consider the following heuristically constructed model to describe the water wave elevation $\eta(x,t)$:

$$\eta(x,t) = \sum_{i=1}^{N} \eta_i(x,t), \tag{2.13}$$

$$\eta_i(x,t) = \mathrm{Re}\{a_i \mathrm{sech}[\sqrt{2}a_i k_0^2(x - c_{gi}t) + \phi_{2i}(t)] e^{-ia_i^2 k_0^2 \omega_0 t/4} e^{ik_0 x - i\omega_0 t + i\phi_{1i}(t)}\}. \tag{2.14}$$

The author calls this wave element as *quasi-soliton*. Here, $\phi_{1i}(t)$ and $\phi_{2i}(t)$ are introduced as the random, phase interference variables to modify the original envelope soliton (2.3). N is the number of interfering waves $\eta_i(x,t)$. From (2.13),the method of superposition also applies here for consideration of the aggregate effect. The above introduced phase random variables are intended to take into account solitary wave phase interference and allow for variations in the linear wave group speed and phase speed. Note that in equation (2.14), the sech function corresponds to slow modulation and the exponential part contains the fast oscillation. Next, the author defines

$$\Theta_i = \sqrt{2}a_i k_0^2(x - c_{gi}t) + \phi_{2i}(t) \tag{2.15}$$

as the modulation phase and

$$\theta_i = k_0 x - \omega_0 t - \frac{a_i^2 k_0^2 \omega_0 t}{4} + \phi_{1i}(t) \tag{2.16}$$

as the oscillation phase. Then, one can obtain the group speed and phase speed after including the phase interference random variables from

$$\frac{\partial \Theta_i}{\partial t} = 0, \frac{\partial \theta_i}{\partial t} = 0, \tag{2.17}$$

since $\Theta_i = C_i (i = 1, \ldots, N)$ characterizes the soliton's propagation in position and $\theta_i = D_i (i = 1, \ldots, N)$ characterizes the phase evolution of the soliton. Both C_i and D_i are constants depending on the initial condition. When considering the asymptotic states of the soliton solutions as $t \to \pm\infty$, synchronizing solitons requires $C_i = 0$ and $D_i = \phi_c$, where ϕ_c is the common phase [41]. From the above equation(2.17), the phase velocity and group velocity have the modified solution

$$c'_{pi} = c_{pi} + \frac{a_i^2 k_0 \omega_0}{4} - \frac{1}{k_0}\frac{d\phi_{1i}}{dt}, \tag{2.18}$$

$$c'_{gi} = c_{gi} - \frac{1}{\sqrt{2} a_i k_0^2}\frac{d\phi_{2i}}{dt}. \tag{2.19}$$

In both equations (2.18) and (2.19), the first term follows from the linear dispersion relation $c_{pi} = \frac{\omega_0}{k_0}, c_{gi} = \frac{\omega_0}{2k_0}$and the rest is due to the phase interference and nonlinearity. Furthermore, the deep-water dispersion relation still holds here. Hence, it follows that

$$\frac{d\phi_{2i}}{dt} = \frac{\epsilon_i}{\sqrt{2}}\frac{d\phi_{1i}}{dt} - \frac{\epsilon_i^3 \omega_0}{4\sqrt{2}}, \tag{2.20}$$

where $\epsilon_i = k_0 a_i$ is the wave steepness and $\omega_0 = \sqrt{g k_0}$ is due to the dispersion relationship. The author wishes to examine the statistical property of the above stochastic model, which includes quasi-soliton interactions.

Let us suppose that one considers the time series of wave elevation recorded by a gauge at sea. As the time series is sampled at one location in space, one can set $x = 0$, which makes the model free from the deep-water dispersion relation. Moreover, the frequency ω_0 and time t can be absorbed into the phase random variables $\phi_1(t)$ and $\phi_2(t)$. Therefore, given the relation from equation (2.20), the problem of the resulting amplitude of interfering waves is mathematically equivalent to computing

the probability of the height a of one-dimensional random walks involving N steps, which is the number of interfering waves. To this end, the nonlinear interference model of quasi-solitons can be written as

$$\eta(t) = \sum_{i=1}^{N} a_i \text{sech}(\frac{\epsilon_i}{\sqrt{2}}\phi_i(t) - \frac{\epsilon_i^3\omega_0}{4\sqrt{2}}t + \psi_i)\cos(\phi_i(t)), \tag{2.21}$$

where N is the number of interfering waves and ψ_i is the phase integral constant related to equation (2.20). Without loss of generality, the author sets $\psi_i = 0$ and $\epsilon_i = k_0 a_0$ in what follows. a_0 is set to be constant since the statistical results of rogue waves are independent of the amplitude distribution [55]. From equation (2.21), it can be discerned that the wave motions are aligned in the order of wave steepness ϵ, with rapid varying harmonic oscillations on the scale t and slowly changing amplitude modulations on the scale ϵt. Given the periodicity of harmonic oscillation and non-periodic wave modulations, the author chooses $\phi_i(t)$ to be the univariate uncorrelated random phases $\phi_i(t) \in [0, R]$. It is remarked that R should be a relatively large value given the shape of sech function in order to allow for significant modulations on wave shape. Here, the author chooses R to be at least 30π.

2.3 Results

2.3.1 IST spectra of Draupner event

The author numerically computes the discrete eigenvalues of the ZS system (2.10) based on time series data associated with the Draupner events. These wave events, which are also known as the New Year wave events, were recorded at the

Draupner jacket platform in the North Sea on January 1, 1995, from 14:00 to 19:20 Universal Time Coordinated (UTC). The single extreme wave height is approximately 25.2 m and exceeds the significant wave height of 11.8 m by a factor of 2.13. A number of observation windows, each 20 minutes long of wave conditions, were obtained by using a laser device and these records were collected during the peak of the storm, which was estimated to last for 6 hours [56]. The conditions associated with the Draupner wave events are summarized in the following: i) large waves were transported from the northwest direction to the southeast with significant wave heights around 8 m on January 1; ii) small-scale, but strong polar low descended rapidly from the north direction to the south, constantly generating large waves with a strong background swell also moving in the same direction; and iii) this swell arrived at the latitude of Draupner platform at 15:00 UTC, when the extreme wave was recorded [57]. Therefore, the Draupner wave happened with the background of a strong unidirectional swell. Instead of following reference [58] to study the proximity of homoclinic solutions to the imaginary axis to elucidate the underlying structure of rogue wave, the wave elevation time series are used here to show the different group and phase speeds of unstable mode calculated by applying IST to justify the author's intent in introducing random phase angles into the model velocity equations (2.18) and (2.19). The author uses 2^{11} to 2^{15} Fourier modes to extract the eigenvalues. The results are shown from Figure 2.1 to Figure 2.6.

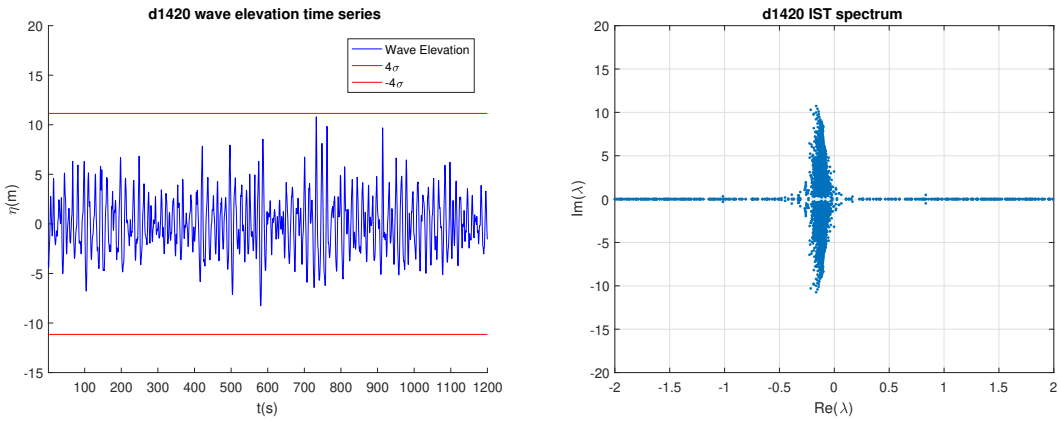

Figure 2.1: *left*: Wave elevation time series during Draupner event recording from UTC 14:00. $\pm 4\sigma$ (4 times standard deviation)values are shown as red lines to help visualize extreme wave height; *right*: corresponding inverse scattering spectrum calculated from the time series in left panel.

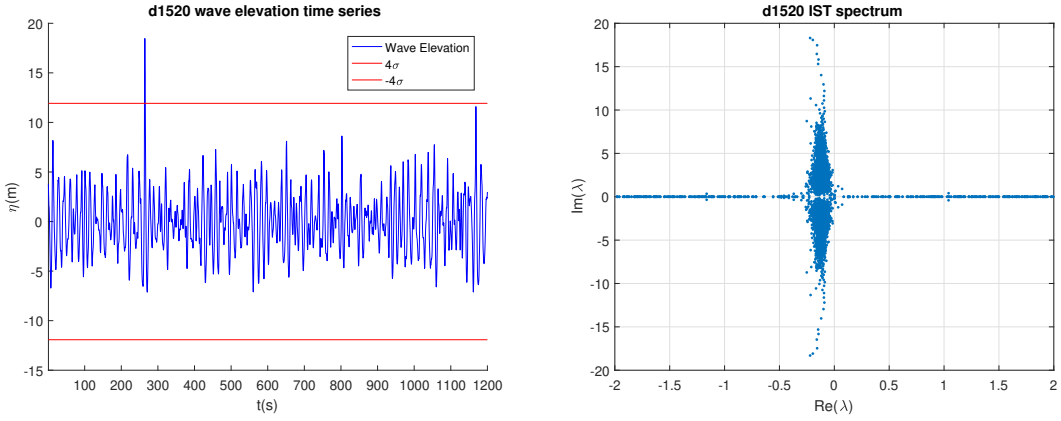

Figure 2.2: Wave elevation time series recording from UTC 15:00 and corresponding inverse scattering spectrum.

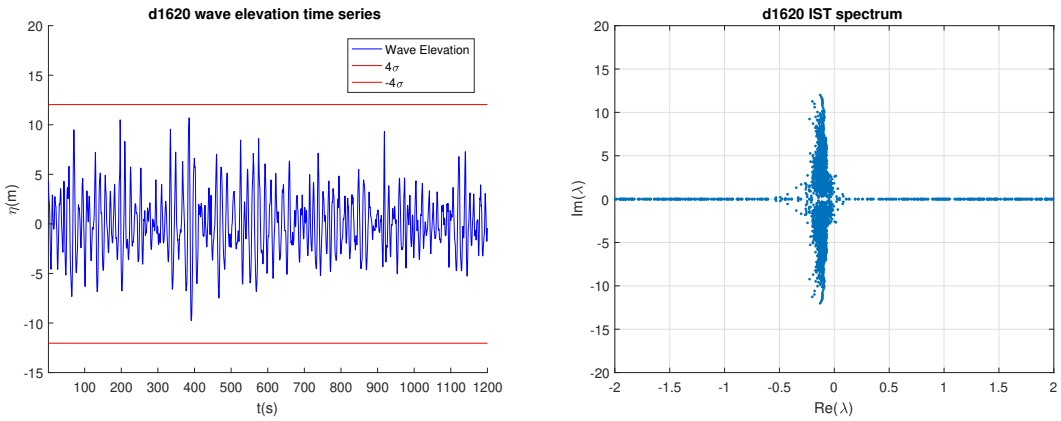

Figure 2.3: Wave elevation time series recording from UTC 16:00 and corresponding inverse scattering spectrum.

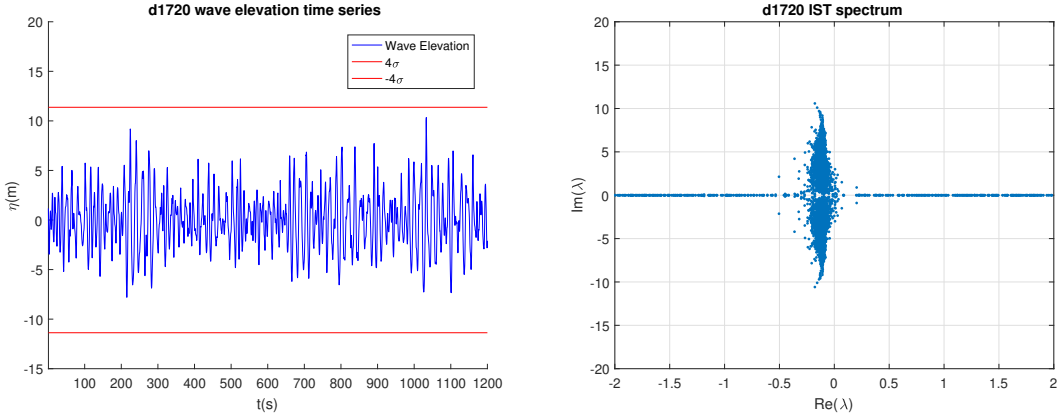

Figure 2.4: Wave elevation time series recording from UTC 17:00 and corresponding inverse scattering spectrum.

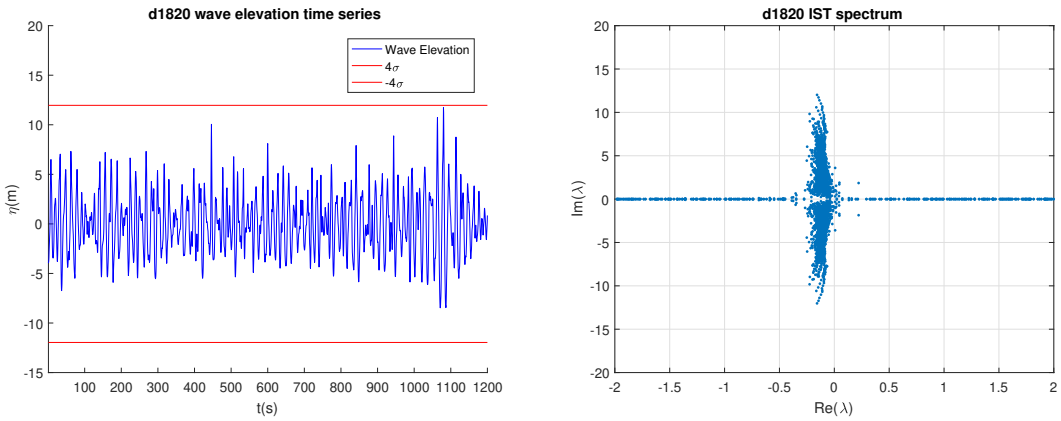

Figure 2.5: Wave elevation time series recording from UTC 18:00 and corresponding inverse scattering spectrum.

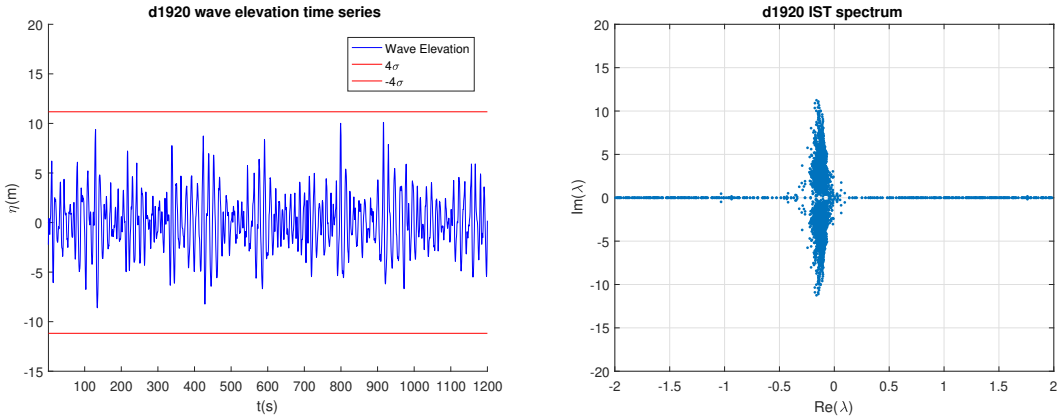

Figure 2.6: Wave elevation time series recording from UTC 19:00 and corresponding inverse scattering spectrum.

It is noted that the spectra are not centered at the imaginary axis. This is due to the fact that the author has applied the Hilbert transform to calculate the complex envelopes $A(x,t)$ from wave elevations $\eta(x,t)$, thus introducing an extra $\frac{\pi}{2}$ phase into the system. Nevertheless, this only results in shifting the whole spectra to the left of the imaginary axis. This does not affect the author's observations about the locations of eigenvalues in the complex plane, since they only consider the relative locations of the eigenvalues. The complex wave envelope $A(x,t)$ is

defined as

$$A(x,t) = \eta(x,t) + i\tilde{\eta}(x,t), \tag{2.22}$$

where $\tilde{\eta}(x,t)$ is the Hilbert transform of η that is given by

$$\tilde{\eta}(x,t) = \int_{-\infty}^{+\infty} \frac{\eta(x',t)}{x-x'} dx'. \tag{2.23}$$

Theoretically, the domain of IST should be an one-dimensional infinite line. But here in the discrete system, the author truncated the t-axis to a finite time series length L. The eigenvector $\mathbf{B} = (b_1(t), b_2(t))^T$ as well as the complex envelope $A(x,t)$ are represented by Fourier series with sufficiently large number of modes. Then, the Fourier expansions are substituted into equation (2.10) and the resulting discretized eigenvalue system is solved by using standard linear algebra methods [54, 59].

From Figure 2.1 to Figure 2.6, it can be clearly seen that large wave amplitude corresponds to the large imaginary part of the eigenvalues of the ZS system (*e.g.,* Figure 2.2). Moreover, each wave group in the wave record on the left panel manifests itself in the spectra on the right panel as bent curves, which is also predicted by the IST theory [51]. The fact that the dotted line in the complex plane bends to the left at various angles suggests that although eigenvalues in different wave groups can share the same imaginary part, the real parts can vary with distinct values. In other words, even though different wave groups possess the same modulation amplitude, the group speeds can be different from each other significantly. This cannot be explained by equation (2.3). However, this feature is captured in the author's model by including the phase dynamics through (2.18) and (2.19). Therefore, these quasi-solitons can possess different group speeds from the linear group propagation speed,

thus enabling collision and phase interference. Here, for purposes of illustration, the author has restricted the wave group shape to a sech function.

2.3.2 Application of stochastic model to Draupner event

The author validates their model by fitting the New Year Wave in the time windows of 200 seconds. In contrast to the traditional Fourier representation of irregular waves, which consists of a large number of elementary sinusoidal waves, here, the author wishes to represent the New Year Wave by as few interfering waves $\eta_i(x,t)$ as possible. To this end, they have tested different number of waves ranging from 4 to 20. It turns out that the minimum number to represent such an extreme wave case with great precision is $N = 6$, as shown in Figure 2.7. The curve fitting residuals are shown in Figure 2.8 and the precision is of the order of 10^{-7}. From Figure 2.9, one can see that the $\eta_i(x,t)$ are phase synchronized at the time $t = 100$ resulting in extreme wave heights, which is similar in manner to linear wave interaction based generation of large waves. This line of work is also similar to the work done by Birkholz *et al.* [55] who showed a reconstruction of the Draupner event by using a minimum of $N = 12$ elementary sinusoidal waves. They considered phase diffusion process in the linear interference model and used a penalty term to suppress the rapid temporal oscillations of the phase functions. They also mentioned that this may be indicative of the unaccounted nonlinear shaping in the immediate vicinity of the rogue wave. However, in the current nonlinear stochastic model, the author has reconstructed the considered rogue wave event by using half the number of

waves and even without penalizing the phase function. The author believes that the inclusion of envelope modulations could be a reason for why the current model works better. Again, modulations play an important role in the formation of large-wave events. The author's model is consistent with the NLSE theory given the inclusion of envelope modulations through the sech function and phase random variables to account for the group speed variations. Therefore, it is expected that the current approach would work better than linear interference models.

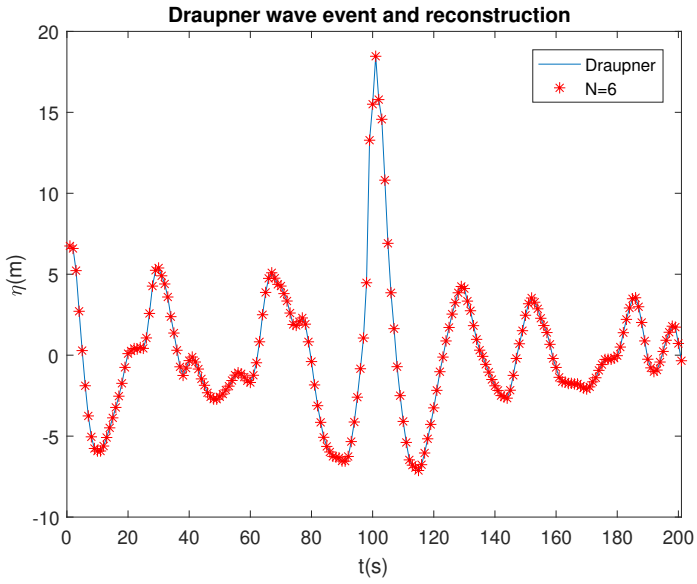

Figure 2.7: Draupner wave event (UTC 15:00, 1 Jan 1995) and the time series reconstruction through the stochastic model of the current work with 6 elementary nonlinear coherent components.

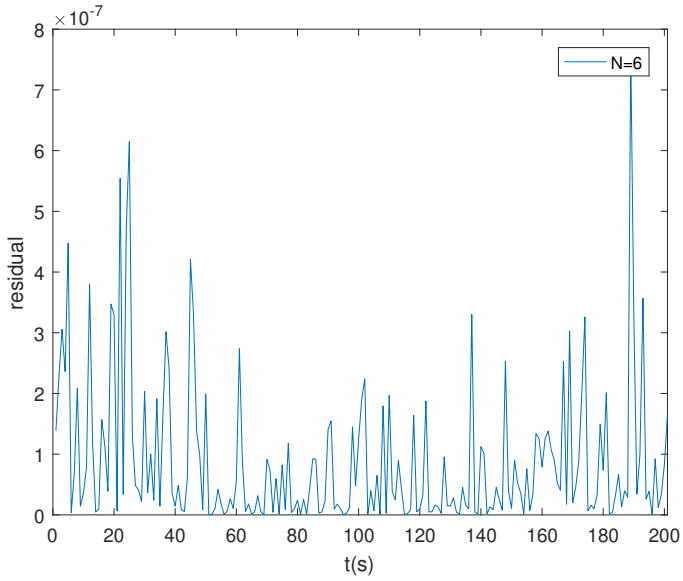

Figure 2.8: Reconstruction residual of the Draupner wave event with 6 elementary nonlinear coherent components.

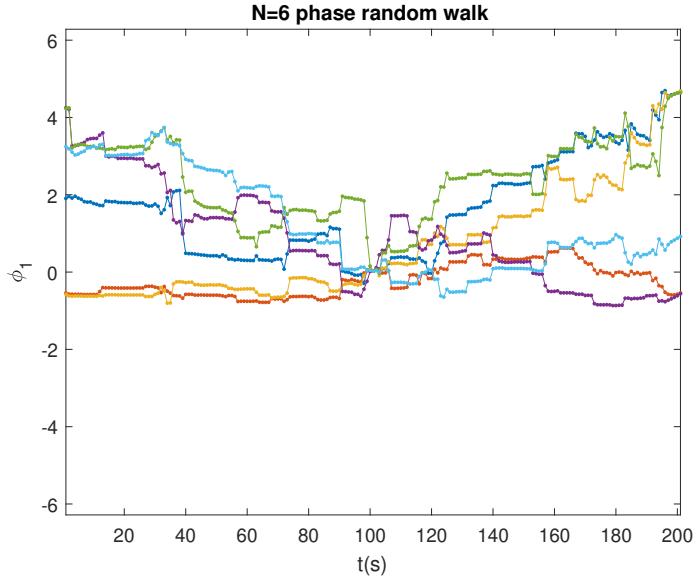

Figure 2.9: Evolution of phase angles of 6 elementary nonlinear coherent components used in the reconstruction. The synchronization of phases occurs at $t = 100$ seconds, when the extreme wave height is realized.

2.3.3 Extreme wave statistics based on stochastic model

In this section, the author has applied their model to study statistical distributions of ocean waves resulting from the interference of $N = 6$ waves with uniformly distributed phase variables. In Figure 2.10, 10^9 resulting wave heights have been simulated. Different curves in the plot correspond to different wave steepness $\epsilon = k_0 a_0$. Black dots are the probability of freak wave observations from Christou and Evans [1]. It is clear that when the ratio H/H_s is below 2, the wave statistics follows the $N = 6$ wave interference with wave steepness $\epsilon = 0.1$ and there is noticeable departure in the region $H/H_s \geq 2$, which is the defining region of rogue waves. This type of behavior echoes the fact that rogue waves occur mostly in rough sea states where the wave steepness ϵ is usually larger than that of calm sea state. Note that the observation data is included in the envelopes of $\epsilon = 0.1$ and $\epsilon = 0.4$, and 0.4 is the wave breaking limit. Again, the author has compared results from the current model with that of Birkholz *et al.* [55]. In contrast to this earlier work, wherein the number of interference waves N was increased to large value (*e.g.*, 100), through the use of the current stochastic interference model, it has been found that even with $N = 6$, the author is still able to capture the event well when reaching the rogue wave region. It is remarked that the smaller the number of the interfering waves involved, the higher the likelihood that extreme waves due to phase synchronization can occur at a certain location and time in a real sea state. The wave-crest probability distribution is also studied here. The second harmonic wave component is added to the wave elevation to account for the nonlinearity that pushes the crest

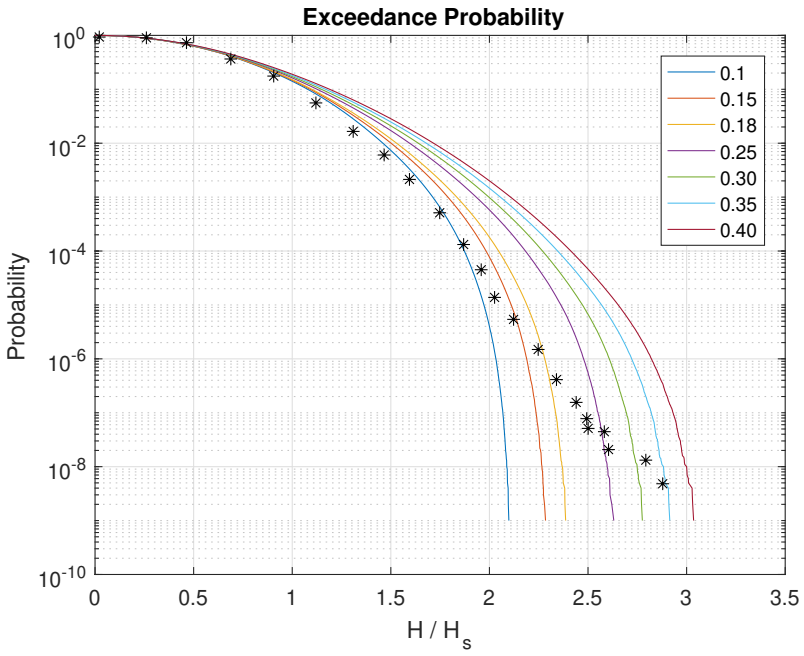

Figure 2.10: Probability of exceeding wave height from simulation of stochastic interference of six waves. Different lines corresponds to different wave steepness ϵ, ranging from 0.10 to 0.40. Black stars are used to represent field measurements of freak waves from reference [1].

up and flattens the trough. Therefore, the probability model for the wave elevation now becomes

$$\eta(t) = \sum_{i=1}^{N} a_i \text{sech}(\frac{\epsilon}{\sqrt{2}}\phi_i - \frac{\epsilon_i^3\omega_0}{4\sqrt{2}}t + \psi_i)(\cos(\phi_i) + \frac{1}{2}\epsilon\cos(2\phi_i)). \tag{2.24}$$

From Figure 2.11, it is seen that the model estimate is an underestimate of the probability of crest height in the region of small value of $\eta_c/4\sigma$, but strictly follows the distribution of long-crest wave in the rogue wave region of unidirectional sea. (*i.e.* $\eta_c/4\sigma \geq 1.25$ [26]), which is the region where the current stochastic model has been constructed to work in. The author has also investigated the kurtosis of the probability distribution of wave surface of the 10^9 runs with $N = 6$ interfering waves, as shown in Figure 2.12. The simulation results are compared with the work

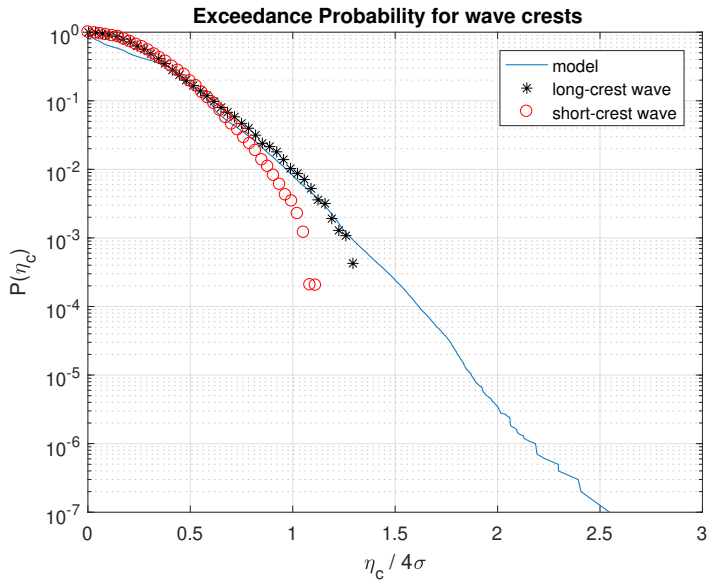

Figure 2.11: Exceedance probability for wave crests from simulation of stochastic interference of six waves. Black stars and red circles represent experimental data from reference [2].

of Onorato *et al.*[3]. Here, $\epsilon = \frac{k_p H_s}{2}$ is used to calculate the wave steepness for the irregular waves based on JONSWAP spectrum. From Figure 2.12, one can observe that the predicted kurtosis based on the model matches well with the results of experiments A and B, which are for short-crested wave and long-crested wave cases, respectively. It is clear that the model matches better with the long-crested wave case than the short-crested one, which again justifies that the current model's use for unidirectional sea states. Besides, the stochastic model also provides non-Gaussian distributions, as indicated by the value of kurtosis above 3. For comparison, the author used the same scheme to calculate the kurtosis of linear superposition of $N = 6$ sinusoidal waves and found that the kurtosis value is relatively stable and around 2.8, which is expected from linear wave theory.

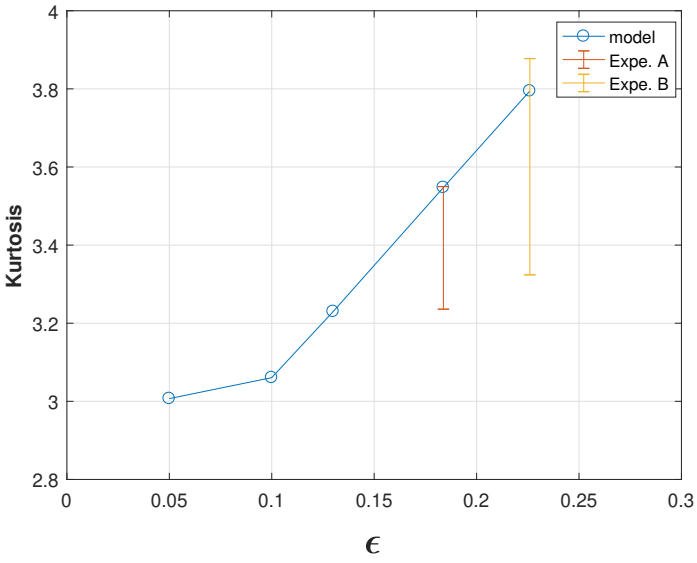

Figure 2.12: Kurtosis of probability density function from the simulation of stochastic interference of six waves with different initial wave steepness. Error bar data are from experiments [3].

Chapter 3: Fundamentals of Deep Neural Networks

3.1 Machine learning basics

This section contains many essential principles on machine learning, especially, deep learning to solve practical problems. For a comprehensive review on machine learning, see Murphy [60] and Bishop [61].

3.1.1 Learning algorithm

The author starts with the definition of a learning algorithm. A learning algorithm has the ability to update itself by learning from the data, be it real or artificial. Mitchell [62] specifies the key elements in a learning algorithm: *A computer program is said to learn from Experience E with respect to some class of tasks T and performance measure M, if its performance at tasks in T, as measured by M, improves with experience E.* Next, the author is going to elaborate on each of these elements.

3.1.1.1 Task T

When people think about machine learning, often times they are attracted to its extraordinary ability to solve so many different and difficult tasks that are otherwise unsolvable by traditional statistical techniques and approaches. Essentially, there are two main task T categories:

- **Classification** ($T1$): This is the most common and successful area where machine learning algorithm has been applied to. In this type of task (the author calls it $T1$), a learning algorithm is trained to figure out which of the k categories the input data belongs to. For simplicity, the author denotes the learning algorithm as f, input data as $x \in \mathbb{R}^n$, output from f as y. Then, $y = f(x) \in \{1, 2, \ldots, k\}$. A good learning algorithm can tag x with the label y successfully after learning from experience. Numerous applications fall under this category canopy. Iandola *et al.* [63] used small deep neural networks to categorize images from ImageNet at a high accuracy level. Bahdanau *et al.* [64] proposed neural machine translation to improve the translation performance and Google Translate directly benefits from this learning algorithm. Esteva *et al.* [65] demonstrated a learning algorithm capable of classifying skin cancer with a level of competence comparable to dermatologists. Usually $T1$ can also be named as object recognition and the associated techniques have been widely used in autonomous vehicles, recommender systems, auto-feeding, and so on.

- **Regression** ($T2$): In this type, the learning algorithm f is intended to learn from input $x \in \mathbb{R}^n$ and predict a numerical value $y \in \mathbb{R}$. Now, the output is continuous instead of being discrete as in the case of $T1$. This kind of technique can be applied in algorithmic trading to predict the future prices in stock market. The contribution in this dissertation work is strongly related to this kind of task.

Undoubtedly, there are other possible tasks, which have been studied in the past several years. However, most of them can be transformed into $T1$ or $T2$.

3.1.1.2 Measure M

The performance measure M is used to differentiate a good learning algorithm from a bad one. Usually, it is a designed quantity that highly depends on the specific applications. One straightforward metric will be the accuracy, denoted as M_a. It is remarked that the accuracy can have different meanings for $T1$ and $T2$. In the case of classification ($T1$), accuracy is the proportion of cases for which the learning algorithm generates the correct output. Therefore, M_a has a value between 0 and 1. On the other hand for $T2$, accuracy usually holds the meaning of closeness since one needs to measure the continuous variables in $\mathbb{R}^n$ space. Hence, M_a ranges from 0 to infinity. The smaller value M_a is, the more accurate a learning algorithm is. An example of M_a could be the $L2$ norm. The choice of M may seem arbitrary, but it is often difficult to choose the best one that maximizes the potential of a learning algorithm.

Oftentimes, one is interested in how well a learning algorithm performs with a data set that the algorithm has not been exposed to before. This data set is called *a test set*. If the learning algorithm also shows good performance, denoted by the value of M, then one has the confidence that the learning algorithm does learn the underlying patterns instead of simply memorizing the training data set. Consequently, it has more potential and capability to solve other similar problems.

3.1.1.3 Experience E

E is about the data set that is learned by an algorithm. There are several types of experiences of interest:

- **Unsupervised Learning (UL)**: In this type, the experience is simply based on the data set itself, without additional external inputs, such as labels. A learning algorithm is used to discover the pattern and connections within the data set itself. In some sense, the algorithm is learning unsupervisedly and it is on its own.

- **Supervised Learning (SL)**: In this type, the experience is labelled. The labels can be thought of as teachers who can show the algorithm what to do in order to maximize its performance.

- **Semi-supervised Learning (SSL)**: As the name suggests, part of the experience is labelled and the rest is untouched. The hope is that an algorithm will be trained jointly by a small amount of labelled experience and a large size unlabelled data set. The concept behind SSL is that labelling experience

will usually take tremendous effort; for example, marking of medical images and satellite data.

- **Reinforcement Learning (RL)**: This type of learning algorithm interacts with the environment and the feedback loop generates experiences at each cycle for an algorithm to learn. Rewards are provided to the algorithm with an aim to improve the overall performance in the long run.

3.2 Learning process

3.2.1 Gradient-based optimization

Deep learning algorithms involve solving optimization problems. Conceptually, an optimization algorithm alters input variables from an allowed set in order to maximize or minimize a real-valued function, or an *objective function*. Maximization and minimization are interchangeable since maximizing an objective function equals to minimizing the negative of the objective function.

Here, the author uses minimization optimization problem as an example to illustrate how to apply gradient-based approach in learning algorithms.

Suppose that one has a smooth objective function $y = f(\theta)$ where $y \in \mathbb{R}$ and $\theta \in \mathbb{R}^q$. The gradient of y with respect to θ is $\nabla_\theta f \in \mathbb{R}^q$. From calculus, one knows that the function value changes most rapidly in the direction of the gradient. In other words, if one adjusts the input variable θ in the opposite direction of the gradient, the objective function y will decrease most rapidly. When $\nabla_\theta f = 0$, there

is no directional information to decrease y further, thus achieving a local optimum. Mathematically, one should update the input variable θ iteratively as follows:

$$\theta^{k+1} = \theta^k - \alpha \nabla_{\theta^k} f, \tag{3.1a}$$

$$\text{until} \quad \nabla_{\theta^k} f = 0. \tag{3.1b}$$

Several remarks should be made here:

- For a quadratic function f, the Newton-like methods exist and they can provide a quadratic convergence rate towards the global minima during minimization [66].
- α is the learning rate, which determines the step size to update the input variable θ. For complex problems, it should be gradually decreased towards the end of the optimization problem.
- For a practical problem, the point with zero-gradient is not usually the global minima. This non-quadratic behavior complicates the optimization problem and the input variable θ is updated in a suboptimal manner.
- The computation of exact gradient $\nabla_\theta f$ is normally unfeasible and one uses the stochastic gradient descent to approximate it.

3.2.2 Stochastic gradient descent

The family of stochastic gradient descent (SGD) is the de facto most popular optimization algorithm used in deep learning. The important issue that SGD is used

to solve is the prohibitively large data set size when computing the gradient. The author will illustrate this idea through the following supervised learning example.

Suppose one has the training data set $Z = [(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)]$ with N independent pairs. The individual loss function, expressed in negative log-likelihood, is

$$L(x_i, y_i; \theta) = -\log p(y_i|x_i; \theta). \tag{3.2}$$

Then, the loss function over the whole data set Z can be expressed as the summation:

$$J(\theta) = \mathbb{E}_{x,y\sim p_{data}} L(x, y; \theta) = \frac{1}{N} \sum_{i=1}^{N} L(x_i, y_i; \theta). \tag{3.3}$$

Next, the gradient computed from the above loss function is

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{x,y\sim p_{data}} L(x, y; \theta), \tag{3.4a}$$

$$= \mathbb{E}_{x,y\sim p_{data}} \nabla_\theta L(x, y; \theta), \tag{3.4b}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta L(x_i, y_i; \theta). \tag{3.4c}$$

The evaluation of the above equation takes $O(N)$, which is typically difficult to compute given the large size of Z.

From (3.4b), one can approximate the gradient expectation by a small amount of elements from the training data set, instead of computing the full expectation. To be concrete, one can divide Z into batches of small groups; that is, $Z = [B_1, B_2, \ldots, B_m]$ where $B_i = [(x_{i1}, y_{i1}), (x_{i2}, y_{i2}), \ldots, (x_{ij}, y_{ij})]$. Now the gradient can be approximated as

$$g_i = \frac{1}{j} \nabla_\theta \sum_{k=1}^{j} L(x_{ik}, y_{ik}; \theta) \tag{3.5}$$

with $g_i \to \nabla_\theta J(\theta)$ as $j \to \infty$. Then the stochastic gradient descent can be written as

$$\theta \leftarrow \theta - \alpha g_i, \tag{3.6}$$

where g_i is also a stochastic variable. This is where SGD got its name from.

3.2.3 Adam

Adam is an adaptive learning rate SGD algorithm that has been widely used to train deep neural networks since its inception. Here, the author lists the major procedures when applying the Adam algorithm in Algorithm 1. For full details, the reader is referred to Kingma and Ba [67].

Adam optimizer is generally robust to the selection of hyperparameters, although the learning rate α should be tailored to each application.

3.3 Recurrent neural network

If the training data set is a sequence indexed by time t, then the proper neural network to learn it is the so-called *recurrent neural network* (RNN) [68]. The main distinguish aspect of RNN compared to multilayer networks is the sharing of parameters across different parts of the network. Sequences can have a variety of lengths. If one had separate parameters for each value of the time index, it would be impossible to scale up or down to different lengths of sequences, thus, reducing the generality of deep learning in solving sequential problems.

Algorithm 1: ADAM

step size $\alpha = 0.001$

exponential decay rates for moment estimates, with $\rho_1 = 0.9$ and $\rho_2 = 0.999$

$\delta = 10^{-8}$ for numerical stabilization

Initialize parameters θ ; initialize first and second moment variables $s = 0, r = 0$; initialize time step $t = 0$

while *stopping criterion not met* **do**

- sample a minibatch of m examples from training set Z
- compute an approximated gradient: $g = \frac{1}{m}\nabla_\theta \sum_i L(x_i, y_i; \theta)$
- $t = t + 1$
- update first moment estimate: $s = \rho_1 s + (1 - \rho_1)g$
- update second moment estimate: $r = \rho_2 r + (1 - \rho_2)g \bigodot g$
- correct bias in first moment: $\hat{s} = \frac{s}{1-\rho_1^t}$
- correct bias in second moment: $\hat{r} = \frac{r}{1-\rho_2^t}$
- Apply update: $\theta = \theta - \alpha\frac{\hat{s}}{\sqrt{\hat{r}}+\delta}$

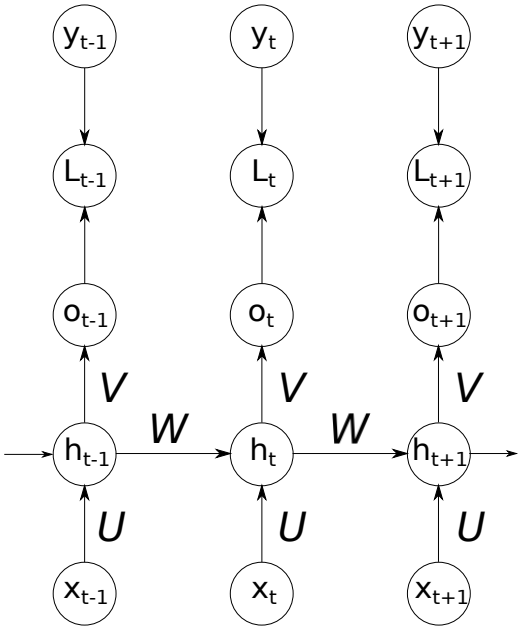

Figure 3.1: Recurrent neural network

Here the author briefly introduces how RNN works with sequence, see Figure 3.1. At every time step,

$$a_t = b + Wh_{t-1} + Ux_t, \tag{3.7}$$

$$h_t = \tanh(a_t), \tag{3.8}$$

$$o_t = c + Vh_t, \tag{3.9}$$

where x_t, h_t, and o_t are the input, hidden state, and output at time step t. tanh is hyperbolic tangent function, acting as a nonlinear activation function. W, U, V, b, and c are the neural network parameters, defined as θ. Assume that the input sequence is $X = \{x_1, \ldots, x_n\}$; then, the output $o_t, 1 \leq t \leq n$ is the cumulative summary of $\{x_1, \ldots, x_t\}$ up until time t. The total loss L for a given pair of input sequence X and $Y = \{y_1, \ldots, y_n\}$ would be the sum of losses over all time steps;

that is,

$$L(X, Y) = \sum_t L_t, \tag{3.10a}$$

$$= -\sum_t \log p_{model}(y_t, o_t), \tag{3.10b}$$

$$= -\sum_t \log p_{model}(y_t | x_1, \ldots, x_t). \tag{3.10c}$$

The above loss function assumes the negative log-likelihood in the model. Then, the next step is to calculate the partial derivatives with respect to the neural network parameters $\theta = \{W, U, V, b, c\}$ and apply the optimization algorithm like Adam in 3.2.2, to minimize the loss function (3.10) and update θ.

3.3.1 Back-propagation through time

Back-propagation through time (BPTT) is the technique to compute gradients in RNN. The author briefly illustrates the steps in the following. At time step t,

$$\frac{\partial L}{\partial L_t} = 1, \tag{3.11}$$

since the total loss L is the summation of individual loss at each time step. At the final time step $t = n$,

$$\nabla_{h_n} L = V^T \nabla_{o_t} L, \tag{3.12}$$

where the superscript T is the matrix transpose operation. One can calculate the derivatives backward from the end of the sequence $t = n - 1$ to $t = 1$. Note that

$h_t, t < n$ contributes both to o_t and h_{t+1}. Then the gradient can be computed as

$$\nabla_{h_t} L = \left(\frac{\partial h_{t+1}}{\partial h_t}\right)^T \nabla_{h_{t+1}} L + \left(\frac{\partial o_t}{\partial h_t}\right)^T \nabla_{o_t} L, \tag{3.13a}$$

$$= W^T(\nabla_{h_{t+1}} L)\ \mathrm{diag}(1 - (h_{t+1})^2) + V^T \nabla_{o_t} L, \tag{3.13b}$$

where $\mathrm{diag}(1-(h_{t+1})^2)$ is the diagonal matrix with elements $1-(h^i_{t+1})^2$. This is the Jacobian matrix of tanh of hidden unit i at time step $t+1$.

After one is ready with the gradients on the hidden nodes, next one can calculate the gradient of L with respect to parameters θ as follows

$$\nabla_c L = \sum_t \left(\frac{\partial o_t}{\partial c_t}\right)^T \nabla_{o_t} L = \sum_t \nabla_{o_t} L, \tag{3.14}$$

$$\nabla_b L = \sum_t \left(\frac{\partial h_t}{\partial b_t}\right)^T \nabla_{h_t} L = \sum_t\ \mathrm{diag}(1 - h_t^2) \nabla_{h_t} L, \tag{3.15}$$

$$\nabla_V L = \sum_t \sum_i \left(\frac{\partial L}{\partial o_t^i}\right) \nabla_V o_t^i = \sum_t (\nabla_{o_t} L)(h_t)^T, \tag{3.16}$$

$$\nabla_W L = \sum_t\ \mathrm{diag}(1 - h_t^2)(\nabla_{h_t} L) h_{t-1}^T, \tag{3.17}$$

$$\nabla_U L = \sum_t\ \mathrm{diag}(1 - h_t^2)(\nabla_{h_t} L) x_t^T. \tag{3.18}$$

This concludes the gradient calculation since the gradient of L with respect to input x_t will be zero.

Chapter 4: Neural Machine Based Forecasting of Chaotic Dynamics

In this chapter[1], the author explores a data-driven modeling approach to explore forecasting viability for systems that exhibit chaotic dynamics. Specifically, a deep recurrent neural network architecture, a neural machine, is constructed for forecasting temporal evolution of different chaotic systems. Data obtained from simulations with well known nonlinear dynamical system prototypes serve as training data for the chosen neural network. In practice, this simulation data may be replaced with data from the field. The trained system is studied to examine the forecasting ability. Two ordinary differential dynamical systems, namely, the Lorenz'63 system and the Lorenz'96 system, and a partial differential system, the Kuramoto-Sivashinsky equation, are studied and the numerical experiments conducted are presented here to demonstrate the forecasting viability of the constructed neural network.

[1] *This chapter is based on the work contained in the publication: Wang, R., Kalnay, E., & Balachandran, B. (2019). Neural machine based forecasting of chaotic dynamics. Nonlinear Dynamics (accepted)*

4.1 Literature review

There is tremendous interest in predicting the behaviour of complex dynamical systems, be it in nature (*e.g.*, ecology [69], ocean rogue waves [70]) or the human society (*e.g.*, financial market [71]). Several of these systems are chaotic, which means an initial misjudgment or error in the state of the system can grow exponentially in time. In addition, with finite precision, this exponential growth of the error can render inaccurate long-term forecasting. For traditional forecasting of chaotic systems, for instance, numerical weather forecasting, one requires two essential ingredients: i) an accurate estimation of the initial condition and ii) a good representative model which reflects the laws of physics. When either of them is not right, one ends up with a forecast that is suspect due to the chaotic dynamics. In recent decades, there has been a shift from physics-based model to data-driven modeling with advancements in sensors and data measurement equipment, as well as machine learning techniques [72, 73]. The combination of readily available data and sophisticated optimization algorithms makes deep learning, a popular machine learning approach, quite attractive for application to chaotic dynamical systems. Besides, with such a data-driven approach, one breaks the barriers between different scientific disciplines, as one eliminates the needs to develop various mathematical models for different dynamical systems, as long as these system evolutions can be described by a common mathematical structure. In this chapter, the author considers one neural network that can be used to describe the evolutions of three different chaotic systems, two of which are governed by ordinary differential equations (ODEs) and another that

is governed by partial differential equation (PDE).

Pathak *et al.* [74] used the echo-state network, or reservoir network, to study the dynamics of the Kuramoto-Sivashinsky equation [75] and examined the prediction abilities of this network under various parameter settings. It is mentioned that this network requires the monitoring of the whole past time series in order to predict the response at future steps. Vlachas *et al.* [76] used Long Short-Term Memory (LSTM) [77] networks to forecast the responses of reduced-order dynamical systems. In this chapter, the author proposes a deep recurrent neural network, which also consists of LSTMs, but with an inhibitor mechanism. By introducing this mechanism, the author is able to forecast the long-term responses of chaotic systems, such as the Lorenz'96 system [78–80] and the Kuramoto-Sivashinsky equation.

The rest of this chapter is organized in the following manner. In next section, the author briefly introduces the several chaotic systems which will be applied to test the forecast ability of the neural machine. In Section III, the author provides a probabilistic interpretation of the data-driven approach with regard to prediction of the future responses of chaotic dynamical systems. The details of the proposed deep recurrent neural network are given in Section IV. Finally, the author presents results obtained through the application of neural machine towards forecasting of chaotic responses. Also, the training details and additional results of the three numerical experiments are given in the appendixes at the end of this dissertation.

4.2 Background

4.2.1 Lorenz'63 system

The Lorenz'63 system [20], which is a set of coupled ordinary differential equations with three components, is given by

$$\begin{cases} \frac{\mathrm{d}x}{\mathrm{d}t} = \sigma(y - x), \\ \frac{\mathrm{d}y}{\mathrm{d}t} = x(\rho - z) - y, \\ \frac{\mathrm{d}z}{\mathrm{d}t} = xy - \beta z, \end{cases} \tag{4.1}$$

where x, y, and z are the state variables and $\sigma = 10, \beta = 8/3$, and $\rho = 28$. This model has been widely studied as a prototype for the demonstration of chaotic behavior and the characteristic attribute of the sensitivity to initial condition for a deterministic system [81, 82]. An infinitesimal perturbation to a chaotic trajectory of this system at any time during the evolution would give rise to the exponential divergence of this solution thereafter. The rate of divergence is commonly expressed by the Lyapunov exponent λ [12, 83]. Specifically, the distance $D(t)$ between two initially close trajectories with separation D_0 in state space can grow exponentially, assuming that the divergence can be treated within a linear approximation. This growth is given by

$$D(t) \approx e^{\lambda t} D_0. \tag{4.2}$$

For a multi-dimensional system, the rate of separation can be different for each projection of the initial perturbation vector on the chosen coordinate axes in the state space. Therefore, a spectrum of Lyapunov exponents along with the dimension

number is used for the response state of the dynamical system to show the overall divergence and contraction behavior in the state space of the dynamical system. In particular, the largest one, also known as Maximal Lyapunov Exponent (λ_M), is used as a measure of the level of unpredictability for a dynamical system. If λ_M is larger than 0, then the system response is labeled as being chaotic. The author follows earlier work [84] to compute λ_M. For the Lorenz'63 system, $\lambda_M = 0.9006$ and it matches well with the known value of 0.9056 [85]. The author uses $\lambda_M t$ as the non-dimensional Lyapunov time to demonstrate the prediction horizon of the neural machine. A typical response to the Lorenz'63 equation is shown in Figure 4.1.

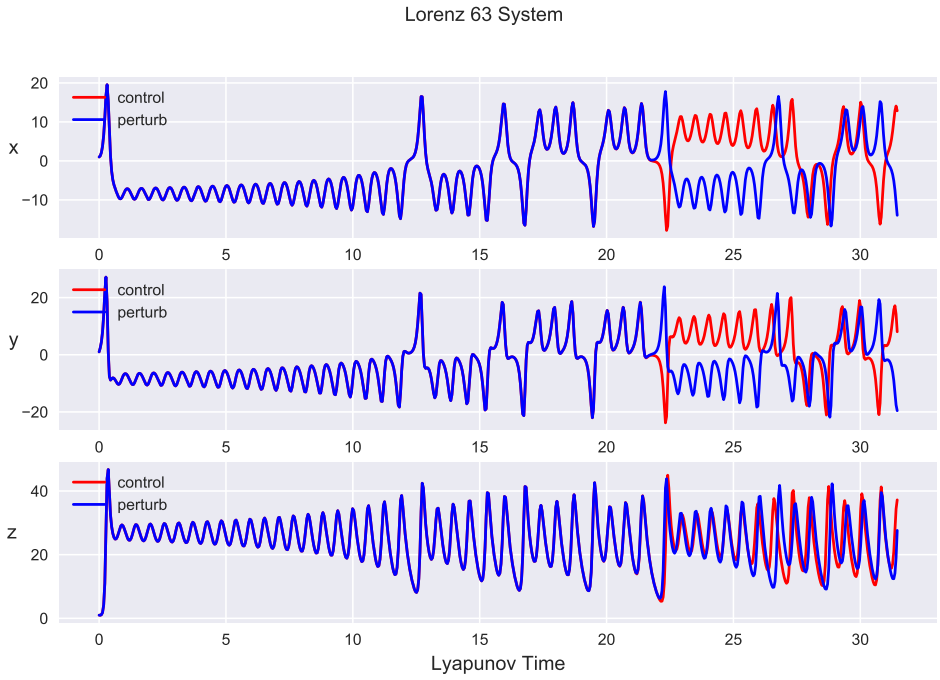

Figure 4.1: x, y, and z component time series of Lorenz'63 system. The two trajectories are initially separated by 10^{-15} units. The divergence is visible after 23 Lyapunov times.

4.2.2 Lorenz'96 system

This system can be written as [79]

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = (x_{i+1} - x_{i-2})x_{i-1} - x_i + F, \quad \text{for} \quad i = 1, \ldots, N, \tag{4.3}$$

with periodic boundary conditions $x_{-1} = x_{N-1}, x_0 = x_N, x_{N+1} = x_1$. Here, x_i is the state variable of the system and F is an external forcing. This model is meant to replicate the dynamic behaviour of an unspecified meteorological quantity x at M equidistant grid points along a latitude circle. The author numerically integrates (4.3) with a time step 0.05 time units, which is equivalent to 6 hours in practice by assuming the characteristic dissipation time scale of 5 days; see references [79, 80] for details.

For the case considered here, the author sets $F = 8$ and $N = 48$ to demonstrate the forecasting ability of the neural machine. Following the same approach as before for determining the Maximal Lyapunov Exponent, it is determined that $\lambda_M = 1.73$. This value is similar to the value obtained based on QR approaches [86]. A typical scalar field $x_i, i = 1, \ldots, N = 48$ to the Lorenz'96 equation is shown in Figure 4.2.

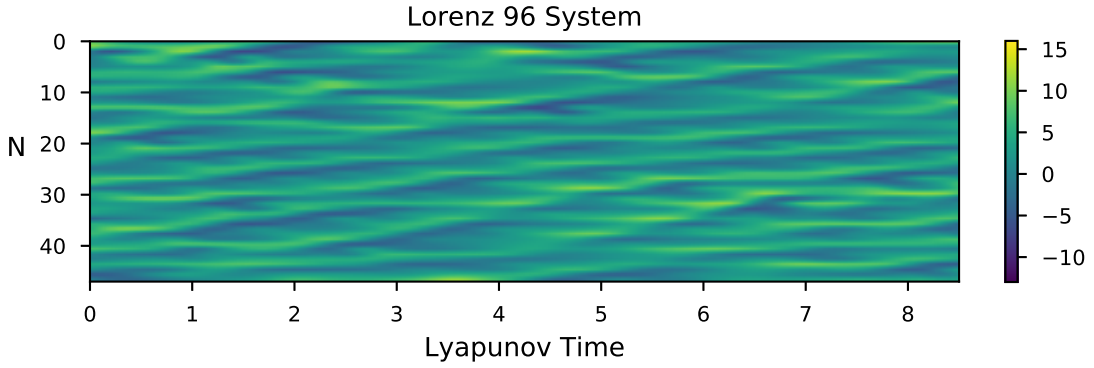

Figure 4.2: Scalar field of the Lorenz'96 equation with periodic boundary conditions. External forcing term F is 8, which commonly leads to chaotic behaviors. Vertical axis is the grid of $x_i, i = 1, \ldots, N = 48$. Horizontal axis has the scale of the non-dimensional Lyapunov time which is the product of the maximal Lyapunov exponent and time. Colorbar denotes the magnitude of the scalar value x_i, ranging from -10 to 15. This system represents the dynamical response of an atmospheric quantity, such as temperature or humidity, at equally spaced grid points in a latitude circle around the earth. It includes the effects of quadratic nonlinearity, dissipation and external forcing.

4.2.3 Kuramoto-Sivashinsky system

In addition, the author considers the homogeneous Kuramoto-Sivashinsky (KS) equation, which is given by

$$u_t + uu_x = -u_{xx} - u_{xxxx}, \quad x \in [0, D), \tag{4.4}$$

where the scalar field $u(x,t)$ is periodic in the domain $[0, D)$. This equation shares similarity with Burgers' equation but has more complicated and interesting behavior due to the presence of second-order and fourth-order spatial derivatives. The second-order derivative acts as a energy source, which can destabilize the scalar field. However, the nonlinear term uu_x can help transferring the energy from a low wavenumber mode to a high wavenumber mode, where the fourth-order derivative term dominates. This can be shown through the dispersion relation determined from the linear part of KS equation.

It has been proven that a unique solution to (4.4) exists and remains bounded as $t \to \infty$ for all D-periodic initial data, where D is the domain length. The solution can highly vary in behavior. It can be spatio-temporal chaos, depending on the amplitude of the initial data and on D. It is remarked that the dimension of the attractor is linearly correlated with the domain length D [87].

A dimension length $D = 35$ was chosen and the initial condition was determined to be

$$u(x, 0) = 0.6(-1 + 2 \times \text{rand}(M)),$$

where $M = 64$ is the discretized dimension of (4.4). Now, the author has used

the Exponential Time Differencing Runge-Kutta 4th-order (ETDRK4) method to numerically integrate one step forward in time (See Appendix C). Note that the integration takes place in the Fourier space. After solving for enough time windows, the transient solutions to (4.4) are discarded and only the steady part is used to train the neural machine. A typical scalar field $u(x,t)$ to the Kuramoto-Sivashinsky equation is shown in Figure 4.3.

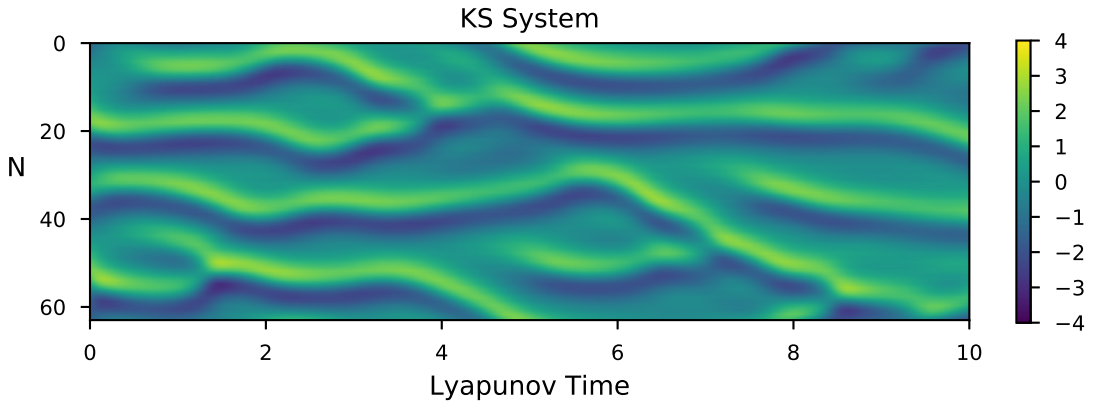

Figure 4.3: Scalar field of the Kuramoto-Sivashinsky equation. Vertical axis is the spatial domain, discretized with the grid size of 64. Horizontal axis shows the non-dimensional Lyapunov time which is the product of the maximal Lyapunov exponent and time. Colorbar denotes the magnitude of the scalar value $u(x,t)$, ranging from -4 to 4.

4.3 Methodology

Next, the author briefly introduces the application of the data-driven approach in forecasting the future responses of a dynamical system (DS). As earlier mentioned, the goal is to build up a single surrogate model $G(\theta)$ that can replicate the dynamical behavior of different systems. A representative dynamical system is described by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}; \zeta), \quad \mathbf{x} \in \mathbb{R}^m, \zeta \in \mathbb{R}^q, \tag{4.5}$$

where $\mathbf{x}$ is the state vector of dimension m and ζ is the parameter vector of dimension q. $\mathbf{f}$ is a deterministic function of the states and the parameters. Starting from the initial value $\mathbf{x}(t = 0)$ by numerically integrating (4.5) for $t > 0$, one can obtain the exact future states $\mathbf{x}_t$. In the case of discrete, integer-value times, a dynamical system can be written as the map [12]:

$$\mathbf{x}_{n+1} = \mathbf{F}(\mathbf{x}_n; \zeta), \tag{4.6a}$$

$$= \mathbf{F}(\mathbf{F}(\mathbf{x}_{n-1}; \zeta); \zeta), \tag{4.6b}$$

$$= \mathbf{F}(\mathbf{F}(\ldots \mathbf{F}(\mathbf{x}_0; \zeta) \ldots; \zeta); \zeta), \tag{4.6c}$$

where $\mathbf{F} : \mathbb{R}^m \to \mathbb{R}^m$ is the state transition mapping function. Note that the next time state variable $\mathbf{x}_{n+1}$ depends on $\mathbf{x}_n$, regardless of the previous histories, bearing similarity with the Markov property.

Generally speaking, there are two stages associated with the data-driven prediction, namely, a training stage and an inferring stage. During the training stage, one applies numerical algorithms, like gradient descent [88], to adjust surrogate

model parameters θ in order to better represent the training data set from the DS. The trained model is denoted as $G(\tilde{\theta})$. Then, as for the second stage, this surrogate model is tested on a new data set that has not been previously seen by it in order to test its inference capacity. The prediction value is distinguished from the true value by using the symbol $\widehat{\bullet}$.

4.3.1 Probabilistic dynamical system

From a probabilistic perspective, consider the conditional probability $P(\mathbf{Y}|\mathbf{X})$, where $\mathbf{X} = \{x_1, \ldots, x_{n_x}\}$, $x_i \in \mathbb{R}^m$ is the multivariate input sequence of dimension m with length n_x and $\mathbf{Y} = \{y_1, y_2, \ldots, y_{n_y}\}, y_i \in \mathbb{R}^m$ is the output sequence of the same dimension with length n_y, from the same discrete dynamical system (4.6a). In the context of forecasting the behaviour of this dynamical system, $\mathbf{Y}$ is the future time series that needs to be predicted based on the preceding input time series $\mathbf{X}$.

Let $\mathbf{Z} = \mathbf{Y}|\mathbf{X}$ be the event that $\mathbf{Y}$ happens after $\mathbf{X}$ and $P_m(\mathbf{Z}; \theta)$ be a family of probability distributions over the same parametric space indexed by θ. In this chapter, the author uses a deep recurrent neural network, parametrized by θ as the surrogate model $G(\theta)$ to determine the conditional probability $P_m(\mathbf{Z}; \theta)$, as an approximation to the true but unknown data-generating distribution $P_d(\mathbf{Z})$. If the time series of event $\mathbf{Z}$ is drawn from a dynamical system with certain initial condition, then the conditional probability $P_d(\mathbf{Z}) \equiv 1$ due to the determinism. However, from the surrogate model, $P_m(\mathbf{Z}; \theta)$ can only be optimized to be close to 1 by adjusting the value of θ without necessarily achieving the global optimum,

especially, for complex dynamical systems. To understand how one transforms a deterministic problem into a probabilistic one, there are two viewpoints to consider.

First, following the maximum likelihood principle [89], the estimator for θ can be defined as

$$\tilde{\theta} = \underset{\theta}{\text{argmax}} \; P_m(\mathbb{Z}; \theta), \tag{4.7a}$$

$$= \underset{\theta}{\text{argmax}} \; \prod_{k=1}^{r} P_m(\mathbf{Z}^k; \theta), \tag{4.7b}$$

where $\mathbb{Z} = \{\mathbf{Z}^k, k = 1, \ldots, r\}$ are independent sequences with sample size r determined by the true but unknown $P_d(\mathbf{Z})$. The above equation (4.7b) can be problematic in terms of numerical computation. Due to the determination of the product over many probabilities which all vary from 0 to 1, it is prone to numerical underflow. Hence, it is more convenient to take the logarithm of both sides of the equation. This results in the following equivalent optimization problem:

$$\tilde{\theta} = \underset{\theta}{\text{argmax}} \; \sum_{k=1}^{r} \log \; P_m(\mathbf{Z}^k; \theta). \tag{4.8}$$

Typically, large value of sample size r can give a better estimation of θ, resulting in $P_m(\mathbf{Z}^k; \tilde{\theta}) \approx 1$. Therefore, the prediction of future response based on this surrogate model is more accurate. But in reality during the training stage, r is often limited and the probability distribution represented by $\mathbb{Z}$ is an empirical data generating distribution; that is labeled as $\tilde{P}_d(\mathbf{Z})$. As a result, equation (4.8) can be written as an expectation over the empirical distribution defined by the training data set:

$$\tilde{\theta} = \underset{\theta}{\text{argmax}} \; \mathbb{E}_{\mathbf{Z} \sim \tilde{P}_d} \log \; P_m(\mathbf{Z}; \theta). \tag{4.9}$$

The second viewpoint is related to the Kullback-Leibler divergence or KL divergence [90]. It is a measure of the distance between two different probability distributions. The KL divergence between $\tilde{P}_d$ defined by the training data set and P_m, which is generated from a surrogate model, is given by

$$D_{KL} = \mathbb{E}_{\mathbf{Z}\sim\tilde{P}_d}[\log\ \tilde{P}_d(\mathbf{Z}) - \log\ P_m(\mathbf{Z};\theta)], \tag{4.10a}$$

$$= \mathbb{E}_{\mathbf{Z}\sim\tilde{P}_d}\log\ \tilde{P}_d(\mathbf{Z}) - \mathbb{E}_{\mathbf{Z}\sim\tilde{P}_d}\log\ P_m(\mathbf{Z};\theta). \tag{4.10b}$$

The goal is to minimize D_{KL} by adjusting the model parameters in $G(\theta)$, thus affecting P_m. The first term in equation (4.10b) is only associated with the probability of generating certain input time series, not with the model. Hence, the estimation of θ should only come from the second term, which is

$$\tilde{\theta} = -\operatorname*{argmin}_{\theta} \mathbb{E}_{\mathbf{Z}\sim\tilde{P}_d}\log\ P_m(\mathbf{Z};\theta). \tag{4.11}$$

Comparing with the maximum likelihood principle from the first viewpoint, one can find that equations (4.9) and (4.11) are essentially the same.

4.3.2 Probability distributions and loss functions

Now, the author is going to discuss the relations between the surrogate model $G(\theta)$ and conditional probability P_m. As mentioned earlier, $G(\theta)$ is a deep recurrent neural network, which in essence is the following mapping function:

$$G(\mathbf{X};\theta) = \mathbf{Y}. \tag{4.12}$$

Again, $\mathbf{X}$ and $\mathbf{Y}$ are the historical time series and future time series generated from certain dynamical system in sequence, respectively. In reality, the mapping output

is $\widehat{\mathbf{Y}} = G(\mathbf{X}, \tilde{\theta})$, which is an approximation to the true target value $\mathbf{Y}$ with certain types of associated errors. Here, three types of error distributions corresponding to three different $P_m(\mathbf{Y}|\mathbf{X};\theta)$ and loss functions are considered, by using one time step univariate series x and y.

4.3.2.1 Type I: Gauss loss function

The error between the mapping output $\widehat{y}$ and the true output y is assumed to follow the Gaussian distribution

$$P_m^g(y|x;\theta) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(\widehat{y}-y)^2}{2\sigma^2}), \tag{4.13a}$$

$$= \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(G(x;\theta)-y)^2}{2\sigma^2}). \tag{4.13b}$$

where σ is the standard deviation of the error distribution. On substituting (4.13b) into equation (4.8) and only keeping the terms associated with θ, the result is

$$\tilde{\theta} = \underset{\theta}{\operatorname{argmax}} \sum_{k=1}^{r} -\frac{(G(x^k;\theta)-y^k)^2}{2\sigma^2}, \tag{4.14a}$$

$$= \underset{\theta}{\operatorname{argmin}} \sum_{k=1}^{r} (\widehat{y}^k - y^k)^2. \tag{4.14b}$$

As one may notice, equation (4.14b) can be used to minimize the mean square error between the model output $\widehat{y}$ and the true value y. In other words, if one attempts to use the mean square error as the loss function

$$\mathcal{L}(x, y, \theta) = \sum_{k=1}^{r} (\widehat{y}^k - y^k)^2, \tag{4.15}$$

during the training stage, it is essentially the same as implying that the model output $\widehat{y}$ predicted by $G(\theta)$ is the superposition of true value y and the Gaussian noise.

4.3.2.2 Type II: Laplace loss function

In this case, the error between the mapping output $\widehat{y}$ and the true output y is assumed to follow the Laplace distribution

$$P_m^l(y|x;\theta) = \frac{1}{2\beta}\exp(-\frac{|\widehat{y}-y|}{\beta}), \tag{4.16a}$$

$$= \frac{1}{2\beta}\exp(-\frac{|G(x;\theta)-y|}{\beta}), \tag{4.16b}$$

where β is a scale parameter. After substituting (4.16b) back into equation (4.8), the result is

$$\tilde{\theta} = \underset{\theta}{\operatorname{argmax}} \sum_{k=1}^{r} -\frac{|G(x^k;\theta)-y^k|}{\beta}, \tag{4.17a}$$

$$= \underset{\theta}{\operatorname{argmin}} \sum_{k=1}^{r} |\widehat{y}^k - y^k|. \tag{4.17b}$$

Equation (4.17b) can be used to minimize the mean absolute error between the model output $\widehat{y}$ and the true value y. Following along the same lines as for equation (4.15), the loss function can be defined as

$$\mathcal{L}(x,y,\theta) = \sum_{k=1}^{r} |\widehat{y}^k - y^k|. \tag{4.18}$$

4.3.2.3 Type III: Cauchy loss function

In this case, it is assumed that the error between mapping output $\widehat{y}$ and true output y is to follow the Cauchy distribution

$$P_m^c(y|x;\theta) = \frac{1}{\pi\gamma[1+(\frac{\widehat{y}-y}{\gamma})^2]}. \tag{4.19}$$

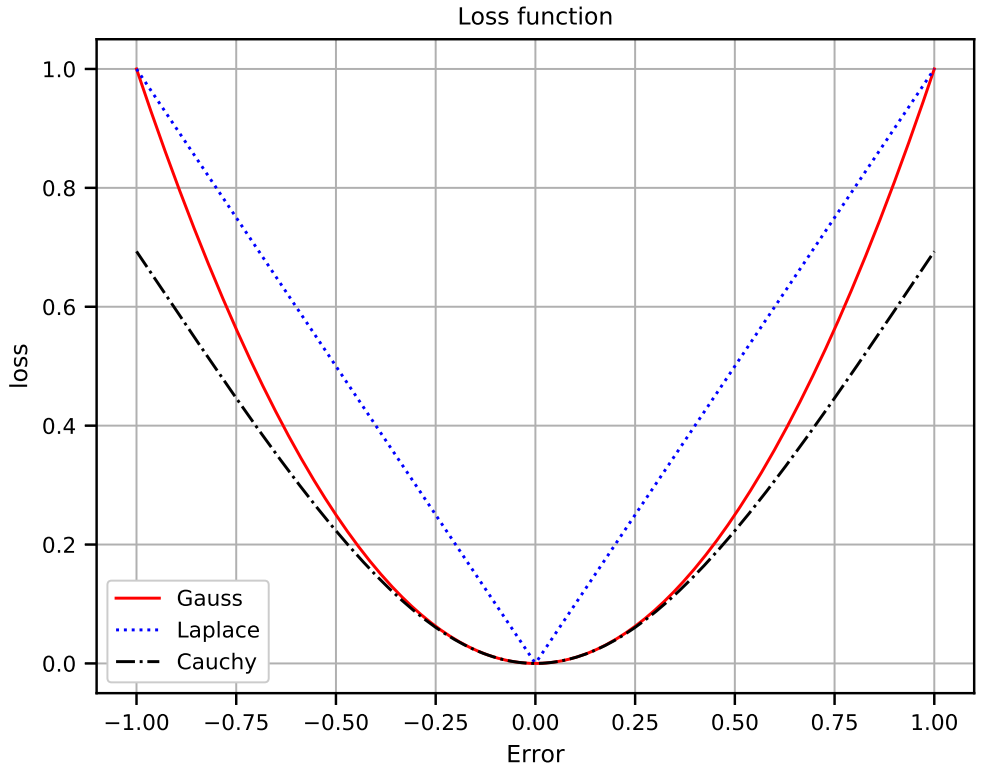

Figure 4.4: Illustration of the variations of three loss functions with respect to the error $e = \widehat{y} - y$. i) Gauss loss function (red solid line): $l = e^2$; ii) Laplace loss function (blue dot line): $l = |e|$; and iii) Cauchy loss function (black dot-dash line): $l = \log\,(1 + e^2)$.

where γ is the scale parameter. On making use of (4.19) in equation (4.8), the result is

$$\tilde{\theta} = \underset{\theta}{\text{argmax}} \sum_{k=1}^{r} -\log\,[1 + (\frac{(G(x^k;\theta) - y^k)}{\gamma})^2]. \tag{4.20}$$

Then, the associated loss function has the form:

$$\mathcal{L}(x, y, \theta) = \sum_{k=1}^{r} \log\,[1 + (\frac{\widehat{y}^k - y^k}{\gamma})^2]. \tag{4.21}$$

The differences amongst the three types of loss functions are illustrated in Figure 4.4. Clearly, with the Laplace loss function, the error decays at a constant rate,

regardless of the error magnitude, whereas the decay rate depends on the error value for the other two types of loss functions. Especially, the decaying rate can be shown to approach 0 when the error approaches 0, which is detrimental for certain optimization algorithms, like the Adam [91]. In practice, there is not much difference between the Gauss type and Cauchy type of loss function. In the current work, the Laplace loss function is applied to facilitate the neural network training.

4.4 Neural machine

In this section, the author briefly discusses the architecture of $G(\theta)$ and demonstrates how the model can be trained to map a history sequence $\mathbf{X}$ to a future sequence $\mathbf{Y}$, which are not necessarily of the same length.

4.4.1 Recurrent neural networks

Recurrent neural networks (RNNs) are one kind of neural networks designed to process sequential data whose entries are correlated in the time domain. The recurrent action is defined as [92]

$$\mathbf{h}_t = \mathcal{R}(\mathbf{h}_{t-1}, \mathbf{u}_t, \theta), \tag{4.22}$$

where $\mathbf{h}_t$ and $\mathbf{u}_t$ are the hidden state and input data at time step t, respectively. In the context of predicting the future from the past, the RNN is trained to use $\mathbf{h}_t$ as a lossy summary of the task-relevant aspects of the past input sequence $\{\mathbf{u}_i, i = 1, \ldots, t-1\}$. Regardless of the input sequence length, the RNN has the same input dimension and parameter θ from one step to another. This is the main advantage

of using RNN for processing sequential data since the parameters are shared across different time steps, thus, greatly reducing the model parameter size, as compared with a convolutional neural network [93].

One may notice the similarity between a RNN and a dynamical system, as given by equation (4.6a). The hidden state vector can be viewed as the state variable in a discrete dynamical system and the input time series can be considered to be similar to an external driving input in the dynamical system counterpart. However, there is a difference in that θ is described by analytical expressions like polynomials in (4.6a) whereas it is represented in terms of matrix weights and vector biases in (4.22).

Note that there is no theoretical restriction on the length of the input sequence $\mathbf{X}$ and RNN can be used to map an arbitrarily long sequence to a current hidden state vector with fixed dimension. Therefore, $\mathbf{h}_t$ is in general necessarily lossy, limited by the information capacity of its dimension. Therefore, conceptually, $\mathbf{h}_t$ may not be able to capture the long-term dependencies within $\mathbf{X}$. In fact, it has been shown that learning long-term dependencies with typical gradient descent method is difficult since the gradients propagated over many stages tend either to vanish or explode [94]. This poses an obstacle for forecasting the long-term behavior of a dynamical system from RNNs, especially, given that a chaotic system's behavior is highly sensitive to small perturbations. Fortunately, many approaches have been proposed to alleviate this problem through the introduction of special structures, like Long Short-Term Memory [77], Gated Recurrent Unit [95], skip mechanisms [96], highway connections [97], and so on. Next, the author elaborates on the techniques

that they have applied in $G(\theta)$.

4.4.2 Long short-term memory

For the one-step mapping function $\mathcal{R}$, a recurrent neural network is a natural choice. However, as mentioned above, typical RNN cells can suffer from the issue of vanishing gradients due to the recurrent multiplication of hidden state matrices when applying a gradient descent algorithm during the training stage. For general-purpose sequence modeling, the author has found that the Long short-term memory [77], which is purposely built to store long dependency information in a memory cell, is better for extraction and transfer of data in long sequences. The memory cell is accessed, written, and cleared by several self-parametrised controling gates. The author has followed earlier work [98] to define the action of a single LSTM cell by

$$\mathbf{i}_t = \sigma(W_{ui}\mathbf{u}_t + W_{hi}\mathbf{h}_{t-1} + W_{ci}\mathbf{c}_{t-1} + \mathbf{b}_i), \tag{4.23a}$$

$$\mathbf{f}_t = \sigma(W_{uf}\mathbf{u}_t + W_{hf}\mathbf{h}_{t-1} + W_{cf}\mathbf{c}_{t-1} + \mathbf{b}_f), \tag{4.23b}$$

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tanh(W_{uc}\mathbf{u}_t + W_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c), \tag{4.23c}$$

$$\mathbf{o}_t = \sigma(W_{uo}\mathbf{u}_t + W_{ho}\mathbf{h}_{t-1} + W_{co}\mathbf{c}_t + \mathbf{b}_o), \tag{4.23d}$$

$$\mathbf{h}_t = \mathbf{o}_t \circ \tanh(\mathbf{c}_t), \tag{4.23e}$$

wherein $\sigma(x) = 1/(1+e^{-x})$ is the logistic sigmoid function, $\circ$ denotes the Hadamard product, and $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{c}_t$, $\mathbf{o}_t$, and $\mathbf{h}_t$ are the *input gate*, *forget gate*, *cell memory*, *output gate*, and *cell hidden state* at time step t, respectively. The weighting matrix subscripts are defined so that W_{hi} is the hidden-input gate matrix, W_{cf} is the memory-forget gate matrix, and so on. By using the memory cell and controlling gates, the

gradient is to be trapped in the cell and prevented from vanishing.

4.4.3 Encoder-decoder neural machine

A basic form of the neural machine, an example of which is shown in Figure 4.5, consists of two components: a) an encoder that is used to summarize the input sequence $\mathbf{X}$ and compute the conceptualized "thought" vector $\mathbf{e}$ and b) a decoder that is used to start from this vector $\mathbf{e}$ and continuously decode one time step information at a time. Thus, the conditional probability is decomposed as

$$\log P_m(\mathbf{Y}|\mathbf{X}) = \sum_{j=1}^{n_y} \log P_m(y_j|y_{j-1}, \mathbf{e}). \tag{4.24}$$

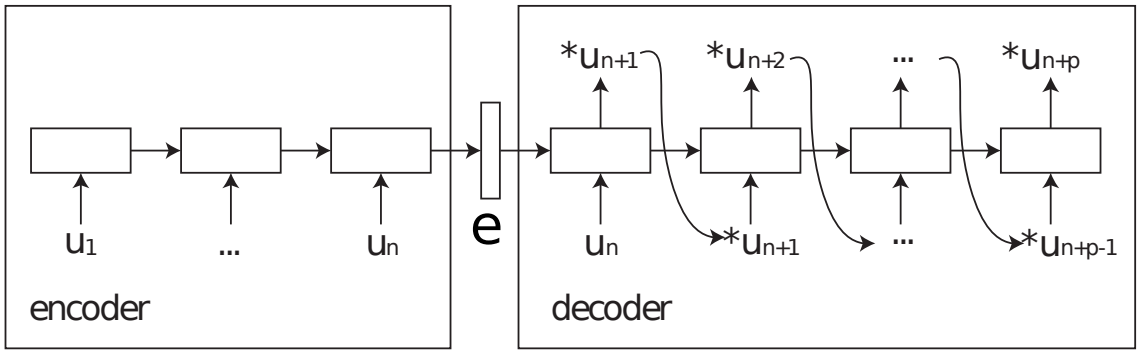

Figure 4.5: The input time series, with length n, is fed into the neural network through the encoder, at the left bottom starting from u_1 to u_n. The output time series, with length p, is generated from the decoder, at the right top from ${}^*\mathrm{u}_{n+1}$ to ${}^*\mathrm{u}_{n+p}$, which is the predicted time series. The corresponding ground truth data set u_{n+1} to u_{n+p} is not shown here. $\mathbf{e}$ is the conceptualized "thought" vector, which is used to aggregate the input series. The decoder is used to decode $\mathbf{e}$ once per time step and feed the results from previous time step output to the next time step as the input.

A potential issue with the use of the fixed-length vector $\mathbf{e}$ when processing long input sequence $\mathbf{X}$ is the bottleneck on the size of information containment. To improve the performance of this basic encoder-decoder architecture, different ways have been proposed to solve this problem by allowing a layer of neurons to automatically (soft-) search for parts of the input sequence $\mathbf{X}$ that are relevant in predicting the output sequence $\mathbf{Y}$. This is known as attention mechanism in Neural Machine Translation [64, 99]. However, in the author's model, the original attention mechanism has been modified in a way that is similar to what is proposed in the delay embedding theorem for a dynamical system. Here, this is called the *inhibitor mechanism*. The presence of an inhibitor will help the generation of the future time series from the chaotic system inference, without having to quickly loose predictability.

The proposed scheme is a general framework where one can freely define the one-step forward mapping function $\mathcal{R}$. Next, the author describes briefly the choices they have made for the encoder and decoder to learn aperiodic behavior of dynamical systems. In addition, the inhibitor mechanism is elaborated upon to demonstrate the viability for predicting a long sequence.

4.4.3.1 Encoder

Multiple LSTMs can be stacked and temporally concatenated to form deep neural structures to solve many practical sequence modelling problems [100, 101]. Many layers or deep neural network can be used to learn multiple levels of abstrac-

tion which can help classification or regression tasks [93]. However, gradient-based training becomes difficult with increasing depth of layers [102]. In the author's model, ideas similar to highway networks [103] have been applied to allow unimpeded information flow across several layers on the so-called *information highways*. With this construction, along the depth dimension, the author has introduced gating mechanisms as well to encourage gradient flow to help with the training.

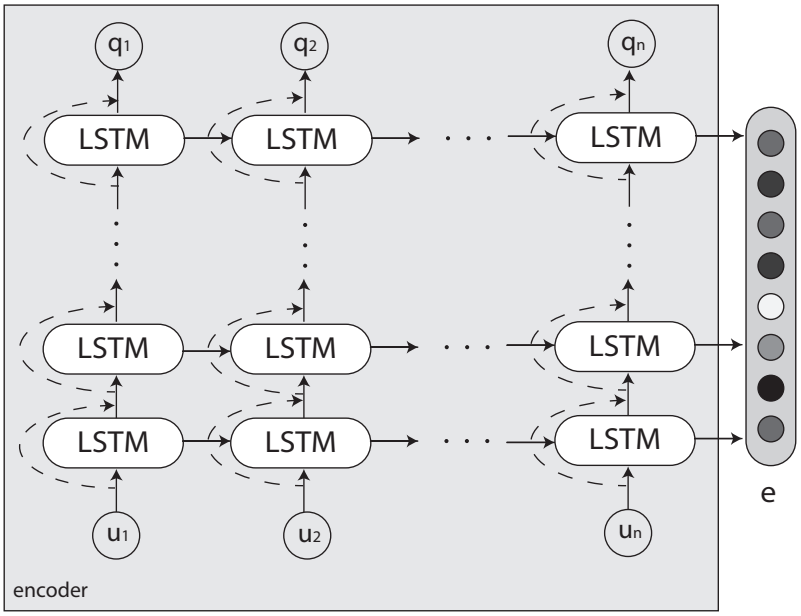

Figure 4.6: Unrolled version of the encoder. Multiple layers of LSTM cells are stacked in order to extract higher abstractions of the input u_i. The hidden state $\mathbf{h}_t$ is the concatenation of all hidden states of all LSTM cells. The record vector $\mathbf{q}_t$ is the output of the top-layer LSTM cell. The thought vector $\mathbf{e}$ is the final state of stacked LSTM cells. The dashed lines are the highway connections that allow the residual to be passed via a gating mechanism.

As shown in Figure 4.6, the encoder, which consists of multiple layers of LSTMs, is used to take the input sequence $\mathbf{X} = \{u_1, \ldots, u_t, \ldots, u_n\}, u_t \in \mathbb{R}^m$ and

produce the conceptualized vector $\mathbf{e}$. During the encoding phase, the hidden states are calculated as

$$\mathbf{h}_{t+1} = \mathcal{R}_E(\mathbf{h}_t; \mathrm{u}_{t+1}), \quad \text{for} \quad t = 1, \ldots, n-1. \tag{4.25a}$$

$$\mathbf{q}_t = \mathcal{Q}_E \mathbf{h}_t, \quad \text{for} \quad t = 1, \ldots, n. \tag{4.25b}$$

where $\mathcal{R}_E$ is the one step forward-mapping of the multi-layer LSTMs in the encoder and $\mathcal{Q}_E$ is the affine transformation to compute $\mathbf{q}_t$, called the *record vector*, which is shown in the Figure 4.6. The thought vector $\mathbf{e}$, which is used to compress the input sequence information, is the concatenation of last hidden states of the multi-layer LSTMs.

4.4.3.2 Decoder

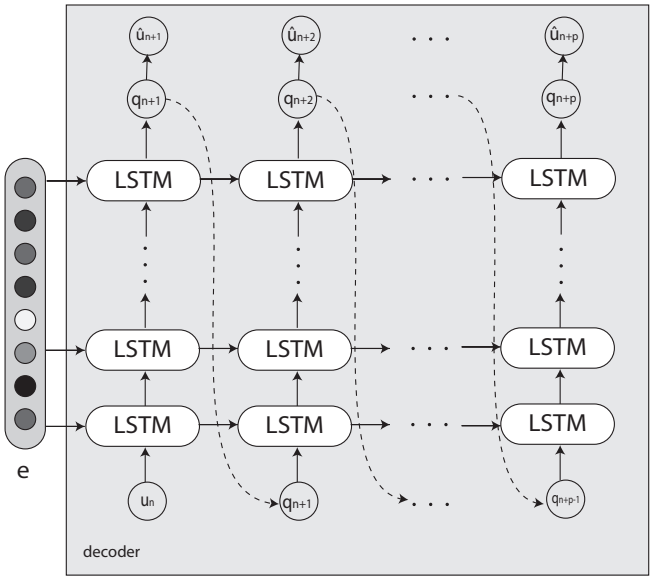

Figure 4.7: Unrolled decoder consists of multiple LSTM layers. The preceding time step output u_t is used as the input at the next time step. Again, highway connections have been added to help train the deep networks (which are not shown here). The output prediction $\{\widehat{\mathrm{u}}_t, n+1 \leq t \leq n+p\ \}$ is expected to be close to $\{\mathrm{u}_t, n+1 \leq t \leq n+p\ \}$ after the training.

With the decoder, $\mathbf{e}$ is used as the initial state vector and u_n, the last input to the encoder, is used as the first input, to continuously decode the thought vector $\mathbf{e}$ at each step. Suppose that the true output time sequence is

$$\mathbf{Y} = \{\mathrm{u}_{n+1}, \ldots, \mathrm{u}_t, \ldots, \mathrm{u}_{n+p}\}, \mathrm{u}_t \in \mathbb{R}^m,$$

and the prediction from the decoder is

$$\widehat{\mathbf{Y}} = \{\widehat{\mathrm{u}}_{n+1}, \ldots, \widehat{\mathrm{u}}_t, \ldots, \widehat{\mathrm{u}}_{n+p}\}, \widehat{\mathrm{u}}_t \in \mathbb{R}^m.$$

Then, the following equations hold

$$\mathbf{h}_{t+1} = \mathcal{R}_D(\mathbf{h}_t, \widehat{\mathrm{u}}_{t+1}), \quad \text{for} \quad t = n, ..., n+p-1. \tag{4.26a}$$

$$\mathbf{q}_t = \mathcal{Q}_D \mathbf{h}_t, \qquad \text{for} \quad t = n, ..., n+p. \tag{4.26b}$$

$$\widehat{\mathrm{u}}_t = \mathcal{U}_D \mathbf{q}_t, \qquad \text{for} \quad t = n, ..., n+p. \tag{4.26c}$$

in which $\mathbf{h}_t$ is hidden state at t time step ($n+1 \leq t \leq n+p$). $\mathbf{q}_t$ is the record vector. $\mathcal{R}_D$ is the action of decoder with multiple layers of LSTM. $\mathcal{Q}_D$ is similar to the definition in the encoder. The prediction at each time step $\widehat{\mathrm{u}}_t$ is obtained through the application of another affine transform $\mathcal{U}_D$ on $\mathbf{q}_t$. By combining equations (4.26a) to (4.26c) for a single time step, one can find the recurrent prediction equation:

$$\widehat{\mathrm{u}}_{t+1} = \mathcal{U}_D \mathcal{Q}_D \mathcal{R}_D(\mathbf{h}_t, \widehat{\mathrm{u}}_t). \tag{4.27}$$

It is recalled that the hidden state $\mathbf{h}_t$ in RNN is a lossy sum of input time series $\mathbf{X}$ and a part of the output time series $\mathbf{Y}$ up to the time step t, as shown in (4.22). Equation (4.27) can be readily written as

$$\widehat{\mathrm{u}}_{t+1} = G(\mathrm{u}_1, \ldots, \mathrm{u}_n, \widehat{\mathrm{u}}_{n+1}, \ldots, \widehat{\mathrm{u}}_t; \theta), \tag{4.28a}$$

$$= G(\mathbf{X}, \widehat{\mathbf{Y}}_{\leq t}; \theta), \tag{4.28b}$$

where $G(\bullet; \theta)$ is the action of the neural machine and subscript $\leq t$ means the entries up to time step t. At every time step, $\widehat{\mathrm{u}}_t$ needs to be targeted at true value u_t. This is the same as minimizing the loss function defined in the previous section. From a probability perspective, the author maximizes the conditional probability $P_m(\widehat{\mathrm{u}}_{t+1} | \widehat{\mathrm{u}}_t, \theta)$. Therefore, for each set of training data $(\mathbf{X}, \mathbf{Y})$, one needs to solve

the following problem:

$$\tilde{\theta} = \underset{\theta}{\operatorname{argmax}} \; P_m(\mathbf{Y}|\mathbf{X};\theta), \tag{4.29a}$$

$$= \underset{\theta}{\operatorname{argmax}} \prod_{j=n+1}^{n+p-1} P_m(\widehat{\mathrm{u}}_{j+1}|\widehat{\mathrm{u}}_{\leq j},\mathbf{X};\theta). \tag{4.29b}$$

Combining equation (4.29a) with equation (4.9), one can obtain the full-form estimator for the parameters θ in the neural machine:

$$\tilde{\theta} = \underset{\theta}{\operatorname{argmax}}\, \mathbb{E}_{\mathbf{X}\sim\tilde{P}_d}\left[\sum_{j=n+1}^{n+p-1} \log\, P_m(\widehat{\mathrm{u}}_{j+1}|\widehat{\mathrm{u}}_{\leq j},\mathbf{X};\theta)\right]. \tag{4.30}$$

4.4.4 Inhibitor

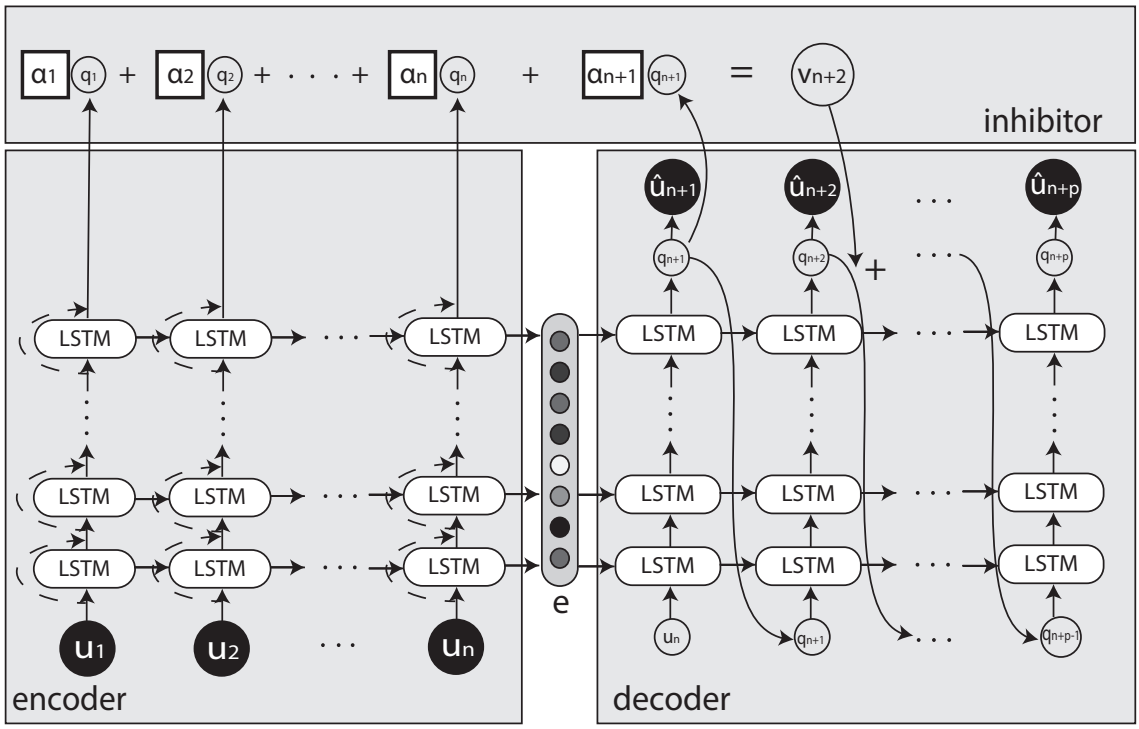

Figure 4.8: The inhibitor is the weighted average of the record vectors $\mathbf{q}_t$ with the self-learned weights α_t. As a result, the decoder has direct access to all previous step information for making the next step inference. Although within the neural machine, one implicitly reads all previous steps to make the next step prediction, the author makes this connection explicit, also facilitating the back-propagation of error during training as well. The inhibitor v_{n+2} will be used in (4.31) to predict the future.

As mentioned earlier, an inhibitor mechanism is introduced at each decoding step, for augmenting the decoder input with the history information; that is, the collection of record vectors $\mathbf{q}_t$ generated both by the encoder and decoder, as shown in Figure 4.8. Without this mechanism, the author finds that just a combination

of only the encoder and decoder cannot provide a long-term prediction due to the exponential growth of error. Now the output transformation function (4.26c) takes the form

$$\widehat{\mathrm{u}}_t = \mathcal{U}_D(\mathbf{q}_t + \mathbf{v}_t), \tag{4.31}$$

wherein $\mathbf{v}_t$ is called the *inhibitor vector*, which is computed at each decoding step as follows:

$$\beta_i = \psi(\mathbf{h}_t) + \phi(\widehat{\mathrm{u}}_t) \in \mathbb{R}, \quad \text{for } i = 1, \ldots, t-1. \tag{4.32a}$$

$$\alpha_i = \frac{e^{T\beta_i}}{\sum_{j=1}^{t-1} e^{T\beta_j}}, \tag{4.32b}$$

$$\mathbf{v}_t = \sum_{i=1}^{t-1} \alpha_i \mathbf{q}_i. \tag{4.32c}$$

First, in equation (4.32a), the functions ψ and ϕ are used to compute a score β_i for each previous time step from the current hidden state $\mathbf{h}_t$ and last time step output $\widehat{\mathrm{u}}_t$. Subsequently, in equation (4.32b), the score β_i is normalized through the softmax function to get the weight α_i. T is the self-learned parameter helping to differentiate the relative importance of each time step from the history in determining the future. It is similar to the definition of Boltzmann constant in statistical mechanics relating to the average kinetic energy of particles in a gas [104]. The denominator in (4.32b) serves as a normalization factor. Finally, the author computes the inhibitor vector $\mathbf{v}_t$ as the weighted average of the collection of $\mathbf{q}_t$ up to time step $t-1$. Moreover, several parallel inhibitor mechanisms can be adopted, resulting in additional performance boosting for large-scale systems. Note that the inhibitor mechanism grows as the decoding step approaches the end of the output time series. Therefore, the entire history of time series is considered for predicting the behavior at the last time step;

this is different from what is done with a traditional attention mechanism as stated in [64]. Another distinction comes from the way in which the scores are computed in equation (4.32a). Instead of comparing the distances of the hidden state vectors as shown in [99], the author proposes a mini full-layer network to capture the scores automatically.

4.5 Results and discussion

As mentioned earlier, three different chaotic systems are considered here, with two of them governed by ordinary differential equations and another by a partial differential system. The numerical experiments conducted with each of these systems are presented next [2].

4.5.1 Lorenz'63 system

First, the author examines the prediction ability of the neural machine with a low-dimensional chaotic system. It is mentioned that the neural machine can predict any time series simulated from the same Lorenz'63 system after the training, regardless of the initial conditions. This is different from the previous work of Pathak *et al.* [74], wherein a continuous data feed from the same initial condition is required for predicting future responses. Therefore, their network trained from one initial condition cannot be applied to predict another time series from a different initial condition. On the contrary, with the current work, the author has developed

[2]The details of hyperparameters for the training of the neural machine are listed in Appendix A. Additional results are presented in Appendix B.

a neural machine that can be used to predict future time series regardless of the initial condition.

The results obtained for three different cases are shown in Figures 4.9, 4.10, and 4.11. In these three cases, the systems are started from different initial conditions. However, with the constructed neural machine, the author is able to forecast the response for 7 Lyapunov times. In other words, the neural machine has the ability to forecast long-term responses of a chaotic dynamical system by only taking in short-term histories.

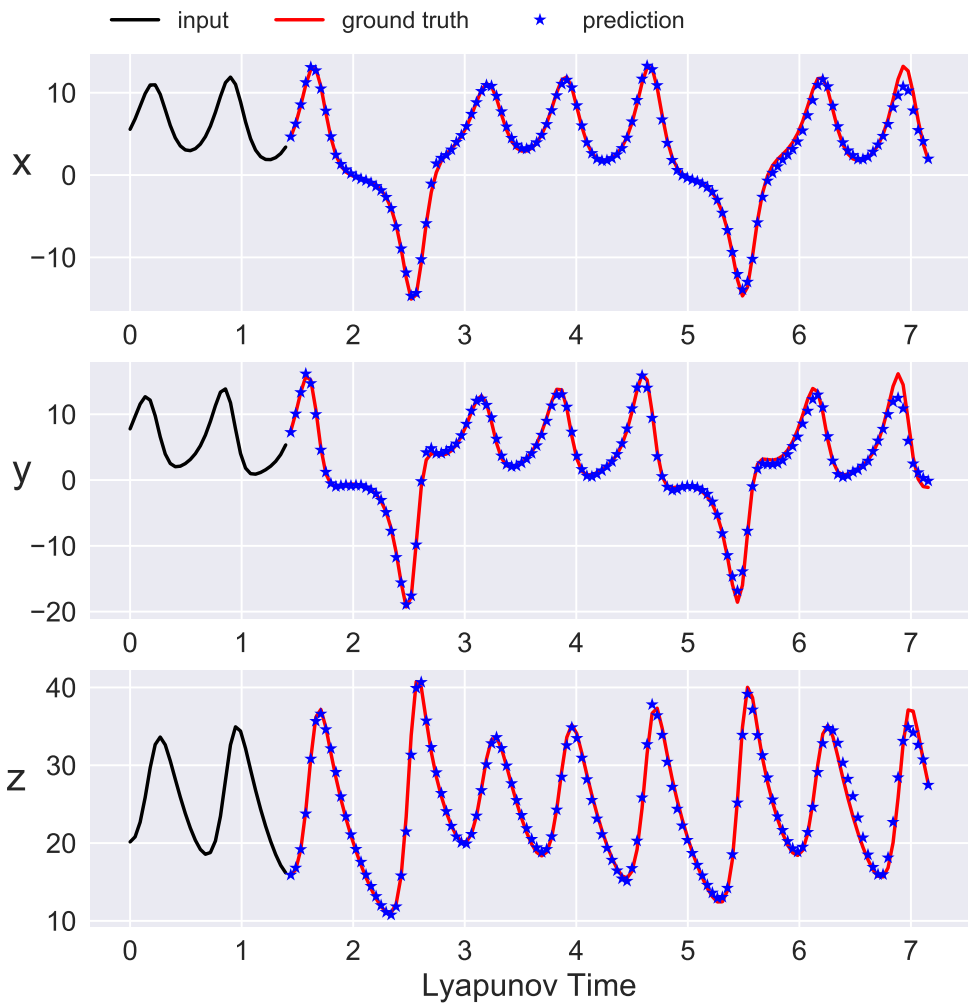

Figure 4.9: Lorenz'63 prediction (No.1).The black curves are history data segments. The blue dots are predictions from the neural machine. The red curves are the ground truth future datasets which are overlaid with the forecasting results for the sake of comparison.

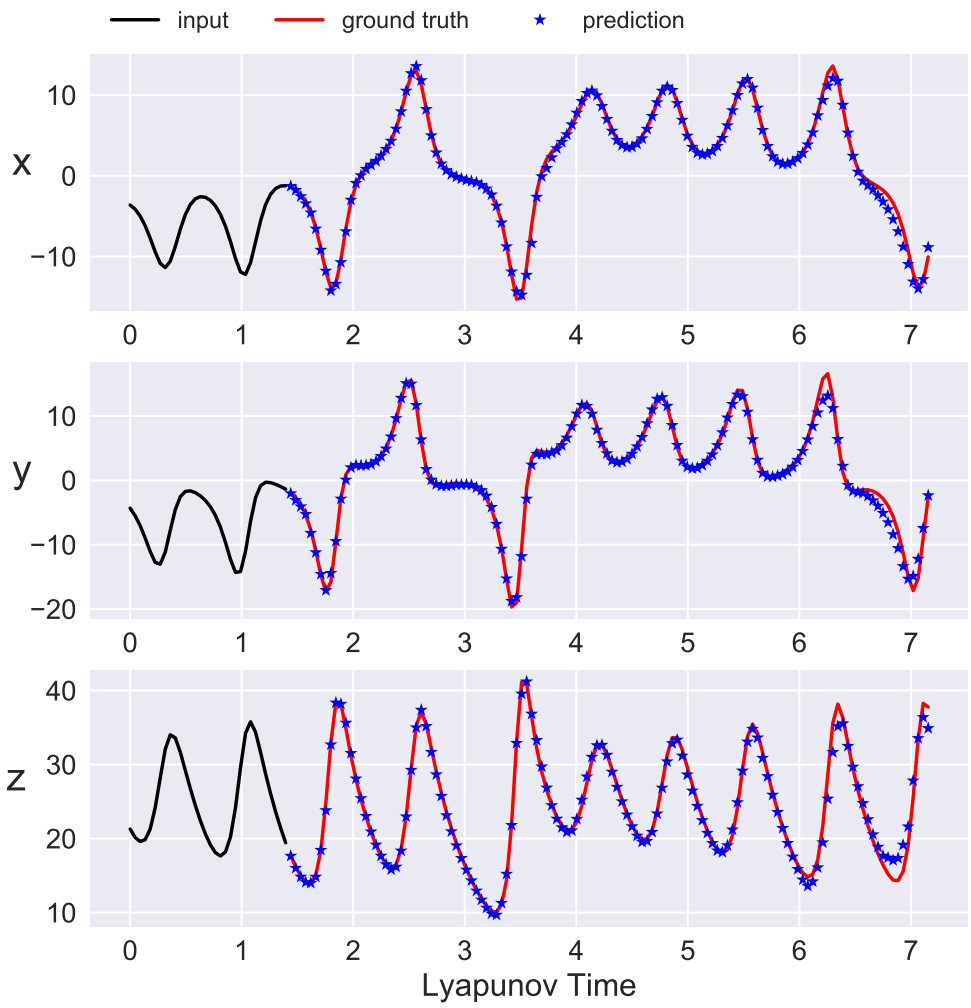

Figure 4.10: Lorenz'63 prediction (No.2). This is a second result obtained by using different historical data set but with the same neural network setting.

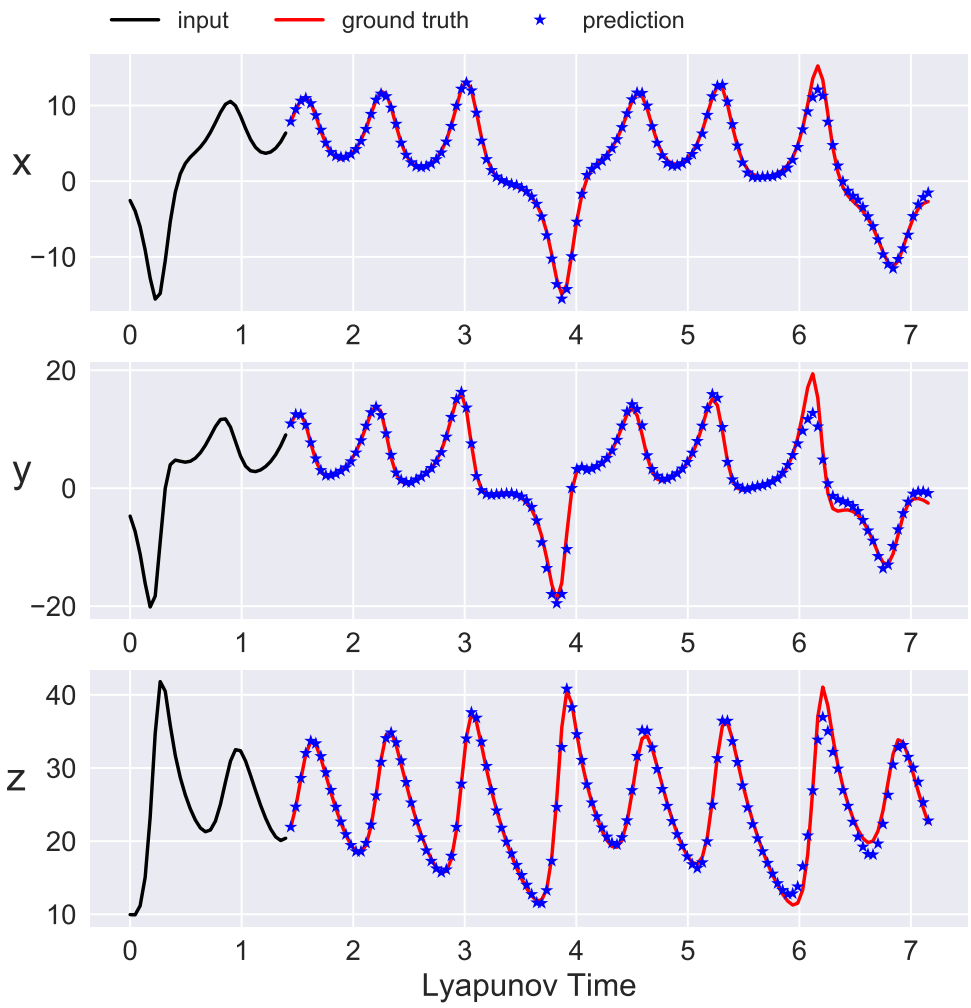

Figure 4.11: Lorenz'63 prediction (No.3). Again, this is a third result coming from a different history.

4.5.2 Lorenz'96 system

Next the author applies the constructed neural machine for forecasting the behaviour of the Lorenz'96 system. The results are shown in Figures 4.12, 4.13, and 4.14. Through the results presented in these figures, it has been shown that the

neural machine can also be used to capture the long-term behavior of a forty-eight dimensional chaotic system, a relatively high-dimensional chaotic system compared to the three dimensional system of the previous section.

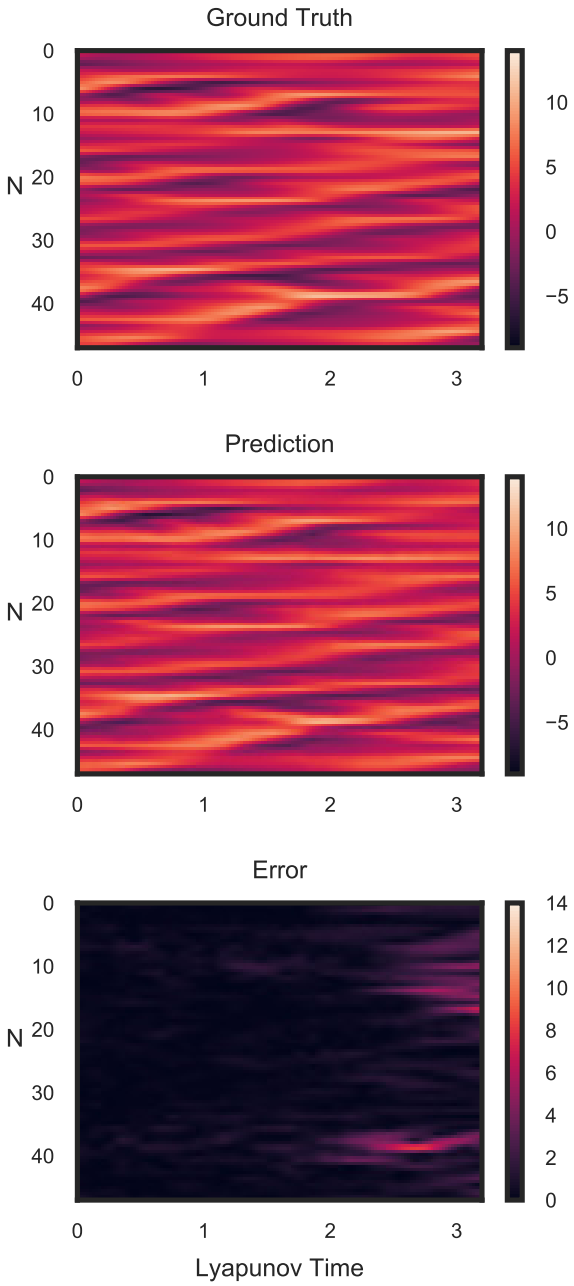

Figure 4.12: Lorenz'96 prediction (No.1). *upper*: ground truth simulation results obtained by solving (4.12) for 3.2 Lyapunov times with N = 48 and F = 8; *middle*: prediction results from the neural machine for the same initial condition; *lower*: absolute error between the ground truth and the prediction.

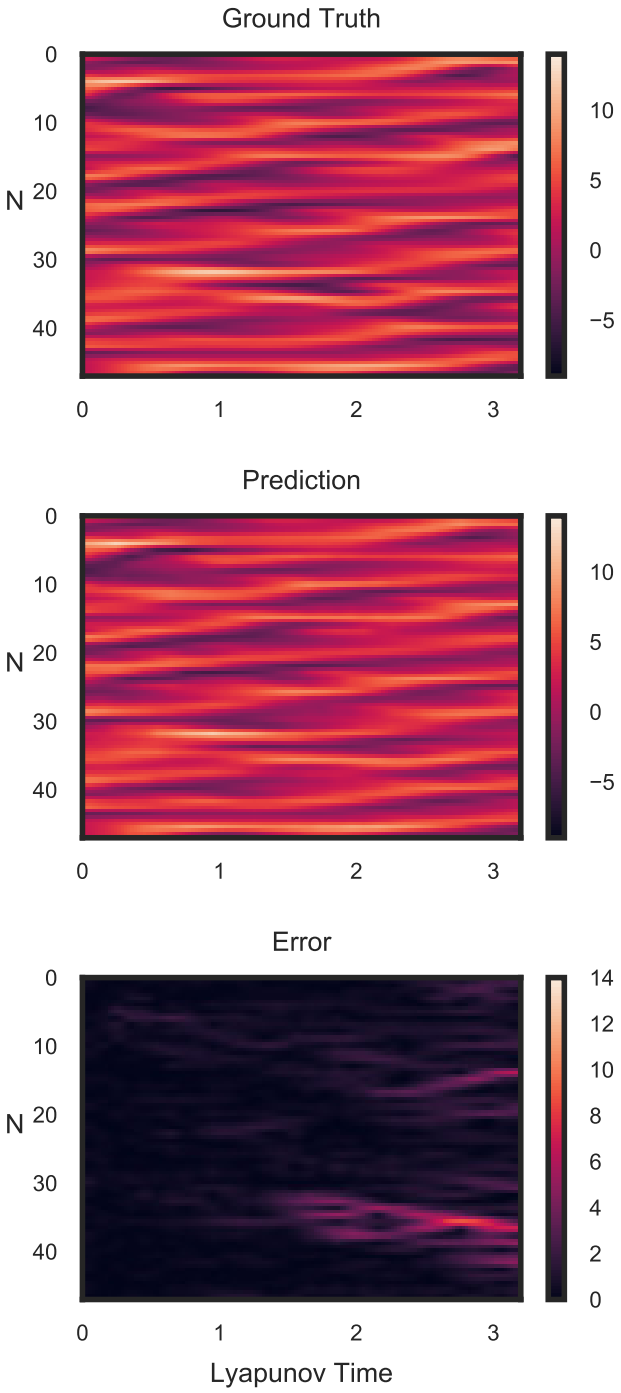

Figure 4.13: Lorenz'96 prediction (No.2). The result is obtained from the neural machine by digesting a different history data set.

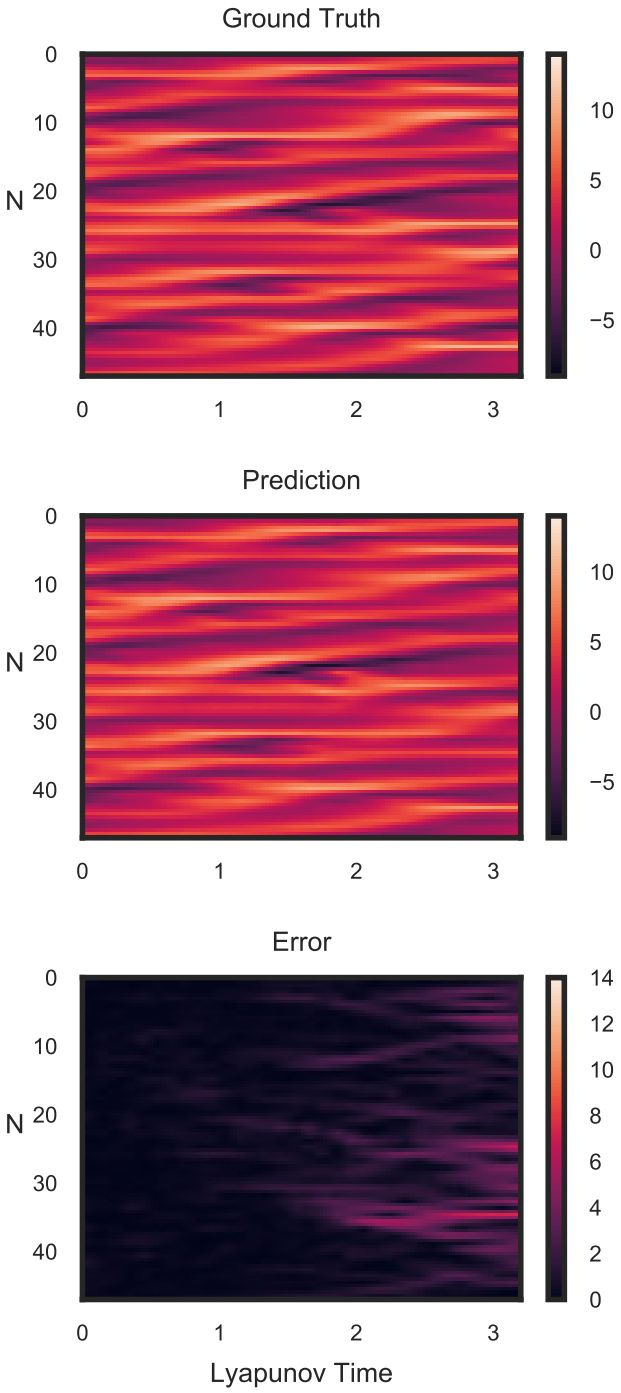

Figure 4.14: Lorenz'96 prediction (No.3).

4.5.3 Kuramoto-Sivashinsky system

In this section, the author would like to apply the same data-driven method for predicting the behaviour of spatio-temporally chaotic systems, by using the Kuramoto-Sivashinsky equation as an example.

The results are shown in Figures 4.15, 4.16, and 4.17. Again, the constructed neural machine has the ability to replicate the long-term evolution defined by the partial differential system without a change in the neural machine configuration used for the ordinary differential systems.

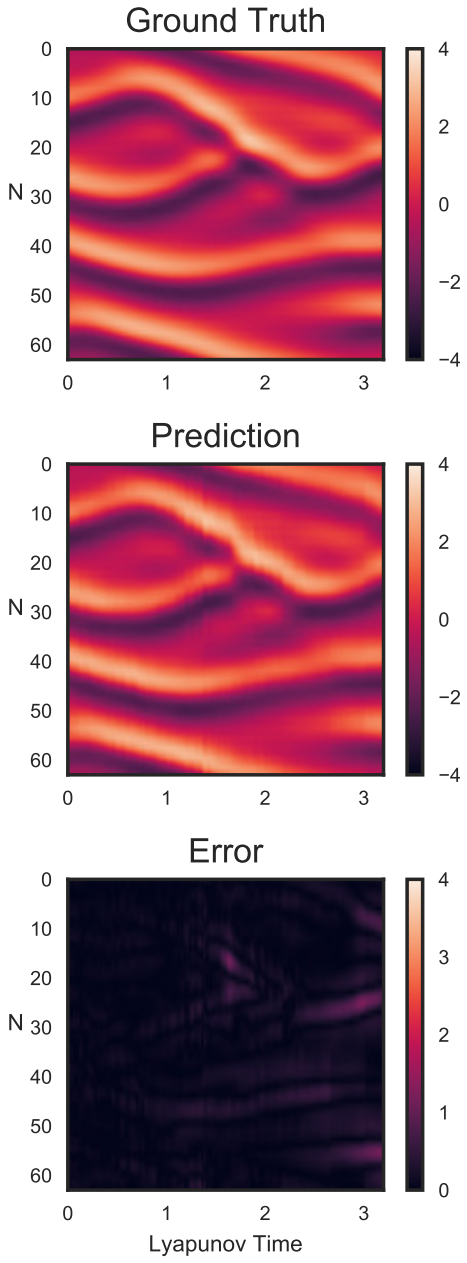

Figure 4.15: KS prediction (No.1). *upper*: true scalar field up to 3.2 Lyapunov times; *middle*: predicted scalar field; *lower*: absolute error as the difference between the true field and the predicted field.

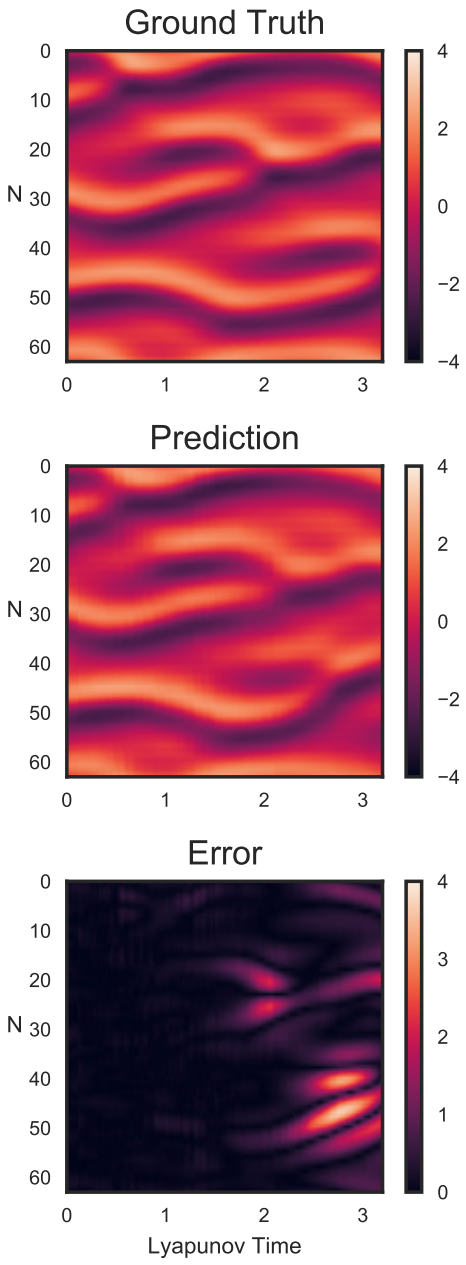

Figure 4.16: KS prediction (No.2) with a different history data set.

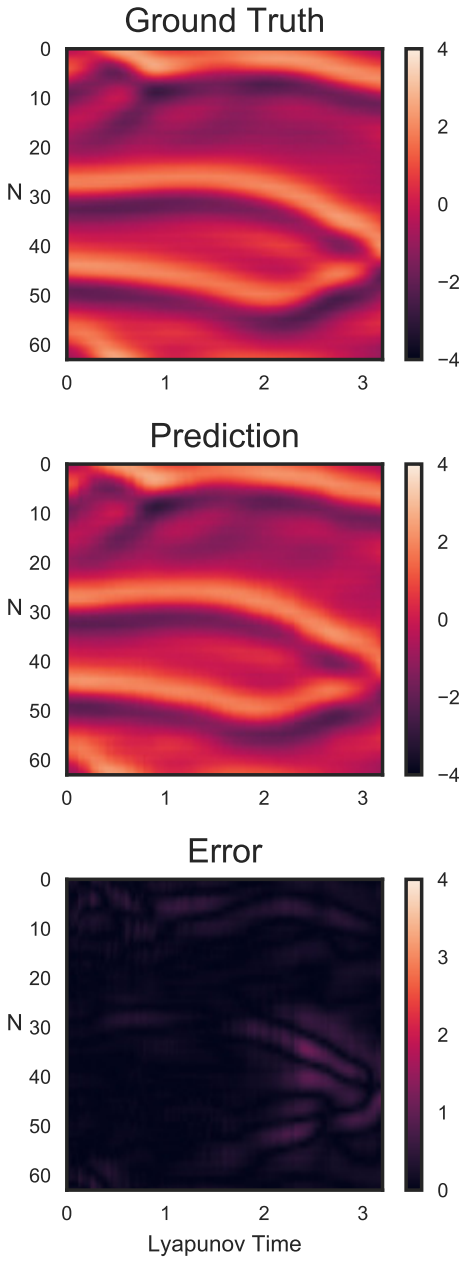

Figure 4.17: KS prediction (No.3).

Chapter 5: Neural Machine Based Forecasting of Non-autonomous System Dynamics

There are an unfathomable number of systems in nature that depend on time. Since the discovery of the expanding cosmos [105] and Big Bang [106], it is known that the universe itself is a time-dependent system. Probably the most notable time-dependent systems are the living creatures. The time effects can be observed through, for example, circadian rhythms [107], where the fluctuations of physical process are synchronized with the diurnal cycle. On a larger time-scale, every living creature undergoes a time-dependent process, called *aging*.

Apart from living systems, the time dependence of dynamics is observed ubiquitously in nature. Highly transient events, such as rogue waves [70], stock market crash, tornadoes, and so on are often shown up, resulting in significant impacts. Network theory about complex systems considers each individual element as being time dependent to study how the local interactions can lead to large-scale synchronizations.

Despite the prevalence of time-dependent dynamics in nature, there has been comparatively little research done on the prediction and analysis of time series from such systems. Mathematically, systems with explicit time dependent terms are

categorized as non-autonomous systems. Next, the author will give the definition of a non-autonomous system.

5.1 Background

5.1.1 Autonomous system

An autonomous system is a set of ordinary differential equations (ODEs) with the form

$$\frac{d}{dt}x(t) = f(x(t)), \tag{5.1}$$

where $x(t) \in \mathbb{R}^n$ is commonly regarded as the state variable and f describes the relation between differentials and the states. Systems that follow (5.1) are considered to be time-invariant systems, which means that these systems are invariant to shifts in time, either in the future or in the past. Suppose that $x = x_1(t)$ is the solution to the initial value problem:

$$\frac{dx}{dt} = f(x), \quad x(t = 0) = x_0. \tag{5.2}$$

Then $x_2(t) = x_1(t - t_0)$ is also a solution to

$$\frac{dx}{dt} = f(x), \quad x(t = t_0) = x_0. \tag{5.3}$$

This can be easily shown by the change of variables in time. Therefore, the above property is called *time-invariant*. The lowest dimension of nonlinear autonomous dynamical system which can exhibit chaotic behaviors is **three**; for example, the Lorenz system [20] and the Rössler system [108].

5.1.2 Non-autonomous system

A non-autonomous system has the form

$$\frac{d}{dt}x(t) = g(x(t), t). \tag{5.4}$$

Here, the governing law not only depends on the state itself, but also on an independent variable t, which is time here. Therefore, the dynamical system described by (5.4) is said to have explicit time-dependent terms.

5.1.3 Duffing system

The Duffing system can be described as second-order ODE with a cubic nonlinearity, written as

$$\frac{d^2x}{dt^2} + \delta\frac{dx}{dt} + \beta x + \alpha x^3 = 0, \tag{5.5}$$

where α is the cubic stiffness, β is the linear stiffness and δ is the damping factor. It can also be expressed in a state-space form as

$$\begin{cases} \frac{dx}{dt} &= y, \\ \frac{dy}{dt} &= -\delta y - \beta x - \alpha x^3, \end{cases} \tag{5.6}$$

where x can be regarded as position and y as velocity of the oscillator.

The above system was first investigated by Georg Duffing in 1918 to study a practical oscillation problem [109]. Readers who are interested in a detailed review on different applications and research efforts on the Duffing system are referred to reference [110].

When there is no damping($\delta = 0$), the integration of the above equation (5.5) results in

$$H(t) \equiv \frac{1}{2}\dot{x}^2 + \frac{1}{2}\beta x^2 + \frac{1}{4}\alpha x^4 = const. \tag{5.7}$$

Therefore, the undamped, unforced Duffing oscillator is a Hamiltonian system which has periodic dynamic characteristics. The form of the potential valley $V = \frac{1}{2}\beta x^2 + \frac{1}{4}\alpha x^4$ depends on the sign of β, if one assumes $\alpha > 0$ as usual. See the figures below.

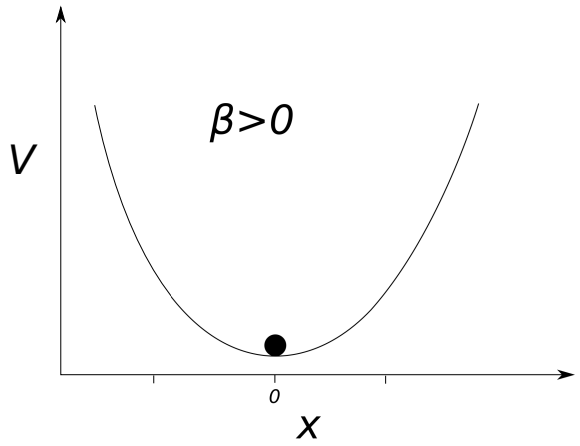

Figure 5.1: $\beta > 0$ with single potential valley for unforced Duffing oscillator. Fixed point is located at $x = 0$.

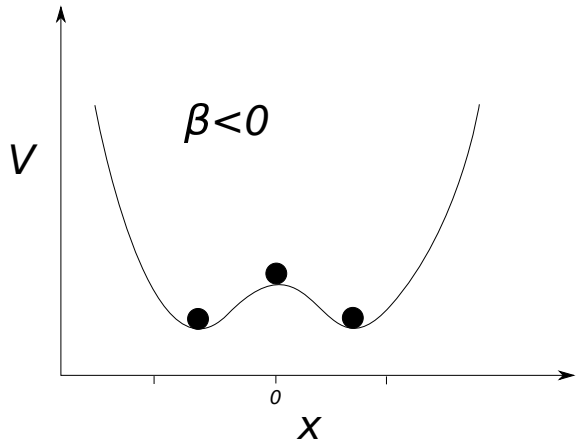

Figure 5.2: $\beta < 0$ with double potential valleys for unforced Duffing oscillator. Fixed point are located at $x = 0, \pm\sqrt{-\beta/\alpha}$. As explained before, the center fixed point is unstable and the other two are stable.

For the case $\alpha > 0, \beta > 0$, $H(t)$ is a Lyapunov function, and $x^* = 0$ is globally asymptotically stable, in the presence of damping, as shown in Figure 5.1. On the other hand, for $\alpha > 0, \beta < 0$ and $\delta > 0$, there are three equilibria as shown in Figure 5.2, one in the peak and the other two in the valleys. In this scenario, trajectories starting from all initial conditions converge to one of two stable valleys, except the one starting from the peak of the hill. The three equilibria can be found by setting $\frac{d^2x}{dt^2} = \frac{dx}{dt} = 0$, resulting in

$$x(\beta + \alpha x^2) = 0. \tag{5.8}$$

Hence, when $\alpha\beta < 0$ there are three fixed points: $x^* = 0, \pm\sqrt{-\beta/\alpha}$. Apart from visualization of the stability in the potential function plot, one can get the stability information by analyzing the eigenvalues of the Jacobian matrix of equation (5.6).

Let one rewrite equation (5.6) as

$$\frac{d}{dt}\begin{pmatrix} x \\ \\ y \end{pmatrix} = \begin{pmatrix} y \\ \\ -\delta y - \beta x - \alpha x^3 \end{pmatrix} \tag{5.9}$$

and the Jacobian of the right-hand side of the above equation reads as

$$J = \begin{pmatrix} 0 & 1 \\ \\ -\beta - 3\alpha x^2 & -\delta \end{pmatrix}. \tag{5.10}$$

Therefore, for the equilibrium $x^* = 0$, the eigenvalue is

$$\lambda = \frac{-\delta \pm \sqrt{\delta^2 - 4\beta}}{2} \tag{5.11}$$

and it is stable when $\beta > 0$ and unstable when $\beta < 0$. For the other two equilibria $x^* = \pm\sqrt{-\beta/\alpha}$, the eigenvalues read as

$$\lambda = \frac{-\delta \pm \sqrt{\delta^2 + 8\beta}}{2}. \tag{5.12}$$

When $\beta < 0$, these two are stable and non-existent when $\beta > 0$.

The forced Duffing system, which has more complex dynamical behavior compared with the unforced version, reads as

$$\frac{d^2x}{dt^2} + \delta\frac{dx}{dt} + \beta x + \alpha x^3 = \gamma\cos(\omega t), \tag{5.13}$$

where ω is the angular frequency and γ is the forcing magnitude. In state-space form, the system reads as

$$\begin{cases} \frac{dx}{dt} &= y, \\ \\ \frac{dy}{dt} &= \gamma\cos(\phi) - \delta y - \beta x - \alpha x^3, \\ \\ \frac{d\phi}{dt} &= \omega. \end{cases} \tag{5.14}$$

This three-dimensional autonomous dynamical system can be proven to be chaotic under certain parameter combinations [110].

In experimental realizations, the forced Duffing oscillator is usually represented by a periodically driven steel beam that vibrates between two magnets as $\beta < 0$ [110–112]. On the other hand when $\beta > 0$, it models a forced spring with restoring force $F = -\beta x - \alpha x^3$.

5.2 Softening Duffing oscillator

When α and β have opposite signs, one has a softening Duffing oscillator. The path to chaos through period-doubling bifurcation [12] can be shown by systematically changing certain parameters in (5.13). Let the parameters be fixed so that $\alpha = 0.2, \beta = -0.5, \delta = 0.085, \omega = 0.42.$ and let γ be varied. Starting from the initial condition $x(t = 0) = 1$ and $\frac{dx}{dt}|_{t=0} = 0$, the rich collection of trajectories governed by (5.13) is shown in the following series of figures.

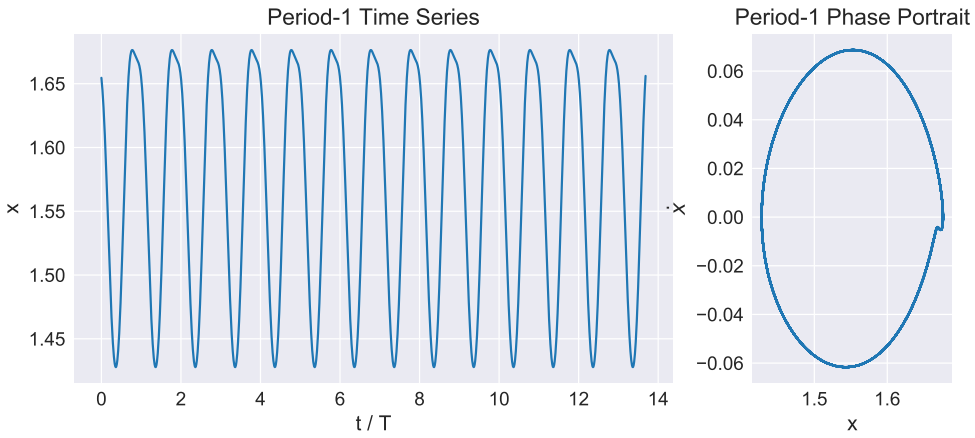

Figure 5.3: Period-1 dynamics with $\gamma = 0.1$. The response period is the same as the forcing period.

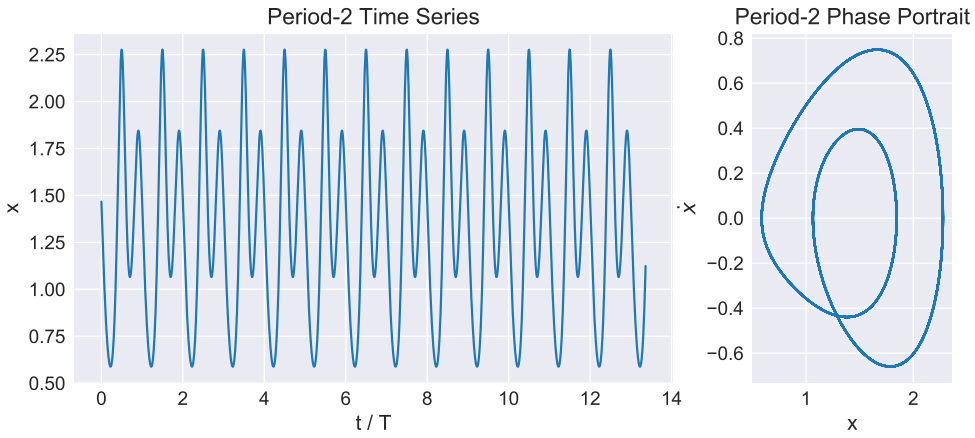

Figure 5.4: Period-2 dynamics with $\gamma = 0.2$. The response period is twice the forcing period.

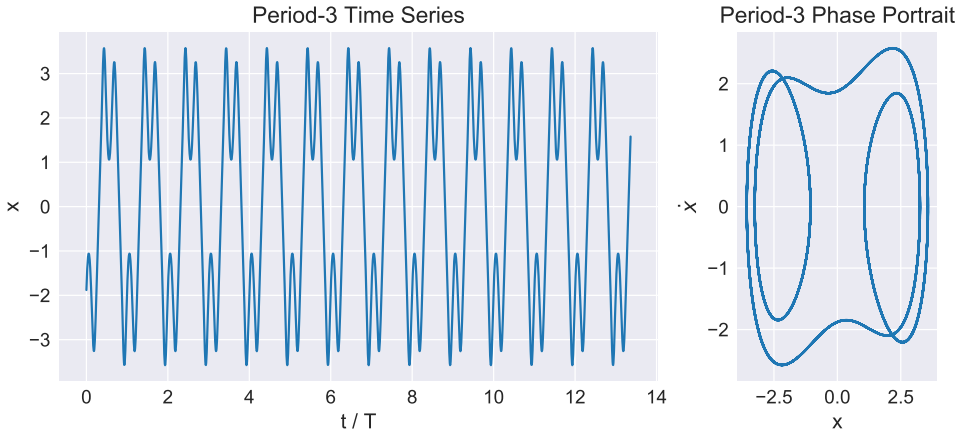

Figure 5.5: Period-3 dynamics with $\gamma = 2.0$.

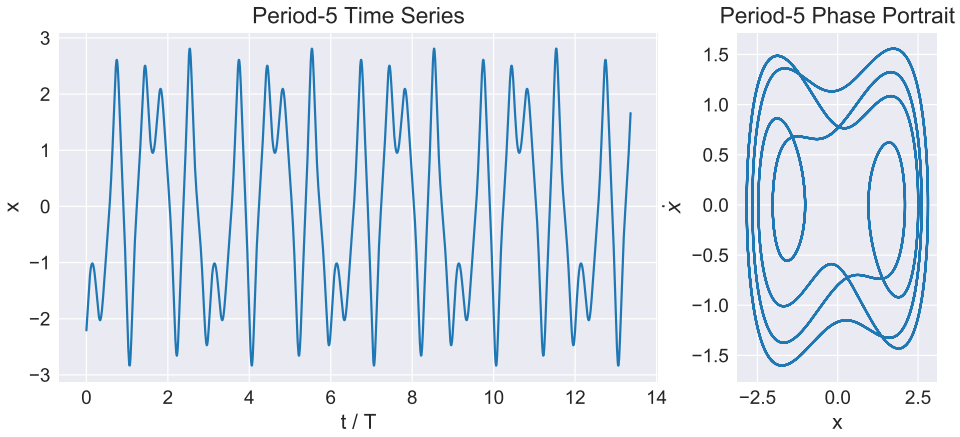

Figure 5.6: Period-5 dynamics with $\gamma = 0.33$.

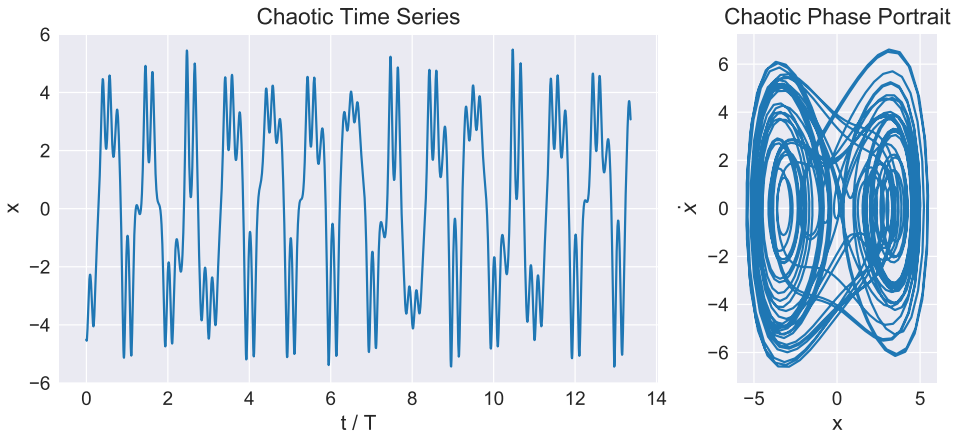

Figure 5.7: Chaotic dynamics with $\gamma = 7.0$.

Here, the author focuses on the prediction of chaotic time series given its complexity and potential challenges, such as that shown in Figure 5.7. In the following context, the author chooses $\gamma = 0.5$ and 1.7, and fixes the rest parameters as specified.

5.2.1 Prediction results: Forcing amplitude $\gamma = 0.5$

The maximal Lyapunov exponent is 0.0479, which is used to non-dimensionalize the time axis in the following figures. The input time series are shown in black lines and the output ground truth data is shown in red. The blue dots are the prediction values from the neural machine discussed in the earlier chapter.

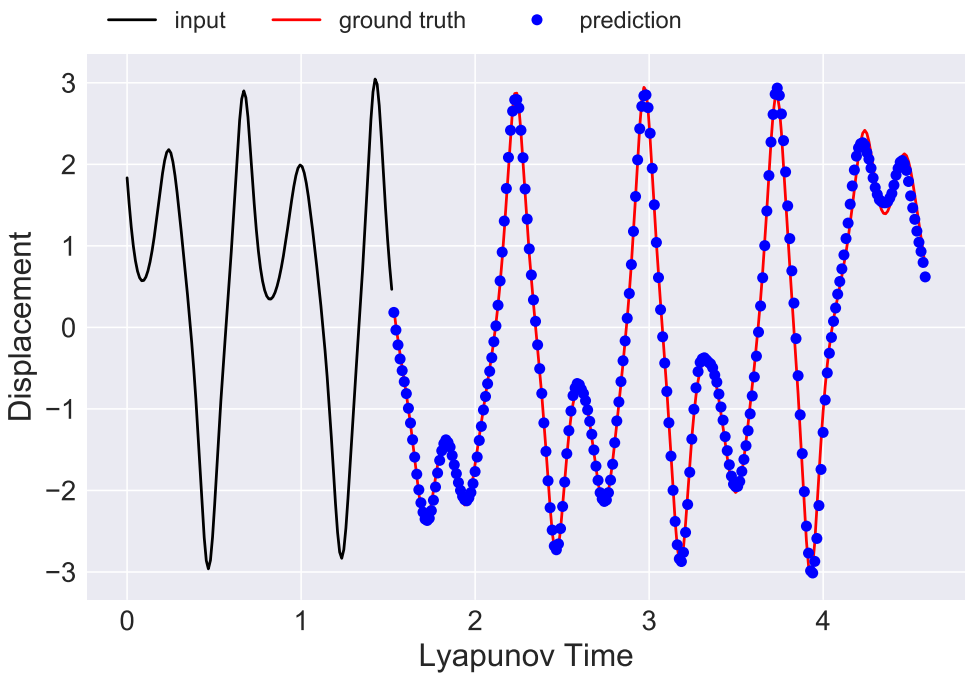

Figure 5.8: Softening forced Duffing oscillator with $\gamma = 0.5$ prediction (No.1).

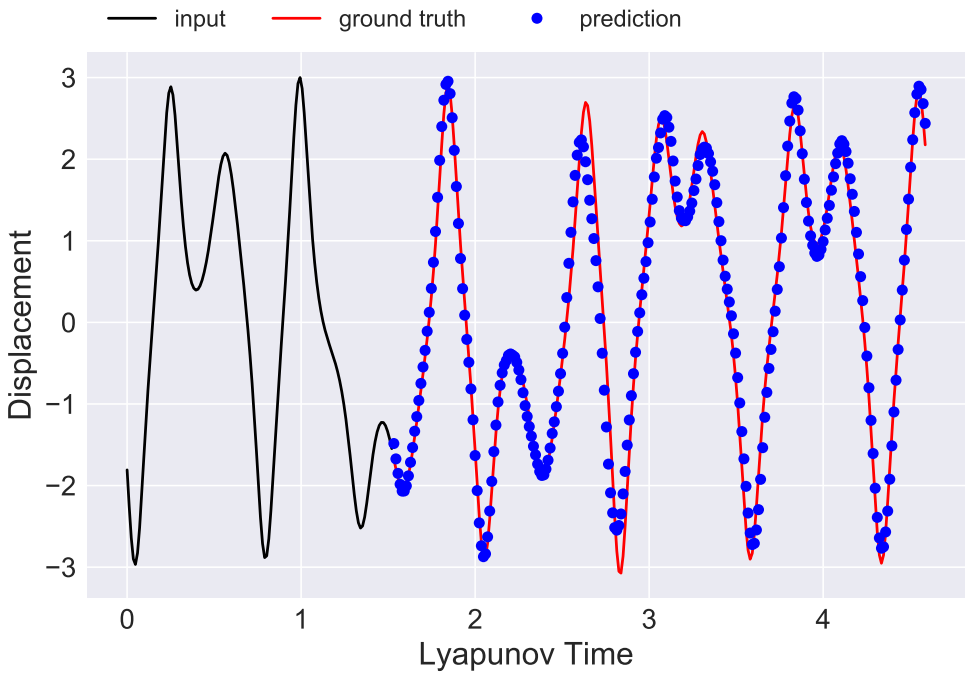

Figure 5.9: Softening forced Duffing oscillator with $\gamma = 0.5$ prediction (No.2).

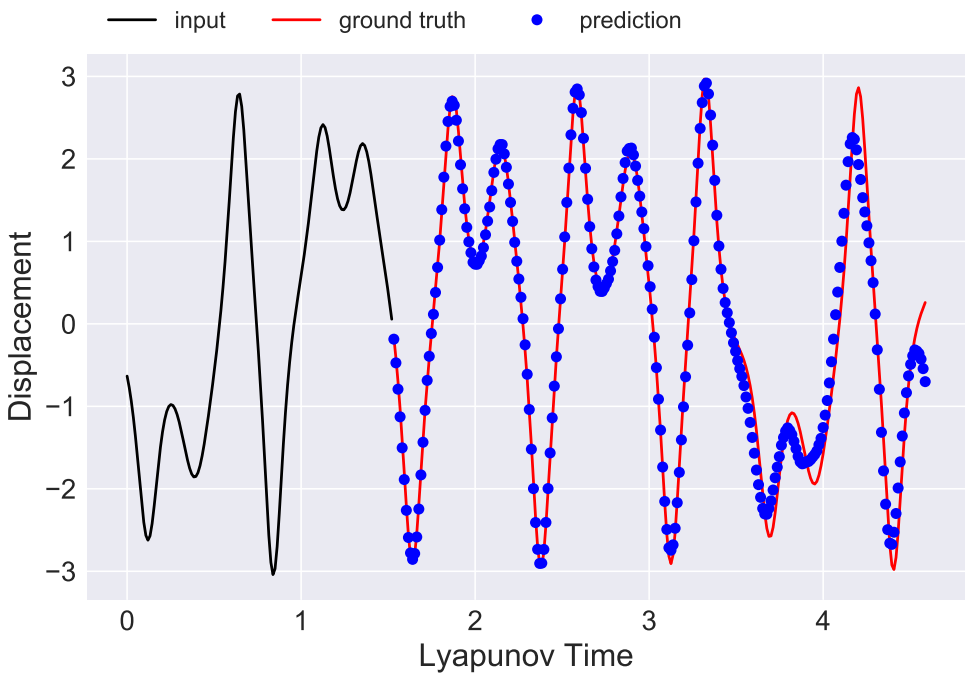

Figure 5.10: Softening forced Duffing oscillator with $\gamma = 0.5$ prediction (No.3).

5.2.2 Prediction results: Forcing amplitude $\gamma = 1.7$

In this case, the maximal Lyapunov exponent has been calculated as 0.076. Again for the sake of comparison, the time axis in the following figures has been non-dimensionalized.

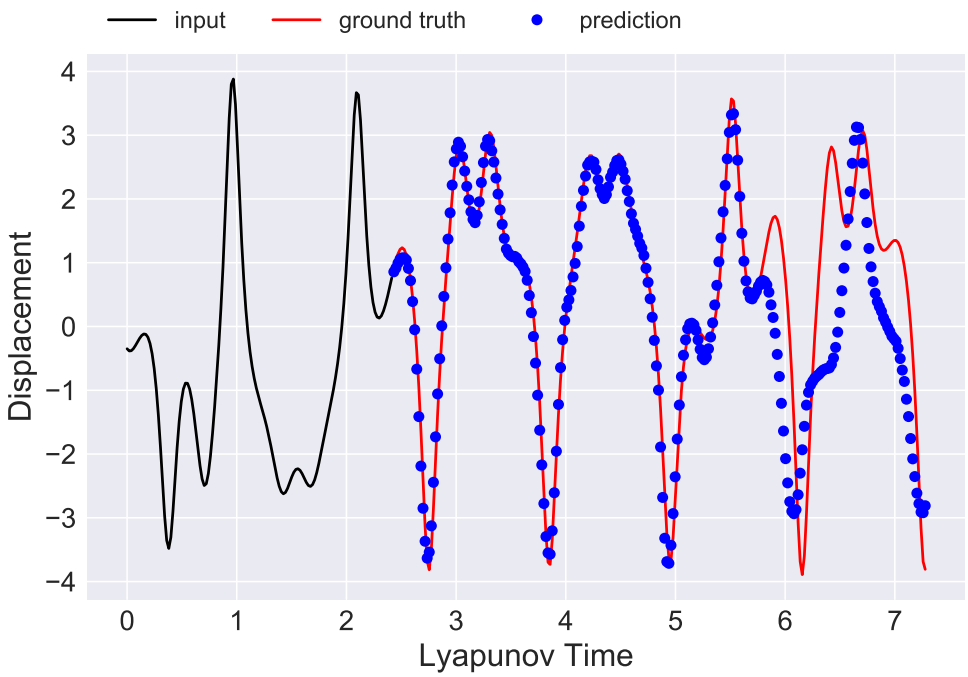

Figure 5.11: Softening forced Duffing oscillator with $\gamma = 1.7$ prediction (No.1).

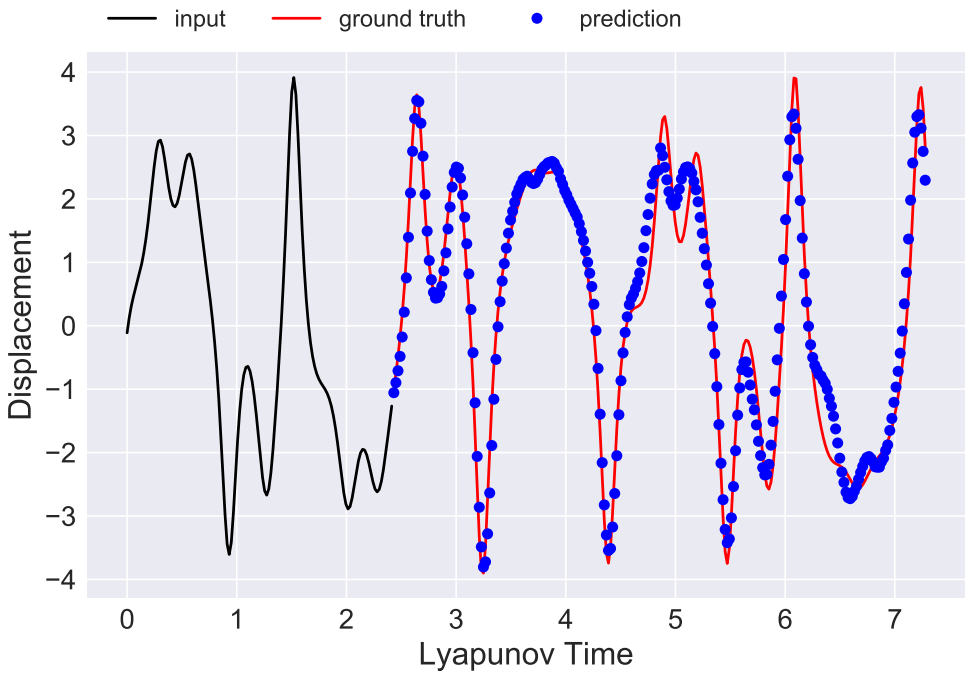

Figure 5.12: Softening forced Duffing oscillator with $\gamma = 1.7$ prediction (No.2).

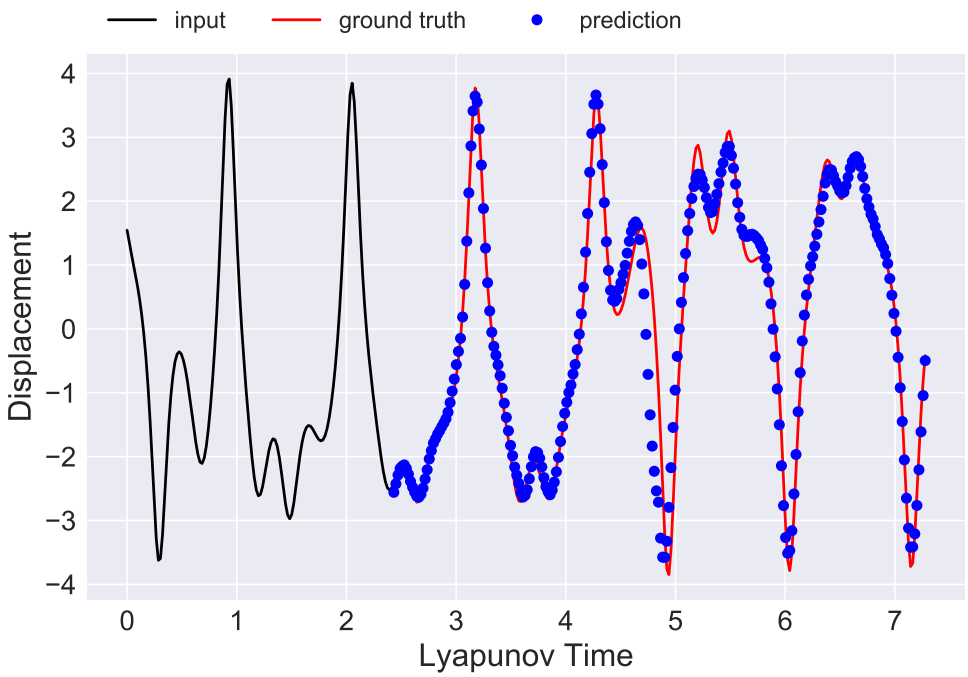

Figure 5.13: Softening forced Duffing oscillator with $\gamma = 1.7$ prediction (No.3).

5.3 Hardening Duffing oscillator

When α and β have the same sign, one has the hardening Duffing oscillator. Here we fix $\alpha = 5, \beta = 1, \delta = 0.02, \omega = 0.5$ and $\gamma = 8$. The prediction results obtained with the neural machine are shown below.

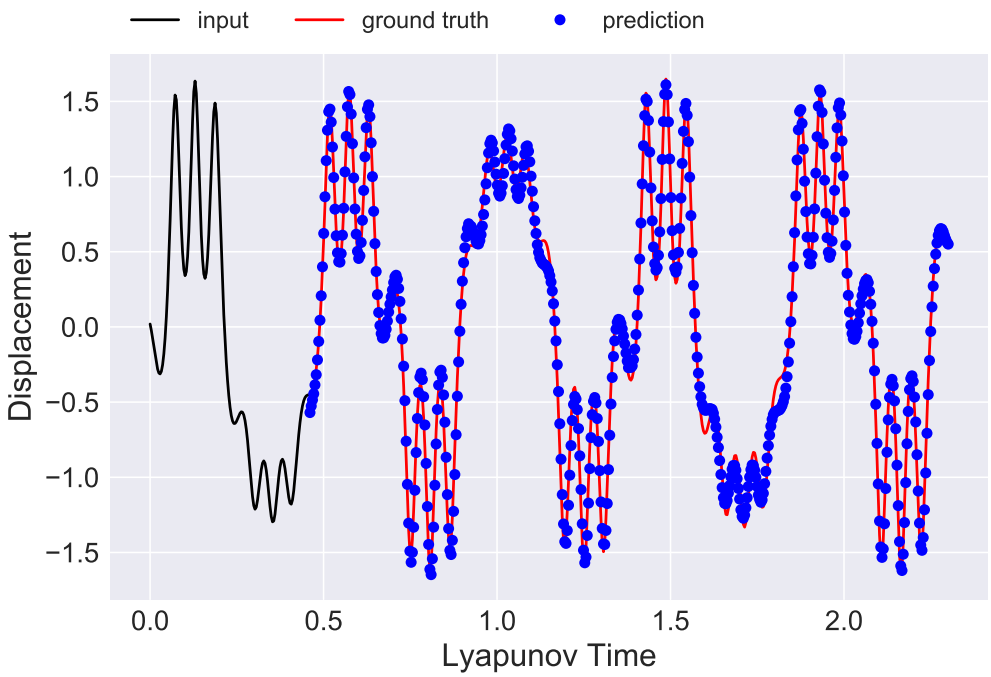

Figure 5.14: Hardening forced Duffing oscillator prediction (No.1).

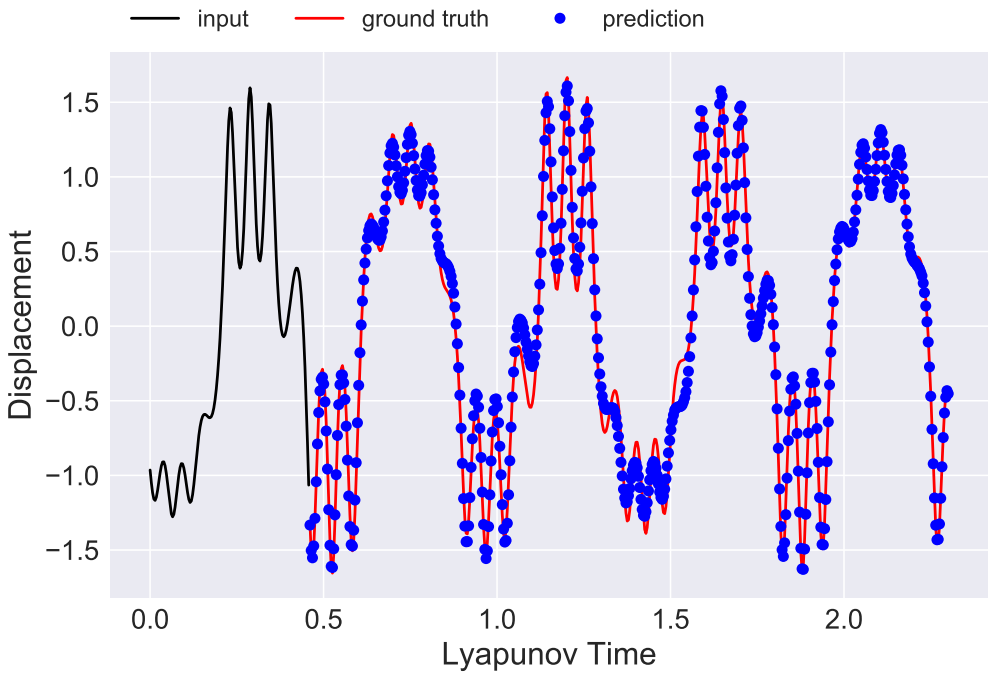

Figure 5.15: Hardening forced Duffing oscillator prediction (No.2).

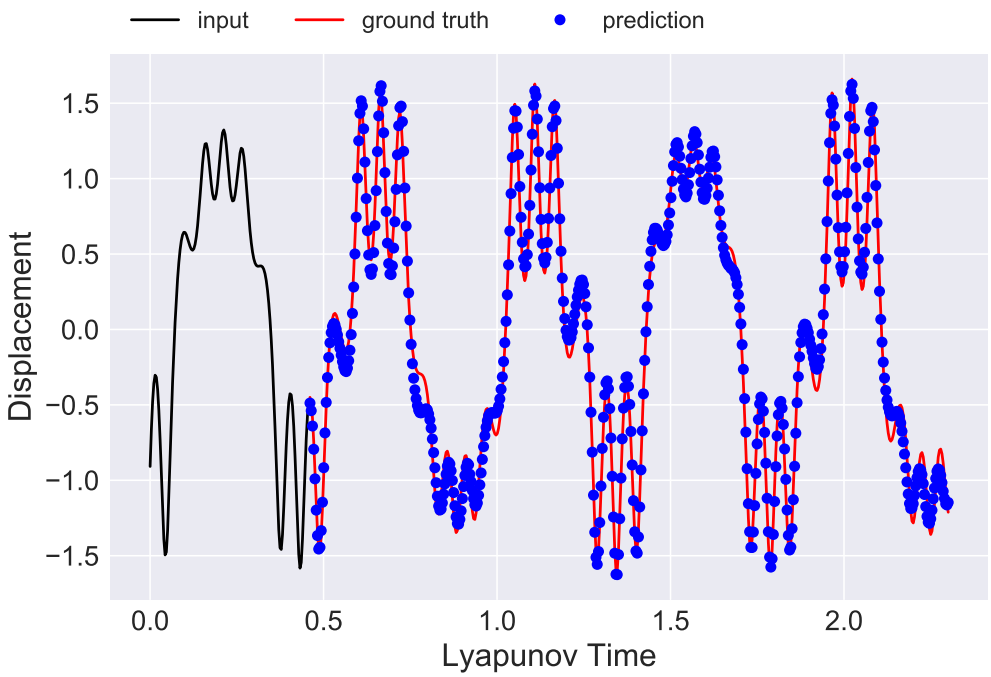

Figure 5.16: Hardening forced Duffing oscillator prediction (No.3).

Chapter 6: Concluding Remarks

6.1 Summary of contributions

In this dissertation, data-driven approaches have been studied for modeling complex behaviors, such as transient events and aperiodic motions, in dynamical systems.

In the first part, the author has applied data-driven approach in the context of rogue waves and explained how to use field measurements to unveil the underlying physical mechanism in the generation of ocean extreme waves. Specifically, from the proposed stochastic wave interference model and the results obtained, it can be inferred that extreme waves in the unidirectional sea might occur as a result of the synchronization of a relative small number of interfering wave components. With this model, one can help explain the observed wave probability distribution better than a model based on superposition of linear waves in the large wave height domain. It has been shown that wave modulation and phase interference are crucial for understanding the occurrence of rogue waves. The stochastic model includes wave envelope modulation to take into account the Benjamin-Feir instability in unidirectional deep water and allows for phase variations, which are essential for phase synchronization at the exact location of rogue wave occurrence. Given the

results, the author believes that the phase information is important for forecasting rogue waves. The proposed stochastic method can be used to modify the widely accepted sinusoidal waves as the basic components in modeling unidirectional ocean waves, as the method proposed here is inherently consistent with nonlinear wave evolution and interactions.

In the second part, the author has constructed a deep recurrent neural network, called a neural machine, and illustrated the long-term prediction capability of this machine for chaotic systems, including the Lorenz'63 system, the Lorenz'96 system, the Kuramoto-Sivashinsky system and forced Duffing systems. This neural machine can be easily adapted for forecasting of the behaviour of other chaotic systems without a change in the configuration, except for some hyperparameter tuning. It is believed that a significant advantage of this machine is that once it is trained by data simulated from a certain dynamical system, it can be used to forecast dynamical behaviours of the same system starting from various initial conditions which have not been used to train the neural machine. Therefore, the prediction of the behavior of a certain dynamical system from an arbitrary time instant is made possible by the neural machine, without requiring continuous, non-stop monitoring of the previous history data stream. This prediction that can be referred to as a neural machine prediction and the network are found to be quite suitable for chaotic time series forecasting.

6.2 Recommendations for future work

For future work, one could study the limits on long-term forecasting of different chaotic systems and incorporate the constructed network as a surrogate model in numerical weather forecasting and data assimilation.

- It would be interesting to see how the neural machine would work with the field data sets, given that these data sets are prone to noise and errors from difference sources. Extra attention needs to be paid towards the uncertainty and instability introduced by the above unfavourable conditions. Specifically, weight functions should be applied to the loss function to average out the effects of anomalies in the signal, to ensure the consistency and accuracy in the forecasting.

- Predictions based on partial observations through the neural machine is also an interesting direction to explore. One should face the reality that full observation of the state space of most dynamical system is not always feasible. Take the numerical weather forecasting (NWF) as an example. The state of the weather in certain area is not fully available to the forecaster. Data measurement may not give extensive details about the meteorological quantities used in NWF. Most times, these quantities are dependent on each other. Therefore, the time history of variable X incorporates the evolution information of variable Y. Does the prediction of Y require the history of dependent variable X? How about when Y is a function of X but not vice versa; that is,

X is an independent variable? The answer to this type of questions is strongly related to the discovery of causality in the system. Much pioneering work has been done to detect the causality chains within a dynamical system, but not through a data-driven perspective.

- Automatic determination of time step and length of the input time series in generating a forecast can be quite challenging to do. According to Takens' embedding theorem, the attractor built from the time series of a single variable is diffeomorphic to the original attractor built from the whole state variables under certain conditions. The right choice of time delay and embedding dimension is crucial. Likewise, the time steps and lengths of input signals can be of paramount importance for generation of long-term accurate forecasting. Reinforcement learning (RL), which has been very successful in the most recent artificial intelligence odyssey, can be applied in this direction to explore the optimal combination of time interval and length.

- Neural machine forecasting provides an alternative way to assimilate data other than the traditional methods such as 4D-Var, in numerical weather forecasting. Therefore, a comparative study between the alternative and the traditional methods can be performed to understand the relative advantage and shortcomings of each method. This study can include the computational cost, accuracy, and robustness. Moreover, a hybrid method based on the combination of 4D-Var and neural machine can be created to improve the aforementioned aspects in numerical weather forecasting.

Appendix A: Neural Network Training

A.1 Details of Lorenz'63 system

The training and testing details for the Lorenz'63 system are listed below.

- dt = 0.05, used for numerically integrating (4.1)
- Batch size = 32
- Input time series length $n_x = 32$
- Output time series length $n_y = 128$
- Learning rate $l = 0.001$
- Number of units of the LSTM cell hidden states: 128
- Number of stacked LSTM cells: 2
- Number of inhibitor mechanisms: 2

A.2 Details of Lorenz'96 system

The training and testing details for the Lorenz'96 system are as listed below.

- dt = 0.05, used for numerically integrating (4.3)

- Batch size = 32
- Input time series length $n_x = 32$
- Output time series length $n_y = 64$
- Learning rate $l = 0.001$
- Number of units of the LSTM cell hidden states: 256
- Number of stacked LSTM cells: 2
- Number of inhibitor mechanisms: 4

A.3 Details of KS system

The training and testing details for the Kuramoto-Sivashinsky system are as listed below.

- dt = 0.25, used for numerically integrating (4.4)
- Batch size = 32
- Input time series length $n_x = 128$
- Output time series length $n_y = 128$
- Learning rate $l = 0.001$
- Number of units of the LSTM cell hidden states: 512
- Number of stacked LSTM cells: 2
- Number of inhibitor mechanisms: 4

A.4 Details of forced Duffing system

The training and testing details for the forced duffing oscillator are as listed below.

- dt = 0.25, used for numerically integrating (5.13)
- Batch size = 64
- Input time series length $n_x = 128$
- Output time series length $n_y = 256$
- Learning rate $l = 0.001$
- Number of units of the LSTM cell hidden states: 128
- Number of stacked LSTM cells: 32
- Number of inhibitor mechanisms: 16

Appendix B: Additional Results from Neural Machine Forecasting

For each of the dynamical systems discussed in Chapter 4, the following results are obtained from the neural machine digesting on different initial conditions, but with the same hyperparameters as specified in Appendix A. The initial condition is varied to generate different historical data sets in order to demonstrate the forecasting capacity of the neural machine. The index numbers in the caption of the following figures follow those in Chapter 4.

B.1 Lorenz'63 system

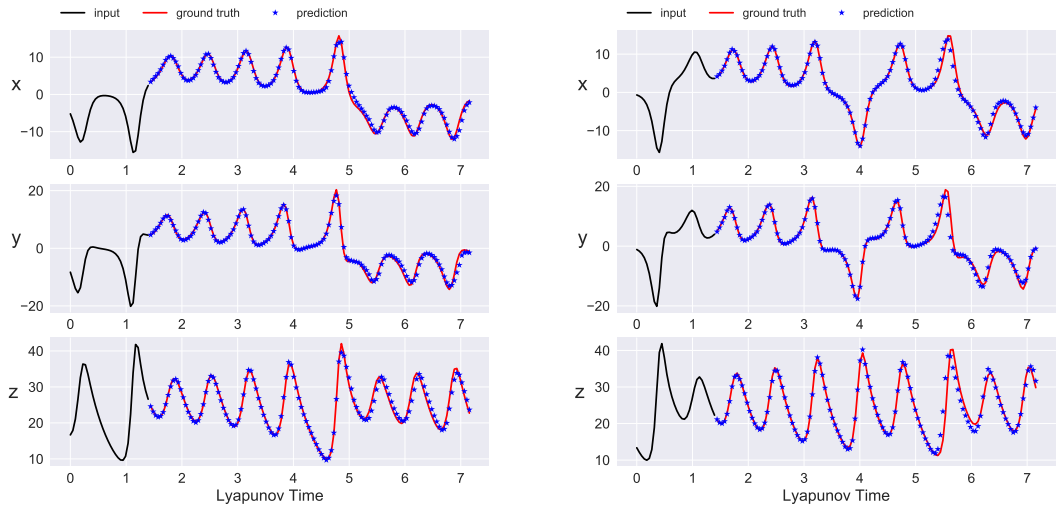

Figure B.1: *left*: Lorenz'63 prediction (No.5); *right*: Lorenz'63 prediction (No.6).

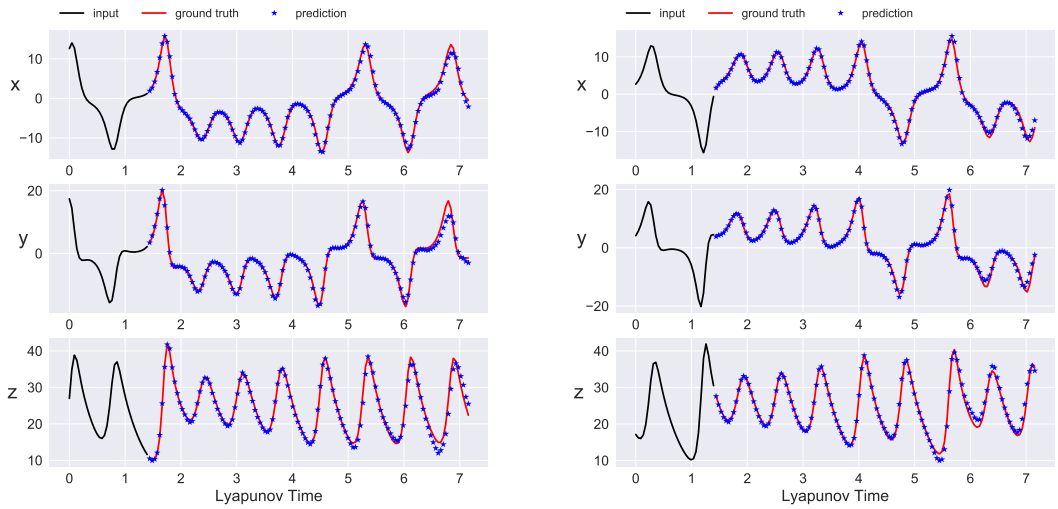

Figure B.2: *left*: Lorenz'63 prediction (No.7); *right*: Lorenz'63 prediction (No.8).

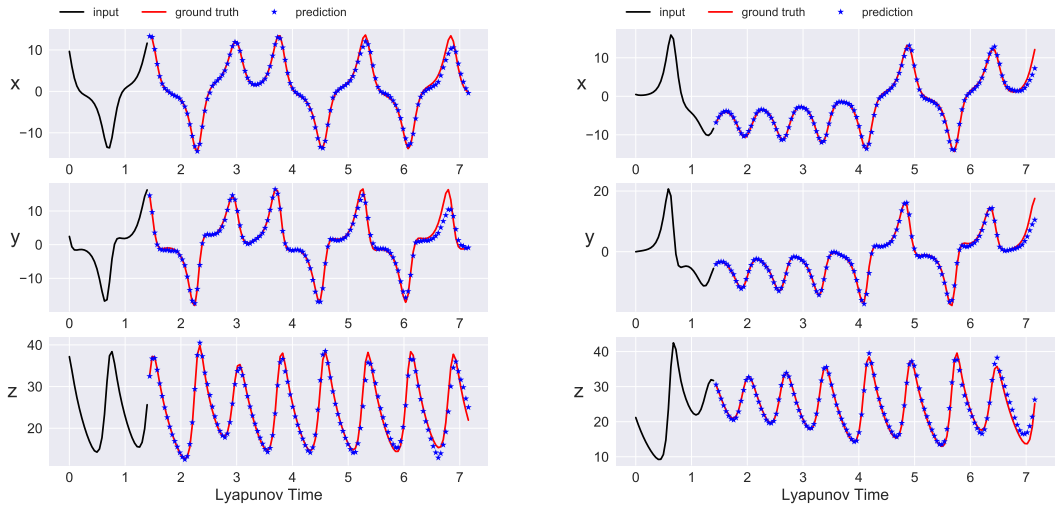

Figure B.3: *left*: Lorenz'63 prediction (No.9); *right*: Lorenz'63 prediction (No.10).

B.2 Lorenz'96 system

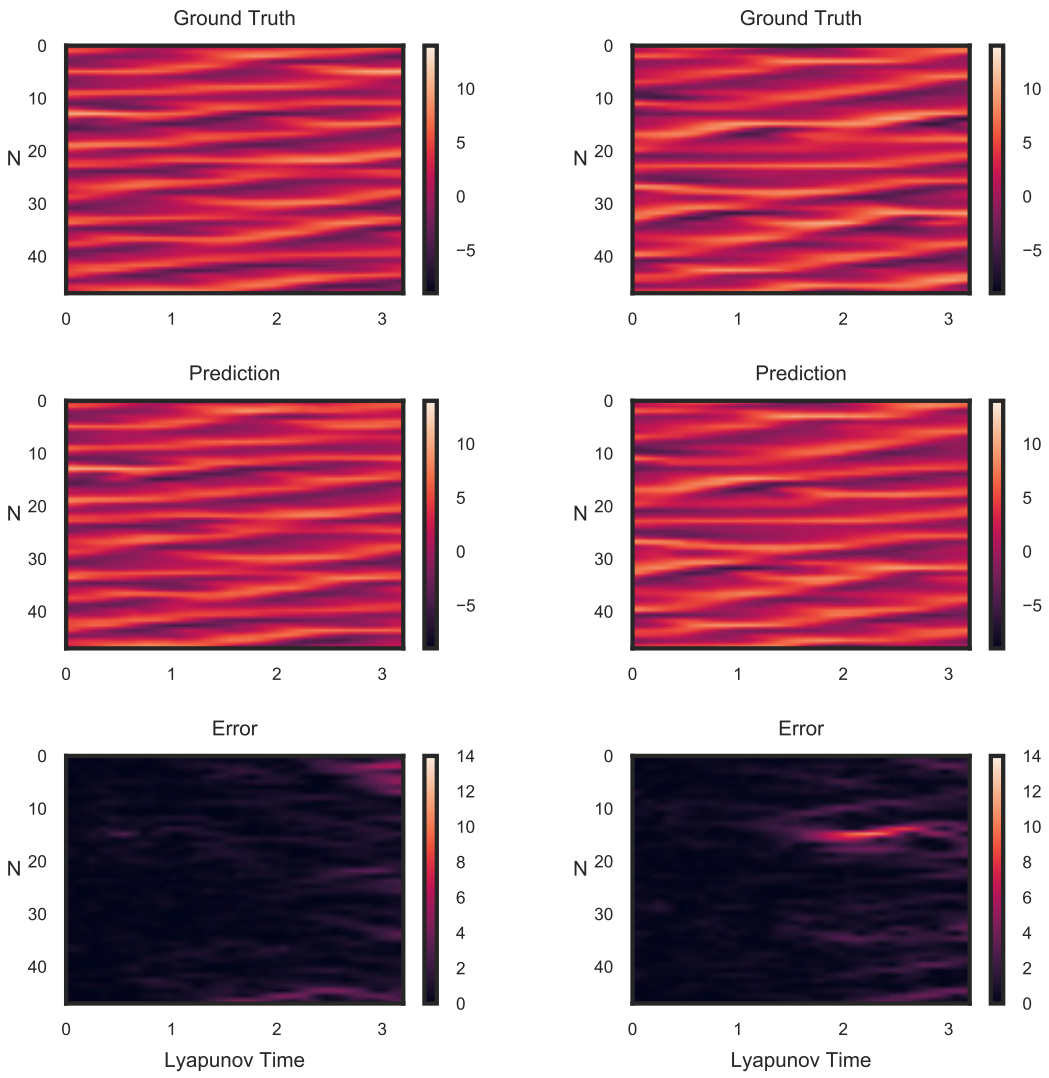

Figure B.4: *left*: Lorenz'96 prediction (No.4); *right*: Lorenz'96 prediction (No.5).

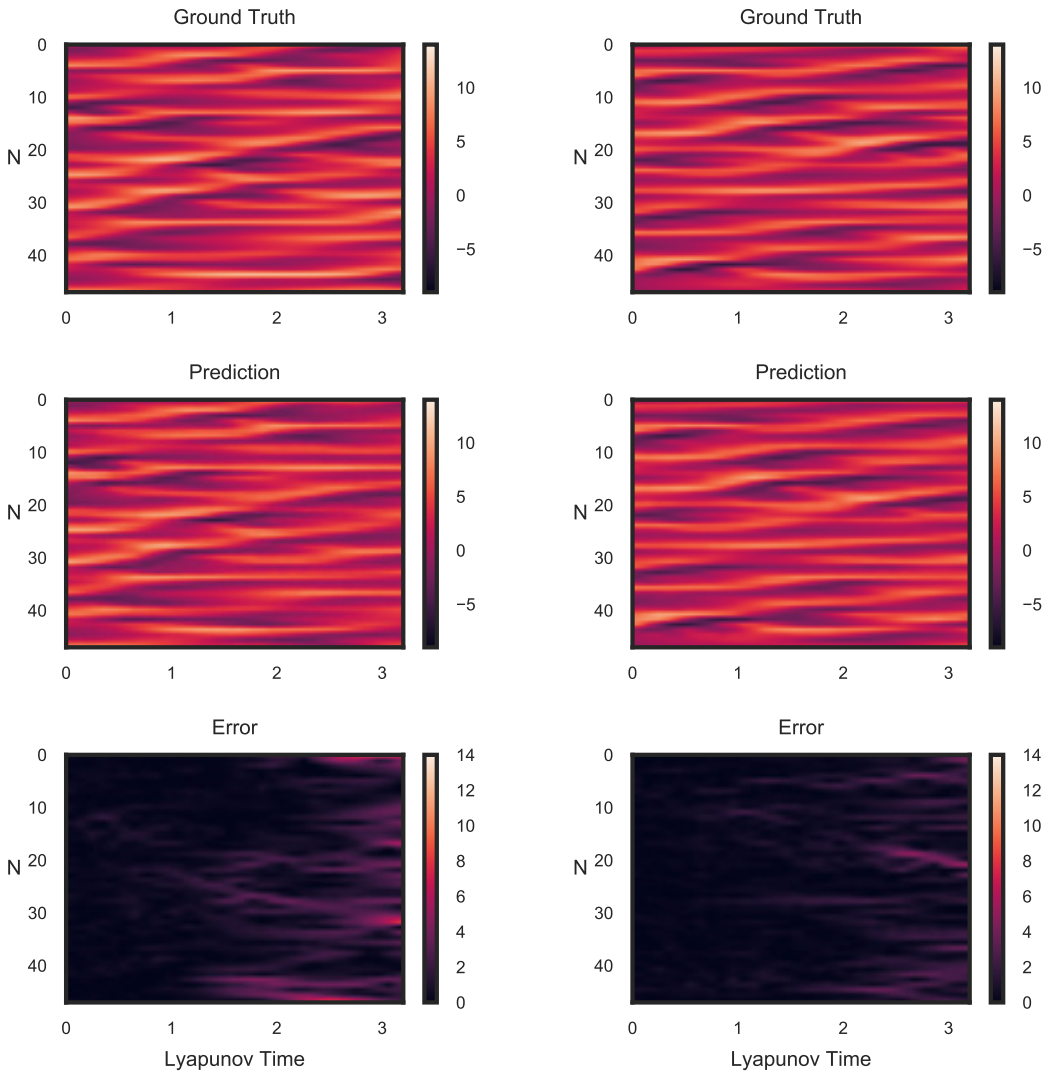

Figure B.5: *left*: Lorenz'96 prediction (No.6); *right*: Lorenz'96 prediction (No.7).

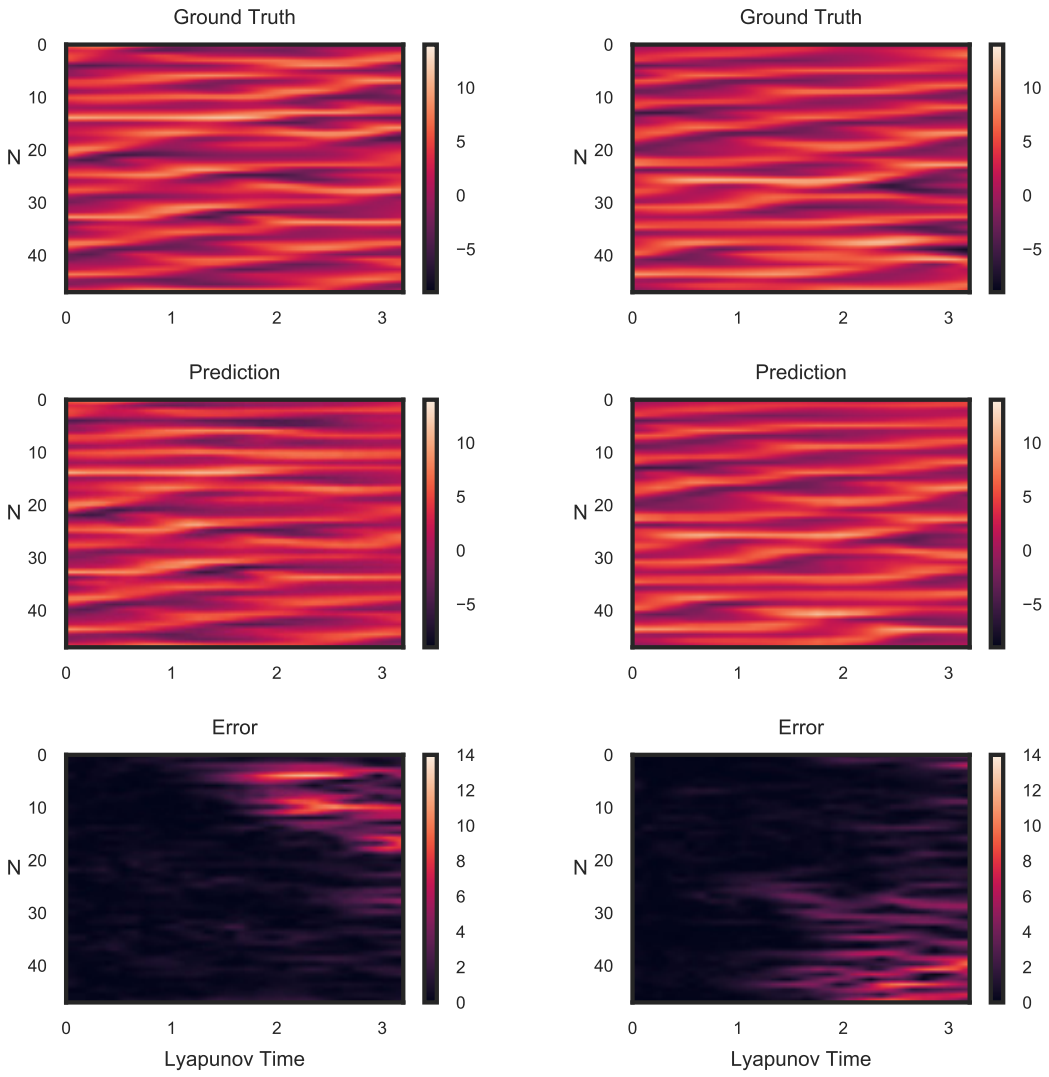

Figure B.6: *left*: Lorenz'96 prediction (No.8); *right*: Lorenz'96 prediction (No.9).

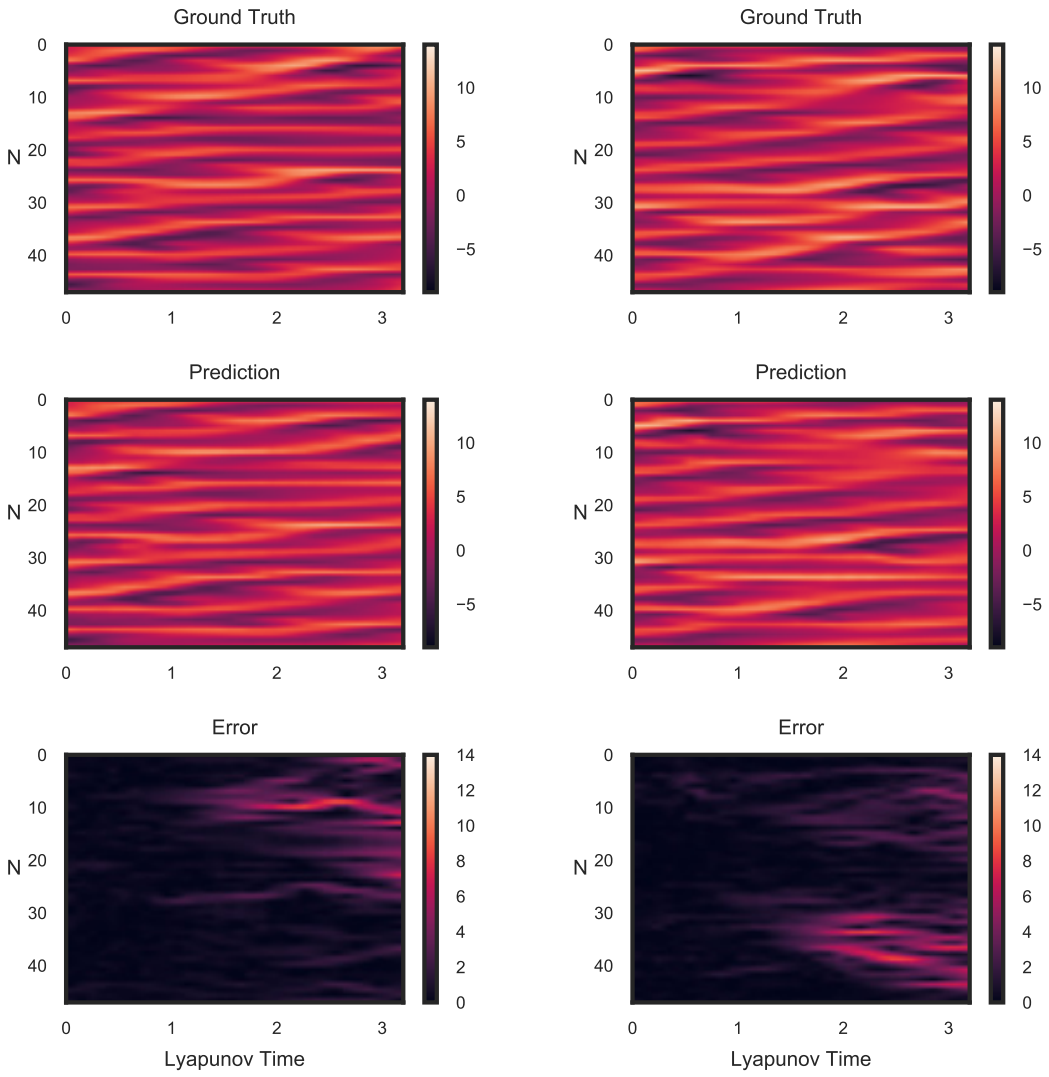

Figure B.7: *left*: Lorenz'96 prediction (No.10); *right*: Lorenz'96 prediction (No.11).

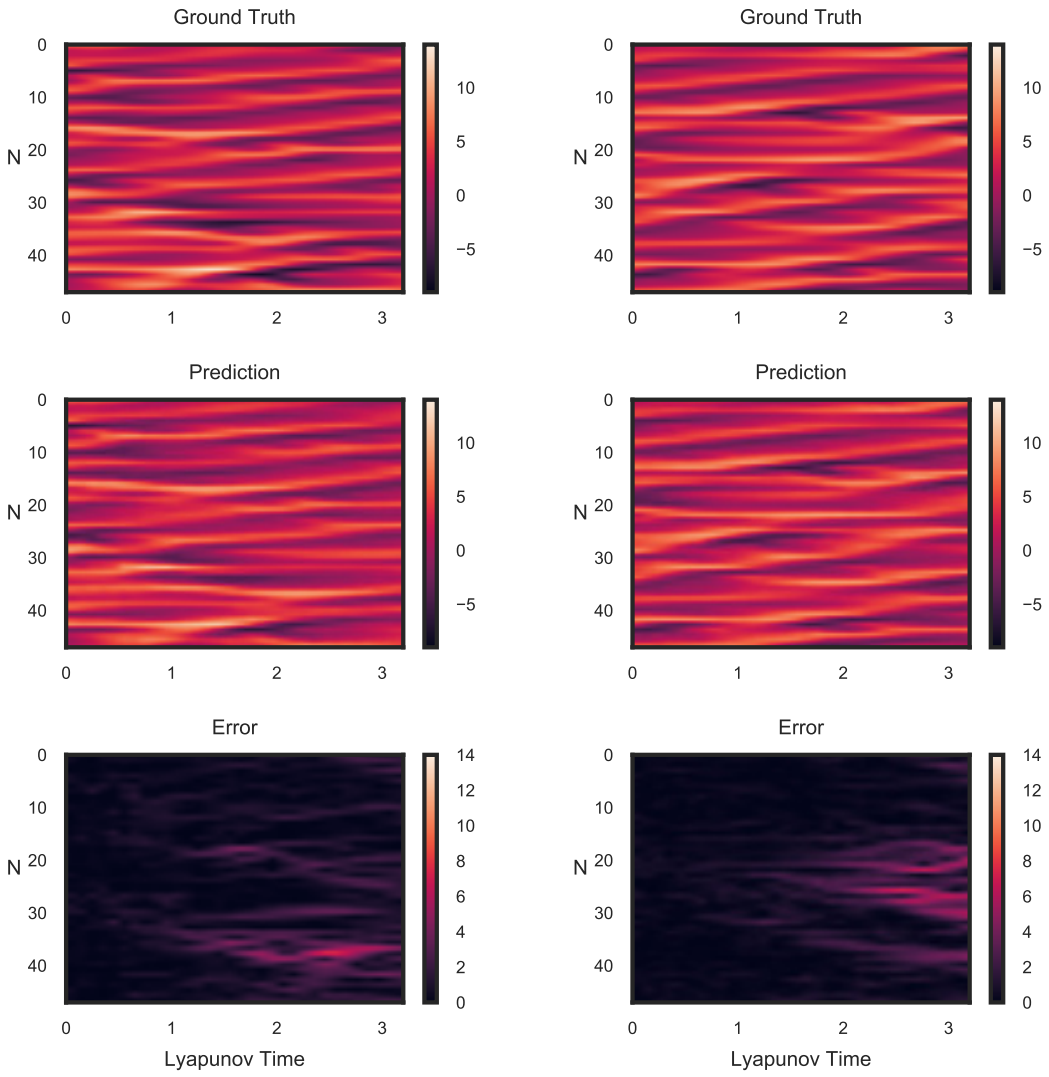

Figure B.8: *left*: Lorenz'96 prediction (No.12); *right*: Lorenz'96 prediction (No.13).

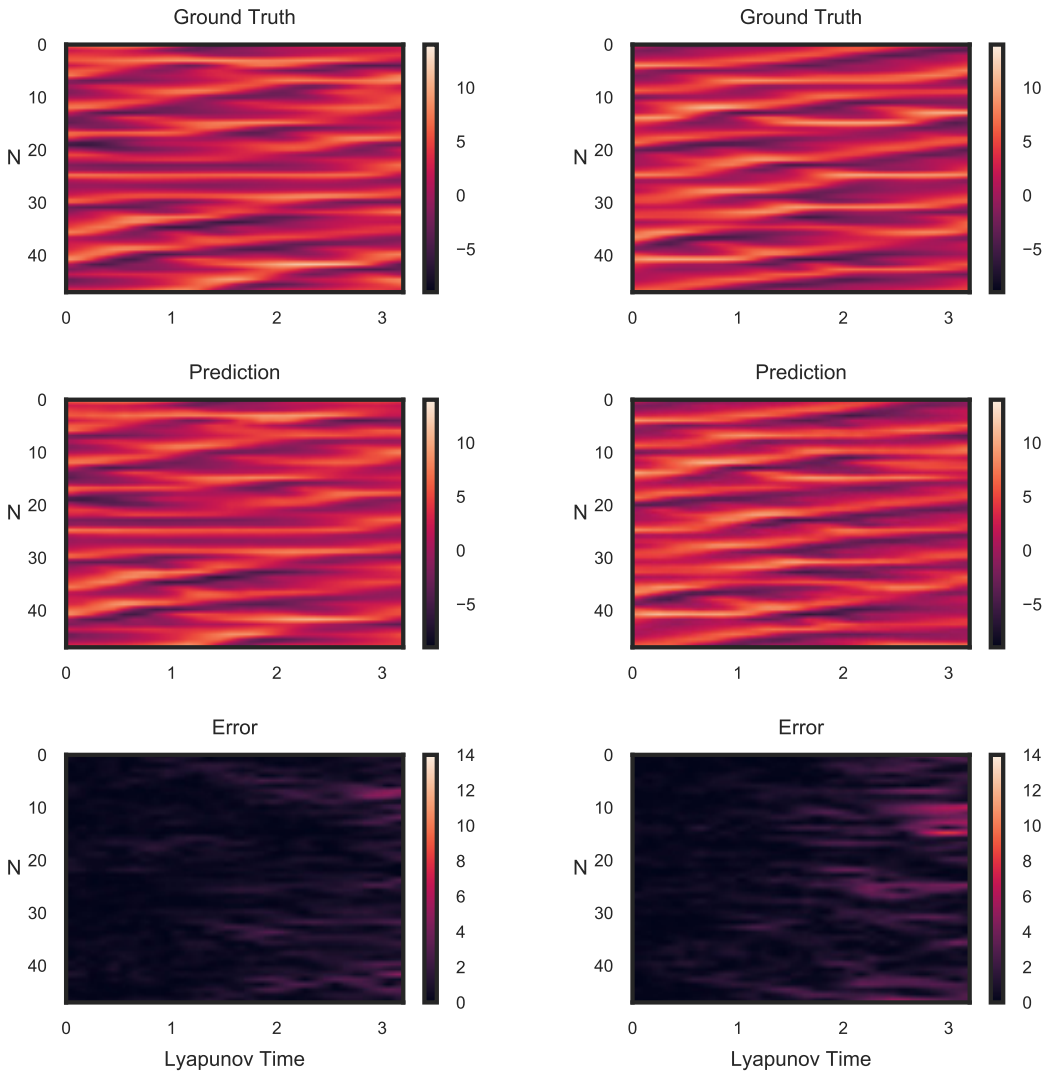

Figure B.9: *left*: Lorenz'96 prediction (No.14); *right*: Lorenz'96 prediction (No.15).

B.3 Kuramoto-Sivashinsky system

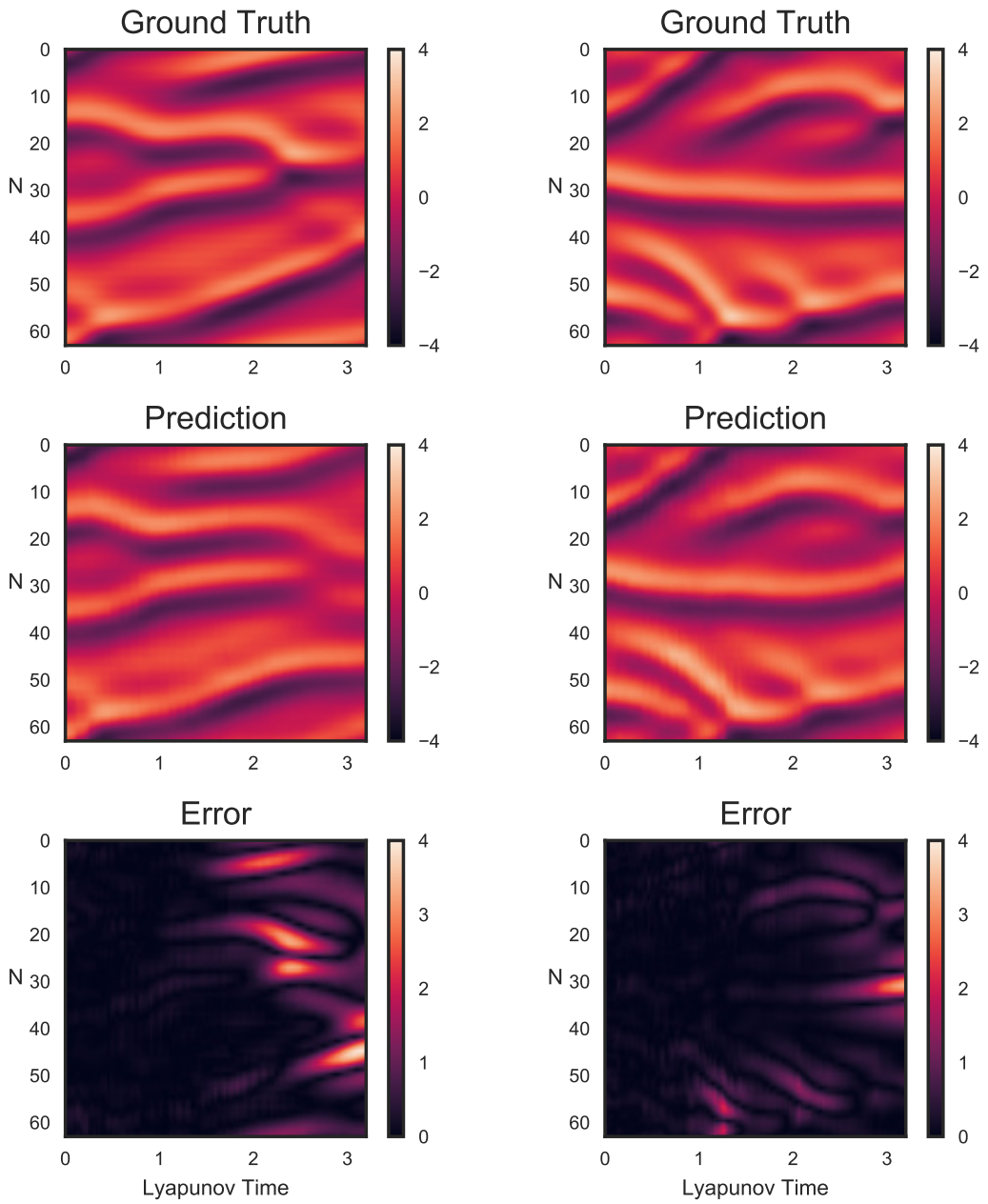

Figure B.10: *left*: KS prediction (No.4); *right*: KS prediction (No.5).

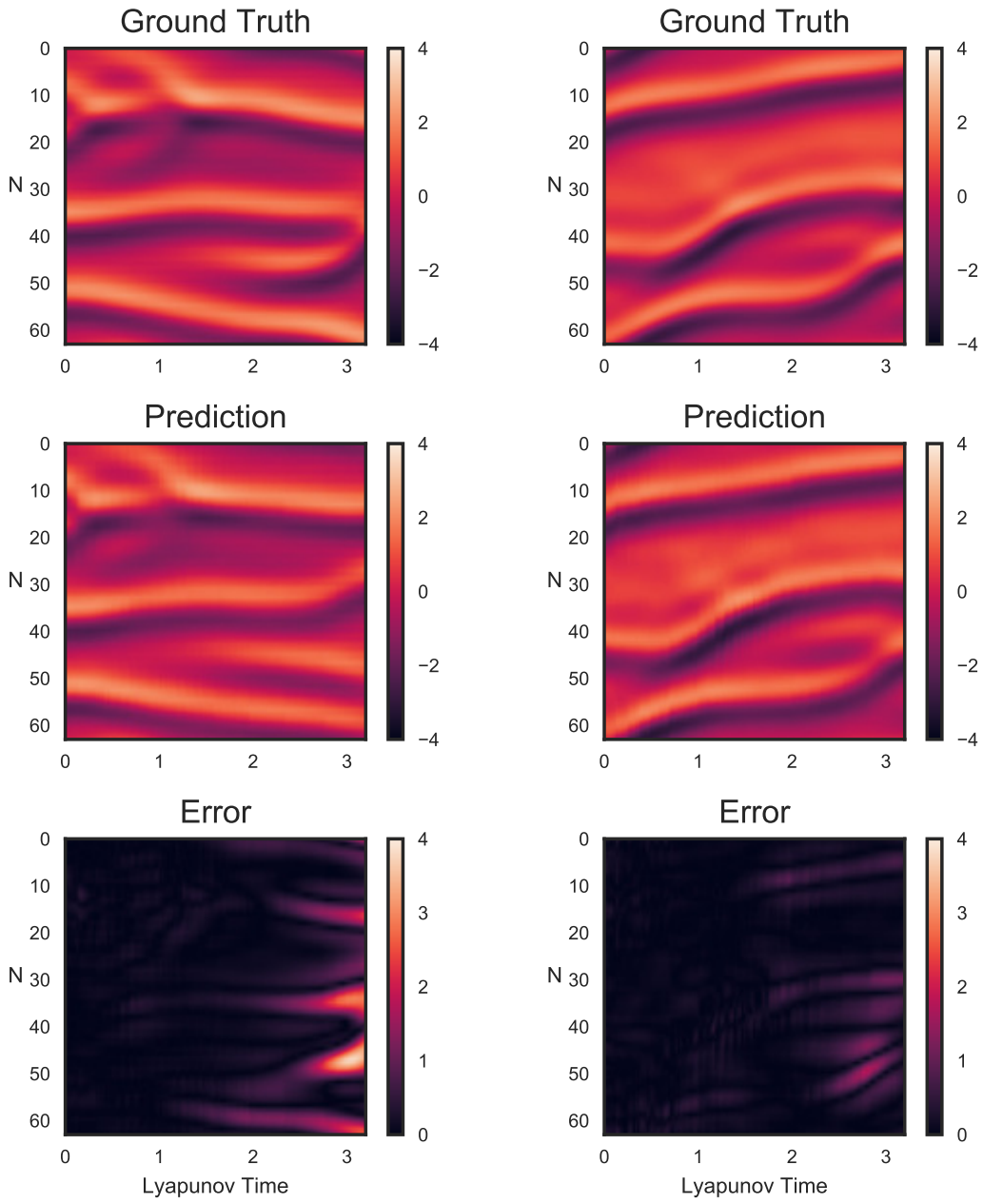

Figure B.11: *left*: KS prediction (No.6); *right*: KS prediction (No.7).

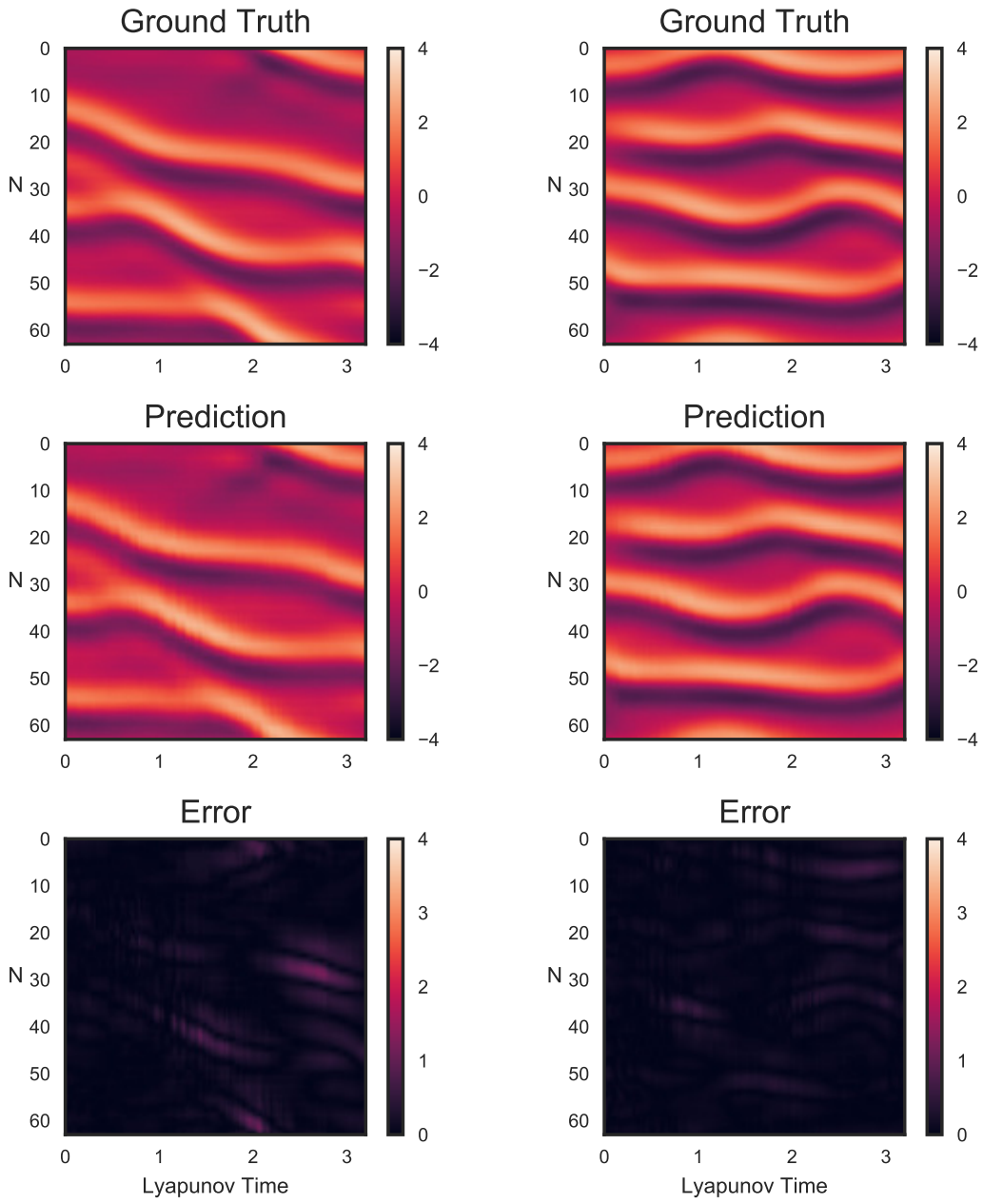

Figure B.12: *left*: KS prediction (No.8); *right*: KS prediction (No.9).

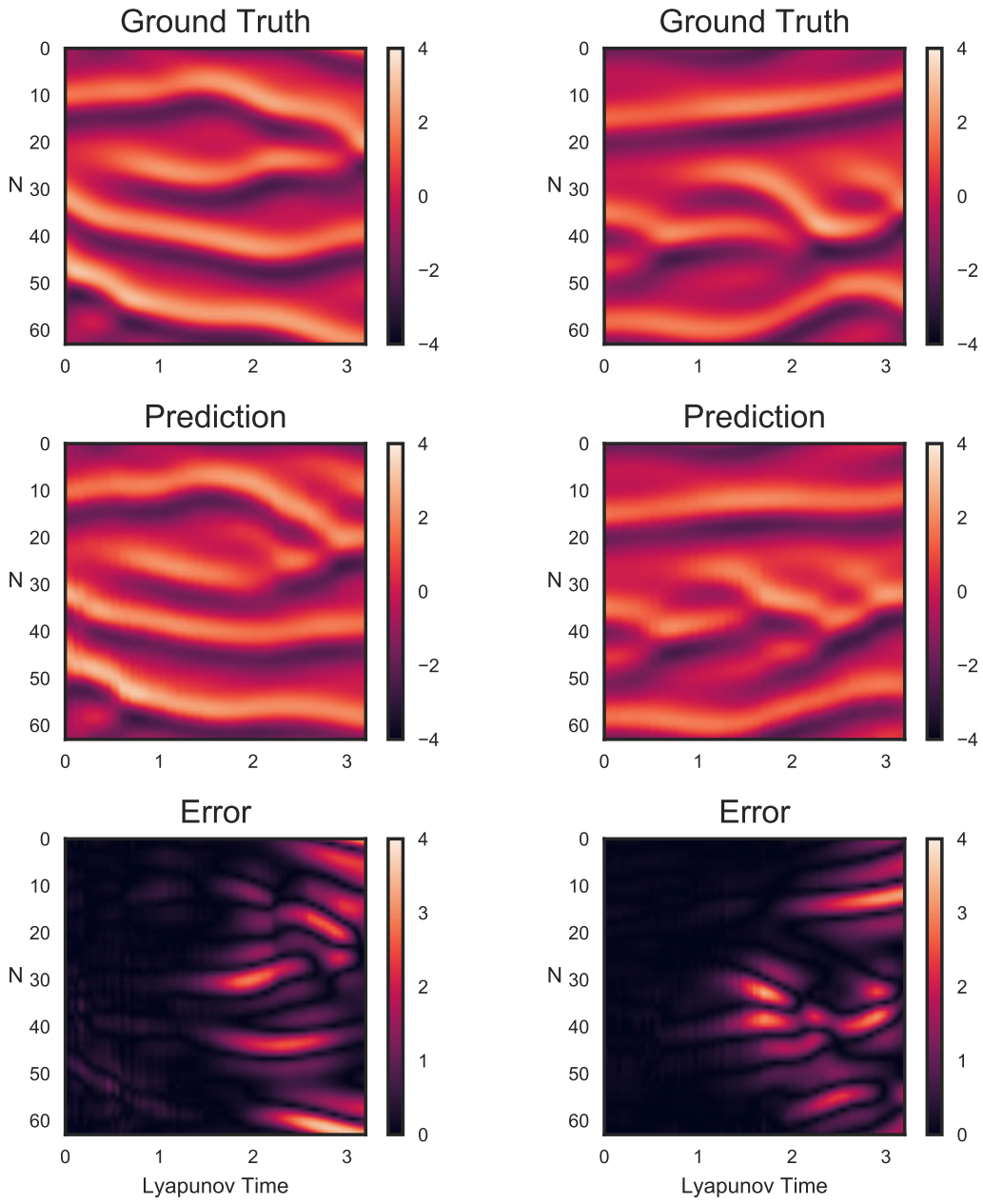

Figure B.13: *left*: KS prediction (No.10); *right*: KS prediction (No.11).

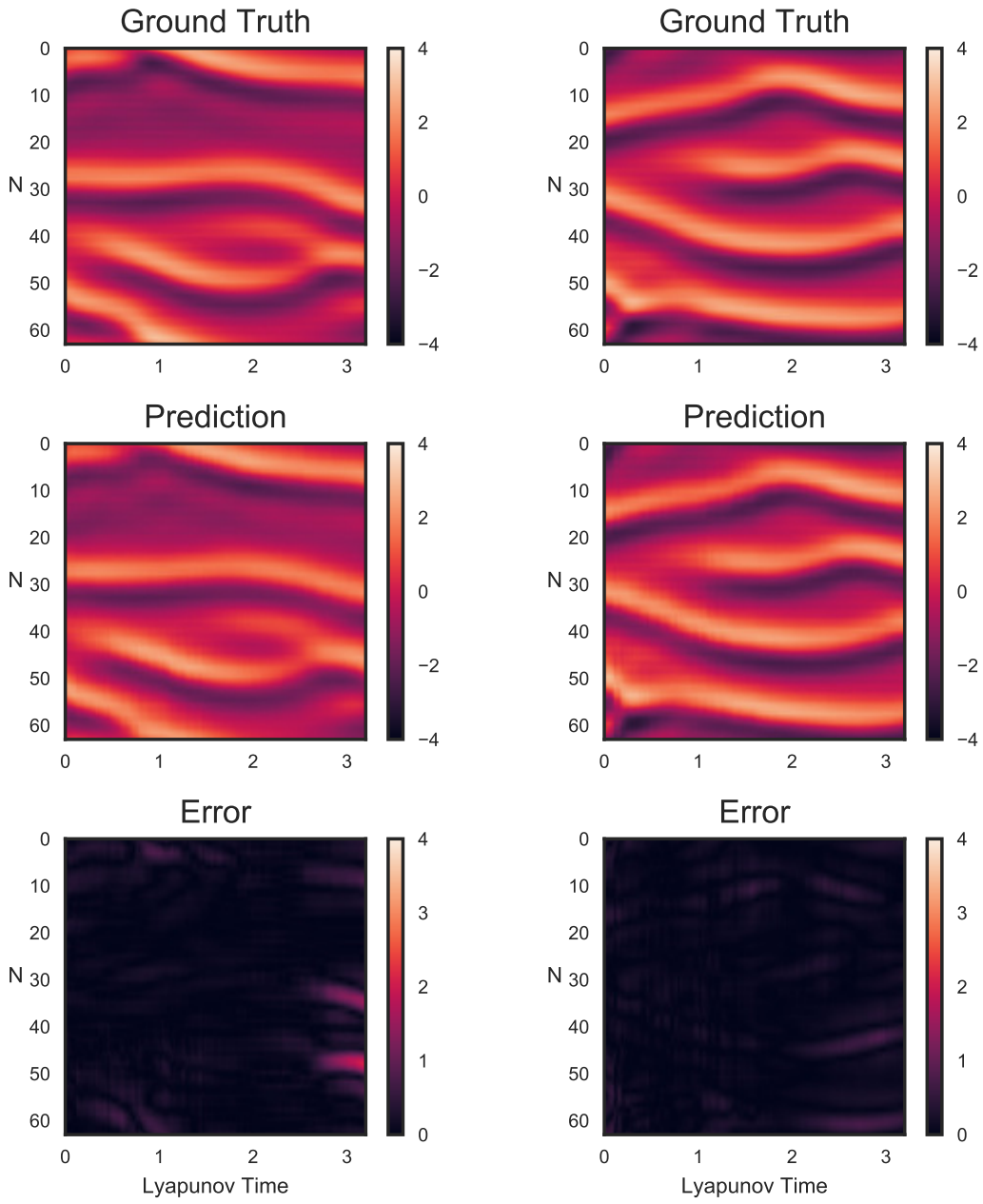

Figure B.14: *left*: KS prediction (No.12); *right*: KS prediction (No.13).

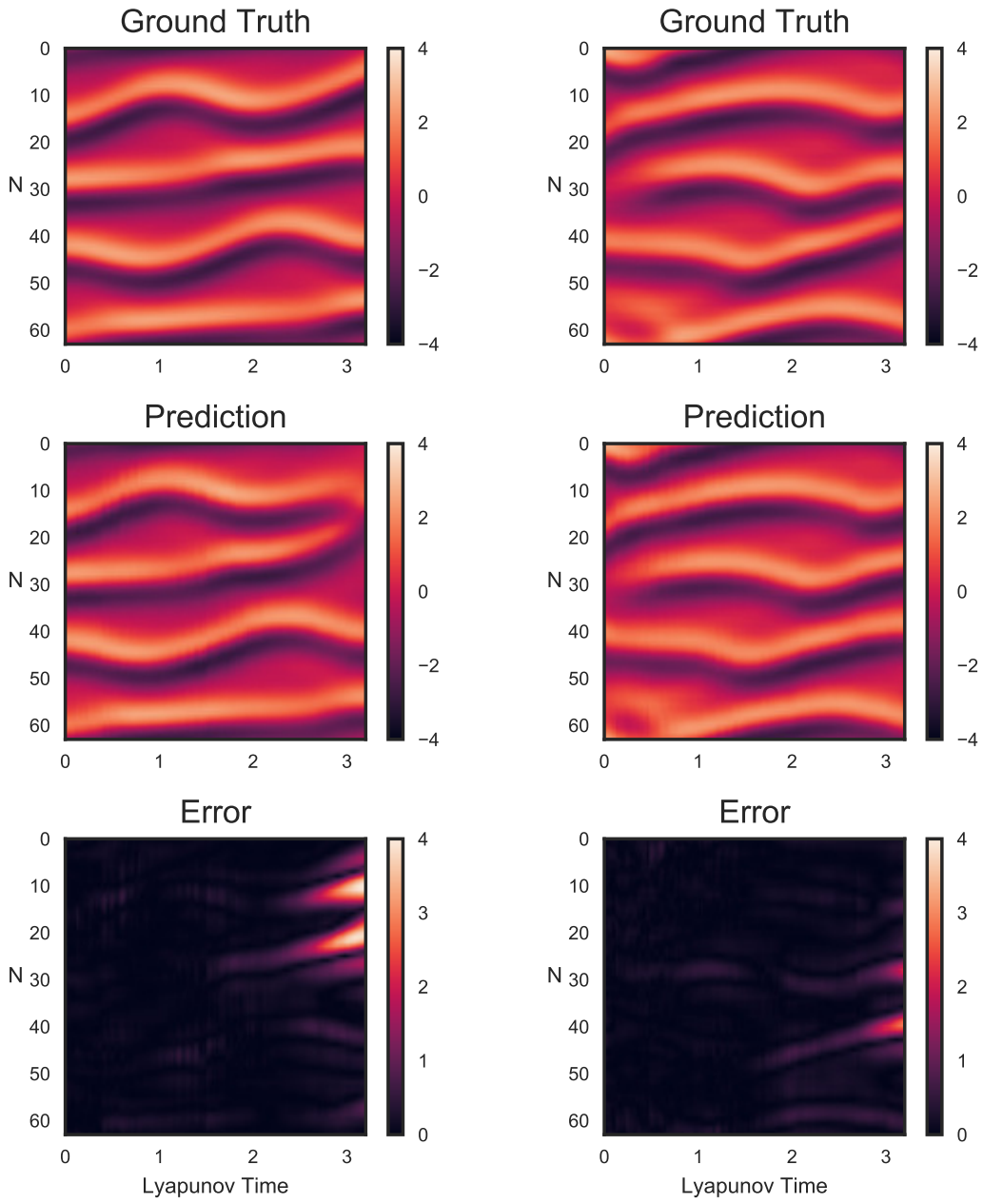

Figure B.15: *left*: KS prediction (No.14); *right*: KS prediction (No.15).

B.4 Softening forced Duffing oscillator with $\gamma = 0.5$

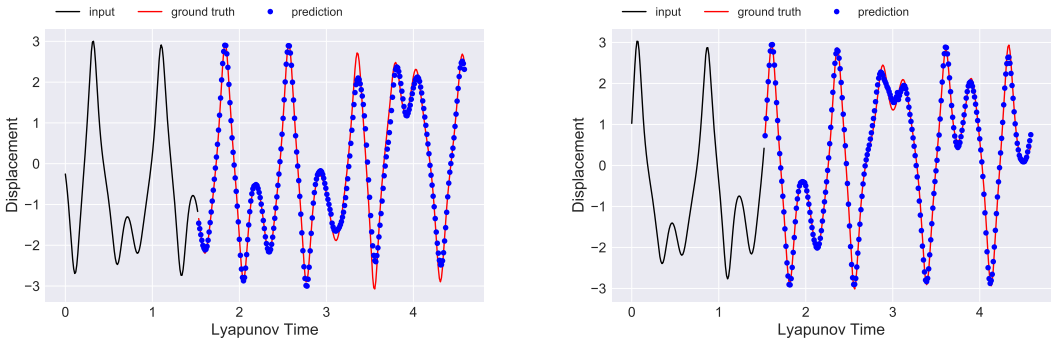

Figure B.16: *left*: Softening Duffing with $\gamma = 0.5$ prediction (No.4); *right*: Softening Duffing with $\gamma = 0.5$ prediction (No.5).

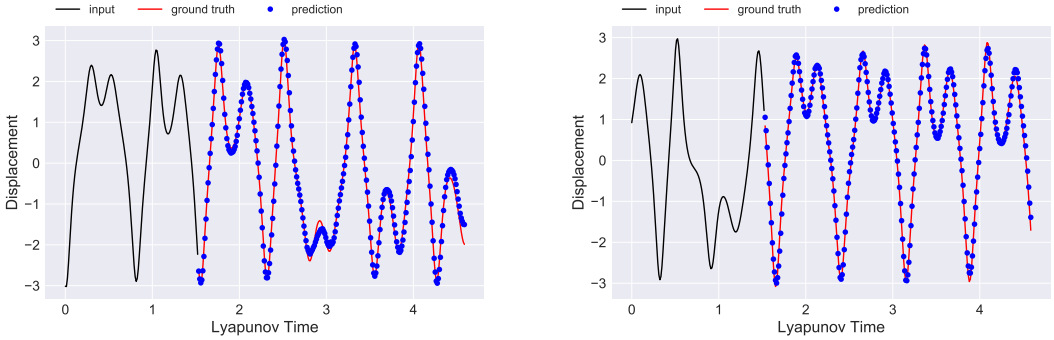

Figure B.17: *left*: Softening Duffing with $\gamma = 0.5$ prediction (No.6); *right*: Softening Duffing with $\gamma = 0.5$ prediction (No.7).

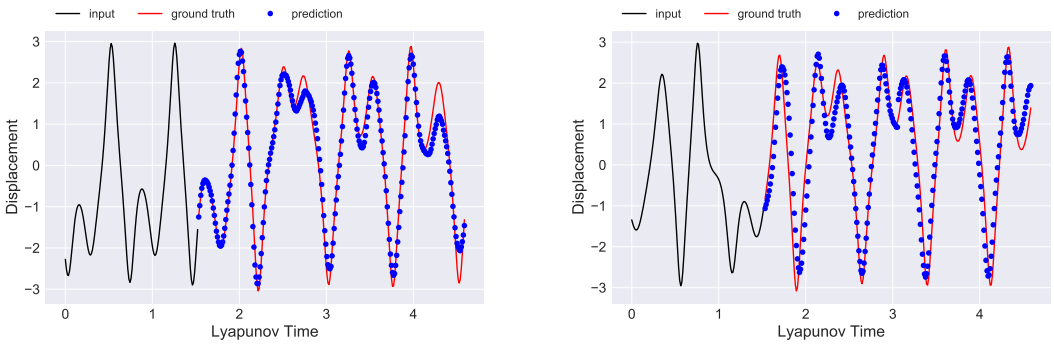

Figure B.18: *left*: Softening Duffing with $\gamma = 0.5$ prediction (No.8); *right*: Softening Duffing with $\gamma = 0.5$ prediction (No.9).

B.5 Softening forced Duffing oscillator with $\gamma = 1.7$

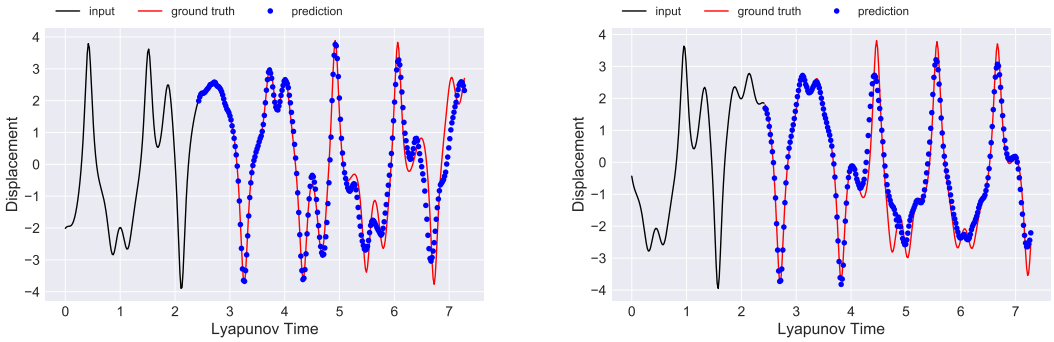

Figure B.19: *left*: Softening Duffing with $\gamma = 1.7$ prediction (No.4); *right*: Softening Duffing with $\gamma = 1.7$ prediction (No.5).

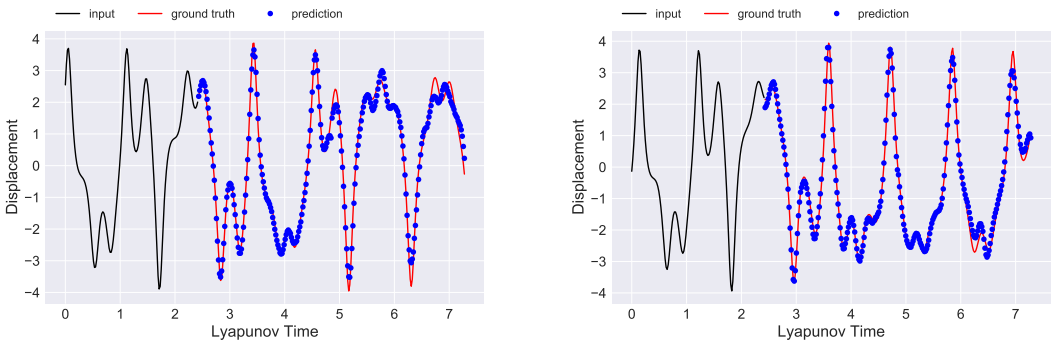

Figure B.20: *left*: Softening Duffing with $\gamma = 1.7$ prediction (No.6); *right*: Softening Duffing with $\gamma = 1.7$ prediction (No.7).

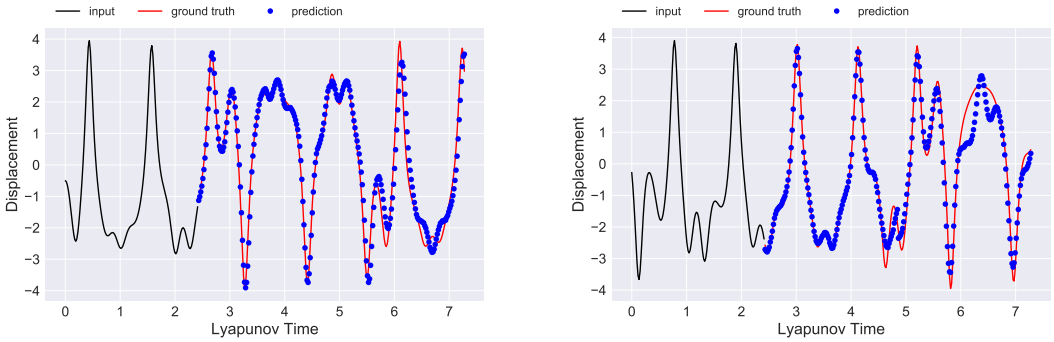

Figure B.21: *left*: Softening Duffing with $\gamma = 1.7$ prediction (No.8); *right*: Softening Duffing with $\gamma = 1.7$ prediction (No.9).

B.6 Hardening forced Duffing oscillator

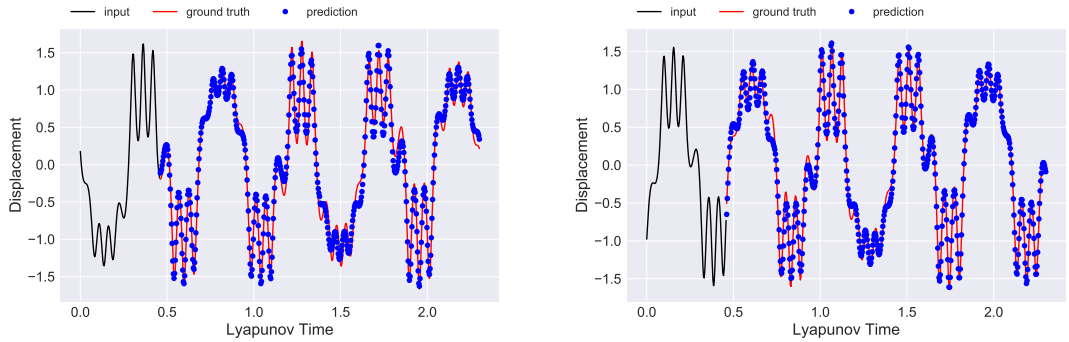

Figure B.22: *left*: Hardening Duffing prediction (No.4); *right*: Hardening Duffing prediction (No.5).

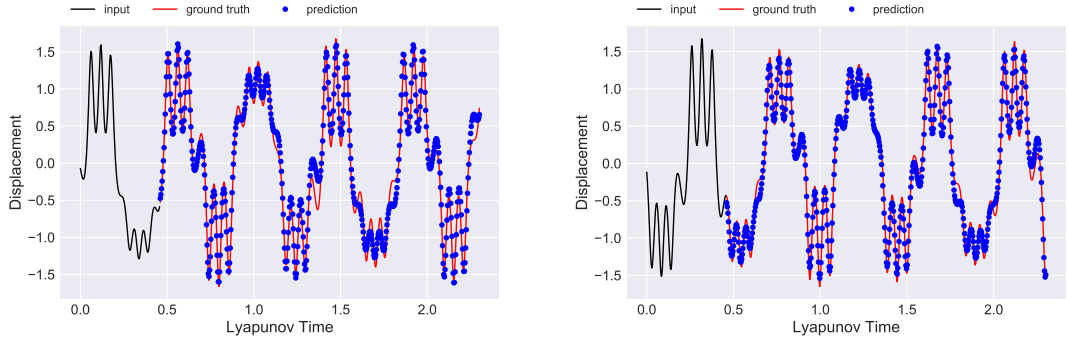

Figure B.23: *left*: Hardening Duffing prediction (No.6); *right*: Hardening Duffing prediction (No.7).

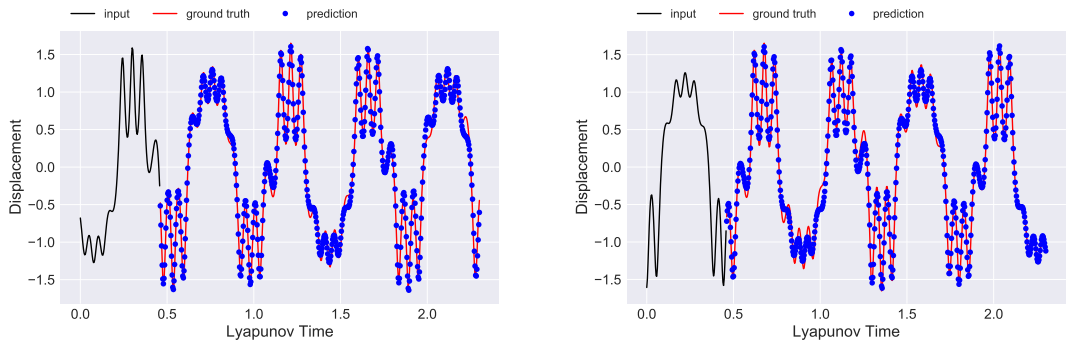

Figure B.24: *left*: Hardening Duffing prediction (No.8); *right*: Hardening Duffing prediction (No.9).

Appendix C: 4th-order Time Stepping for Stiff PDEs

Many PDEs have nonlinear terms with low-order derivatives and linear terms with high-order derivatives, such as Allen-Cahn, Burgers, Fitzhugh-Naguno, and Kuramoto-Sivashinsky equations. High-order approximations to the derivatives are desired in order to obtain good accuracy. However, most computations are restricted to second order in time by the combination of stiffness and nonlinearity. Exponential Time-differencing Runge-Kutta 4th-order (ETDRK4) method is designed to improve the temporal accuracy of such stiff PDEs [113]. This method can be illustrated briefly in the following.

Generally, a PDE can be written in the form

$$u_t = \mathcal{L}u + \mathcal{N}(u,t), \tag{C.1}$$

where $\mathcal{L}$ and $\mathcal{N}$ are linear and nonlinear operators, respectively. Discretizing the spatial derivatives will lead to a system of ODEs, which can be written as

$$u_t = \mathbf{L}u + N(u,t). \tag{C.2}$$

In order to discuss ETDRK4, one needs to first mention integrating factor (IF) and Runge-Kutta 4th-order(RK4) methods.

With IF, one uses the idea of changing the variable in PDE in order to solve

the linear part exactly, and then uses a numerical scheme to solve the transformed, nonlinear part. This has been widely applied in references [114–117].

Starting with equation (C.2), one can make a change of variable

$$v = e^{-\mathbf{L}t}u. \tag{C.3}$$

The multiplier $e^{-\mathbf{L}t}$ is called as the *integrating factor*. When working with Fourier collocation method in the spatial discretization scheme, the multiplier will be a matrix exponential. Differentiating both sides of (C.3) results in

$$v_t = -e^{-\mathbf{L}t}\mathbf{L}u + e^{-\mathbf{L}t}u_t. \tag{C.4}$$

Now if multiples (C.2) by $e^{-\mathbf{L}t}$, one can get

$$e^{-\mathbf{L}t}u_t - e^{-\mathbf{L}t}\mathbf{L}u = e^{-\mathbf{L}t}N(u), \tag{C.5}$$

which is

$$v_t = e^{-\mathbf{L}t}N(e^{\mathbf{L}t}v). \tag{C.6}$$

The removal of the linear high-order part will allow us to use any kind of time-differencing scheme, such as RK4. Let $f = e^{-\mathbf{L}t}N(e^{\mathbf{L}t}v)$. Then, the RK4 scheme reads as

$$a = hf(v_n, t_n), \tag{C.7a}$$

$$b = hf(v_n + a/2, t_n + h/2), \tag{C.7b}$$

$$c = hf(v_n + b/2, t_n + h/2), \tag{C.7c}$$

$$d = hf(v_n + c, t_n + h), \tag{C.7d}$$

$$v_{n+1} = v_n + \frac{1}{6}(a + 2b + 2c + d), \tag{C.7e}$$

where h is the time step.

ETD is algebraically similar to the IF method. Integrating both sides of (C.6) leads to

$$v_{n+1} = v_n + \int_0^h e^{-\mathbf{L}(t_n+\tau)} N(u(t_n+\tau), t_n+\tau) d\tau. \tag{C.8a}$$

$$e^{-\mathbf{L}(t_n+h)} u_{n+1} = e^{-\mathbf{L}t_n} u_n + \int_0^h e^{-\mathbf{L}(t_n+\tau)} N(u(t_n+\tau), t_n+\tau) d\tau. (change\ of\ variable) \tag{C.8b}$$

$$u_{n+1} = e^{\mathbf{L}h} u_n + e^{\mathbf{L}h} \int_0^h e^{-\mathbf{L}\tau} N(u(t_n+\tau), t_n+\tau) d\tau. (cancel\ e^{-\mathbf{L}t_n}) \tag{C.8c}$$

This equation is exact since no approximation is introduced at this point. Cox and Matthews [118] have provided a generic formula to obtain high-order approximations

$$u_{n+1} = e^{\mathbf{L}h} u_n + h \sum_{m=0}^{s-1} g_m \sum_{k=0}^{m} (-1)^k \binom{m}{k} N_{n-k}, \tag{C.9}$$

where s is the order scheme, and g_m can be obtained by the recurrence relation

$$\mathbf{L}h g_0 = e^{\mathbf{L}h} - \mathbf{I}, \tag{C.10a}$$

$$\mathbf{L}h g_{m+1} + \mathbf{I} = g_m + \frac{1}{2} g_{m-1} + \frac{1}{3} g_{m-2} + \cdots + \frac{g_0}{m+1}, \quad m \geq 0. \tag{C.10b}$$

They also reported the RK4 version of the above formula in the matrix form as

$$a_n = e^{\mathbf{L}h/2} u_n + \mathbf{L}^{-1}(e^{\mathbf{L}h/2} - \mathbf{I}) N(u_n, t_n), \tag{C.11a}$$

$$b_n = e^{\mathbf{L}h/2} u_n + \mathbf{L}^{-1}(e^{\mathbf{L}h/2} - \mathbf{I}) N(u_n, t_n + h/2), \tag{C.11b}$$

$$c_n = e^{\mathbf{L}h/2} a_n + \mathbf{L}^{-1}(e^{\mathbf{L}h/2} - \mathbf{I})(2N(b_n, t_n + h/2) - N(u_n, t_n)), \tag{C.11c}$$

$$\begin{aligned} u_{n+1} = {} & e^{\mathbf{L}h} u_n + h^{-2}\mathbf{L}^{-3}\{[-4 - \mathbf{L}h + e^{\mathbf{L}h}(4 - 3\mathbf{L}h + (\mathbf{L}h)^2)] N(u_n, t_n) \\ & + 2[2 + \mathbf{L}h + e^{\mathbf{L}h}(-2 + \mathbf{L}h)](N(a_n, t_n + h/2) + N(b_n, t_n + h/2)) \\ & + [-4 - 3\mathbf{L}h - (\mathbf{L}h)^2 + e^{\mathbf{L}h}(4 - \mathbf{L}h)] N(c_n, t_n + h)\}. \end{aligned} \tag{C.11d}$$

It is remarked that ERDRK4 suffers from the numerical instability due to the calculation of such form

$$g(z) = \frac{e^z - 1}{z}. \tag{C.12}$$

The accurate numerical calculation of the above form is notoriously problematic [119, 120], mostly due to the cancellation error when expressing the exponentials in the numerator(considering its Taylor expansion). The coefficients

$$\alpha = h^{-2}\mathbf{L}^{-3}[-4 - \mathbf{L}h + e^{\mathbf{L}h}(4 - 3\mathbf{L}h + (\mathbf{L}h)^2)], \tag{C.13a}$$

$$\beta = h^{-2}\mathbf{L}^{-3}[2 + \mathbf{L}h + e^{\mathbf{L}h}(-2 + \mathbf{L}h)], \tag{C.13b}$$

$$\gamma = h^{-2}\mathbf{L}^{-3}[-4 - 3\mathbf{L}h - (\mathbf{L}h)^2 + e^{\mathbf{L}h}(4 - \mathbf{L}h)], \tag{C.13c}$$

are high-order analogues to $(e^z - 1)/z$. If $\mathbf{L}$ has eigenvalues close to zero, the cancellation effect will be more severe, which paralyses the ETDRK4 in practical applications. Kassam and Trefethen [113] used complex contour to bypass the direction calculation of the above coefficients. Here the author uses Padé approximation as the approach to address the numerical instability. The kth-order Padé approximation has the form

$$\phi_k(z) = \frac{1}{(k-1)!}\int_0^1 e^{z(1-x)}x^{k-1}dx, \quad for\ k = 1, 2, ... \tag{C.14}$$

The first three orders can be explicitly written as

$$\phi_1(z) = \frac{e^z - 1}{z}, \tag{C.15a}$$

$$\phi_2(z) = \frac{e^z - 1 - z}{z^2}, \tag{C.15b}$$

$$\phi_3(z) = \frac{2e^z - 2 - 2z - z^2}{2z^3}. \tag{C.15c}$$

Now the (C.13) can be written as

$$\alpha = h(\phi_1(z) - 3\phi_2(z) + 4\phi_3(z)), \tag{C.16a}$$

$$\beta = h(\phi_2(z) - 2\phi_3(z)), \tag{C.16b}$$

$$\gamma = h(-\phi_2(z) + 4\phi_3(z)), \tag{C.16c}$$

where $z = \mathbf{L}h$. Since the evaluation of fractions (C.12) has been transformed to the integration such as in the case of (C.15), the numerical stability is improved significantly.

For spatial derivatives in the periodic Kuramoto-Sivashinsky equation, the author uses Fourier spectral method to transform (4.4) into the Fourier space, resulting in

$$\widehat{u}_t(k) = -\frac{ik}{2}\widehat{u^2} + (k^2 - k^4)\widehat{u}, \tag{C.17}$$

where $k = \frac{2\pi n}{L}, n = -N/2+1, \ldots, N/2$ and $\widehat{u}(k) = \mathcal{F}(u(t))$, the Fourier transform of $u(t)$ with wave number k. Following the standard form in (C.2), the above equation can be written as

$$\widehat{u}_t = \mathbf{L}\widehat{u} + N(\widehat{u}, t), \tag{C.18}$$

where $(\mathbf{L}\widehat{u})(k) = (k^2 - k^4)\widehat{u}(k), N(\widehat{u}, t) = -\frac{ik}{2}(\mathcal{F}((\mathcal{F}^{-1}(\widehat{u}))^2))$. Then the Kuramoto-Sivashinsky equation can be solved by using ETDRK4.

Bibliography

[1] Marios Christou and Kevin Ewans. Examining a comprehensive dataset containing thousands of freak wave events: Part 2 analysis and findings. In *ASME 2011 30th International Conference on Ocean, Offshore and Arctic Engineering*, pages 827–837. American Society of Mechanical Engineers, 2011.

[2] M Onorato, T Waseda, A Toffoli, L Cavaleri, O Gramstad, PAEM Janssen, T Kinoshita, Jaak Monbaliu, N Mori, AR Osborne, and others. Statistical properties of directional ocean waves: the role of the modulational instability in the formation of extreme events. *Physical Review Letters*, 102(11):114502, 2009.

[3] M Onorato, L Cavaleri, S Fouques, O Gramstad, PAEM Janssen, Jaak Monbaliu, AR Osborne, C Pakozdi, M Serio, CT Stansberg, and others. Statistical properties of mechanically generated surface gravity waves: a laboratory experiment in a three-dimensional wave basin. *Journal of Fluid Mechanics*, 627: 235–257, 2009.

[4] Benoit B. Mandelbrot. *The fractal geometry of nature*, volume 2. WH freeman New York, 1982.

[5] Philip Holmes, John L. Lumley, Gahl Berkooz, and Clarence W. Rowley. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge University Press, 2012.

[6] M. A. Katsoulakis, A. J. Majda, and Alexandros Sopasakis. Intermittency, metastability and coarse graining for coupled deterministicstochastic lattice systems. *Nonlinearity*, 19(5):1021, 2006.

[7] Peter Deuflhard, Michael Dellnitz, Oliver Junge, and Christof Schtte. Computation of essential molecular dynamics by subdivision techniques. In *Computational molecular dynamics: challenges, methods, ideas*, pages 98–115. Springer, 1999.

[8] Andrew Majda and Xiaoming Wang. *Nonlinear dynamics and statistical theories for basic geophysical flows*. Cambridge University Press, 2006.

[9] Stephen Wolfram. Statistical mechanics of cellular automata. *Reviews of Modern Physics*, 55(3):601, 1983.

[10] Sophia B. Betzler, Andreas Wisnet, Benjamin Breitbach, Christoph Mitterbauer, Jonas Weickert, Lukas Schmidt-Mende, and Christina Scheu. Template-free synthesis of novel, highly-ordered 3d hierarchical Nb 3 O 7 (OH) superstructures with semiconductive and photoactive properties. *Journal of Materials Chemistry A*, 2(30):12005–12013, 2014.

[11] Yaneer Bar-Yam. *Dynamics of complex systems*, volume 213. Addison-Wesley Reading, MA, 1997.

[12] Ali H. Nayfeh and Balakumar Balachandran. *Applied nonlinear dynamics: analytical, computational, and experimental methods*. John Wiley & Sons, 2008.

[13] Yoshiki Kuramoto. *Chemical oscillations, waves, and turbulence*. Courier Corporation, 2003.

[14] James D. Murray. *Mathematical biology, vol. 19 of Biomathematics*. Springer, Berlin, Germany, 1989.

[15] Edward Ott. *Chaos in dynamical systems*. Cambridge University Press, 2002.

[16] Michael Baranger. Chaos, Complexity, and Entropy: A physics talk for non-physicists, April 2000.

[17] Henry J. Stephen Smith. On the integration of discontinuous functions. *Proceedings of the London Mathematical Society*, 1(1):140–153, 1874.

[18] M. Sierpinski. Sur une courbe dont tout point est un point de ramification. *Comptes rendus hebdomadaires des sances de l'Acadmie des Sciences*, 160: 302–305, January 1915.

[19] Tien-Yien Li and James A. Yorke. Period three implies chaos. *The American Mathematical Monthly*, 82(10):985–992, 1975.

[20] Edward N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, 1963.

[21] Claude Elwood Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

[22] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[23] David JC MacKay and David JC Mac Kay. *Information theory, inference and learning algorithms*. Cambridge University Press, 2003.

[24] Jaeseung Jeong, Soo Yong Kim, and Seol-Heui Han. Non-linear dynamical analysis of the EEG in Alzheimer's disease with optimal embedding dimension. *Electroencephalography and clinical Neurophysiology*, 106(3):220–228, 1998.

[25] Alfrd Rnyi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California, 1961.

[26] Kristian Dysthe, Harald E Krogstad, and Peter Mller. Oceanic rogue waves. *Annu. Rev. Fluid Mech.*, 40:287–310, 2008.

[27] Christian Kharif and Efim Pelinovsky. Physical mechanisms of the rogue wave phenomenon. *European Journal of Mechanics-B/Fluids*, 22(6):603–634, 2003.

[28] Simon Birkholz, Carsten Bre, Ayhan Demircan, and Gnter Steinmeyer. Predictability of rogue events. *Physical Review Letters*, 114(21):213901, 2015.

[29] L Cavaleri, A Benetazzo, F Barbariol, J-R Bidlot, and PAEM Janssen. The Draupner event: the large wave and the emerging view. *Bulletin of the American Meteorological Society*, 98(4):729–735, 2017.

[30] T Brooke Benjamin and JE Feir. The disintegration of wave trains on deep water Part 1. Theory. *Journal of Fluid Mechanics*, 27(03):417–430, 1967.

[31] A Shabat and V Zakharov. Exact theory of two-dimensional self-focusing and one-dimensional self-modulation of waves in nonlinear media. *Soviet physics JETP*, 34(1):62, 1972.

[32] Marios Christou and Kevin Ewans. Field measurements of rogue water waves. *Journal of Physical Oceanography*, 44(9):2317–2335, 2014.

[33] Wenting Xiao, Yuming Liu, Guangyu Wu, and Dick KP Yue. Rogue wave occurrence and dynamics by direct simulations of nonlinear wave-field evolution. *Journal of Fluid Mechanics*, 720:357–392, 2013.

[34] AV Slunyaev and AV Kokorina. Soliton groups as the reason for extreme statistics of unidirectional sea waves. *Journal of Ocean Engineering and Marine Energy*, 3(4):395–408, 2017.

[35] Will Cousins and Themistoklis P Sapsis. Reduced-order precursors of rare events in unidirectional nonlinear water waves. *Journal of Fluid Mechanics*, 790:368–388, 2016.

[36] Nail Akhmediev, Jose M Soto-Crespo, and Adrian Ankiewicz. Extreme waves that appear from nowhere: on the nature of rogue waves. *Physics Letters A*, 373(25):2137–2145, 2009.

[37] Christopher Chabalko, Ayan Moitra, and Balakumar Balachandran. Rogue waves: new forms enabled by GPU computing. *Physics Letters A*, 378(32): 2377–2381, 2014.

[38] A Moitra, C Chabalko, and B Balachandran. Extreme wave solutions: Parametric studies and wavelet analysis. *International Journal of Non-Linear Mechanics*, 83:2377–2381, 2014.

[39] Jose M Soto-Crespo, N Devine, and N Akhmediev. Integrable turbulence and rogue waves: breathers or solitons? *Physical Review Letters*, 116(10):103901, 2016.

[40] Alexey Slunyaev, Gnther F Clauss, Marco Klein, and Miguel Onorato. Simulations and experiments of short intense envelope solitons of surface water waves. *Physics of Fluids*, 25(6):067105, 2013.

[41] Yu-Hao Sun. Soliton synchronization in the focusing nonlinear Schrdinger equation. *Physical Review E*, 93(5):052222, 2016.

[42] AV Slunyaev and EN Pelinovsky. Role of multiple soliton interactions in the generation of rogue waves: the modified Kortewegde Vries framework. *Physical Review Letters*, 117(21):214501, 2016.

[43] Atiqur Chowdury, DJ Kedziora, Adrian Ankiewicz, and Nail Akhmediev. Soliton solutions of an integrable nonlinear Schrdinger equation with quintic terms. *Physical Review E*, 90(3):032922, 2014.

[44] Nail Akhmediev, Adrian Ankiewicz, and Jose M Soto-Crespo. Rogue waves and rational solutions of the nonlinear Schrdinger equation. *Physical Review E*, 80(2):026601, 2009.

[45] AL Latifah and E van Groesen. Coherence and predictability of extreme events in irregular waves. *Nonlinear processes in geophysics*, 19(2):199–213, 2012.

[46] A Slunyaev. Freak wave events and the wave phase coherence. *The European Physical Journal Special Topics*, 185(1):67–80, 2010.

[47] Michael S Longuet-Higgins. The statistical analysis of a random, moving surface. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 249(966):321–387, 1957.

[48] Odin Gramstad and Karsten Trulsen. Influence of crest and group length on the occurrence of freak waves. *Journal of Fluid Mechanics*, 582:463–472, 2007.

[49] Didier Clamond, Marc Francius, John Grue, and Christian Kharif. Long time interaction of envelope solitons and freak wave formations. *European Journal of Mechanics-B/Fluids*, 25(5):536–553, 2006.

[50] Alfred R Osborne, Miguel Onorato, and Marina Serio. The nonlinear dynamics of rogue waves and holes in deep-water gravity wave trains. *Physics Letters A*, 275(5):386–393, 2000.

[51] Alexey Slunyaev. Nonlinear analysis and simulations of measured freak wave time series. *European Journal of Mechanics-B/Fluids*, 25(5):621–635, 2006.

[52] Walter Edwin Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of applied mathematics*, 9(1):17–29, 1951.

[53] John P. Boyd. *Chebyshev and Fourier spectral methods*. Courier Corporation, 2001.

[54] Stphane Randoux, Pierre Suret, and Gennady El. Inverse scattering transform analysis of rogue waves using local periodization procedure. *Scientific Reports*, 6:29238, July 2016. ISSN 2045-2322. doi: 10.1038/srep29238.

[55] Simon Birkholz, Carsten Bre, Ivan Veseli, Ayhan Demircan, and Gnter Steinmeyer. Ocean rogue waves and their phase space dynamics in the limit of a linear interference model. *Scientific Reports*, 6:35207, October 2016. ISSN 2045-2322. doi: 10.1038/srep35207.

[56] TAA Adcock, PH Taylor, S Yan, QW Ma, and PAEM Janssen. Did the Draupner wave occur in a crossing sea? In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 467, pages 3004–3021. The Royal Society, 2011.

[57] Luigi Cavaleri, Francesco Barbariol, Alvise Benetazzo, Luciana Bertotti, Jean-Raymond Bidlot, Peter Janssen, and Nils Wedi. The Draupner wave: A fresh look and the emerging view. *Journal of Geophysical Research: Oceans*, 121 (8):6061–6075, 2016.

[58] AL Islas and CM Schober. Predicting rogue waves in random oceanic sea states. *Physics of Fluids*, 17(3):031701, 2005.

[59] Jianke Yang. *Nonlinear waves in integrable and nonintegrable systems*. SIAM, 2010.

[60] Kevin P. Murphy. *Machine learning: a probabilistic perspective*. MIT Press, 2012.

[61] Christopher M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[62] Tom M. Mitchell. *Machine Learning*. McGraw-Hill Series in Computer Science. McGraw-Hill Education, 1 edition, 1997. ISBN 978-0-07-042807-2.

[63] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv preprint arXiv:1602.07360*, 2016.

[64] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv:1409.0473 [cs, stat]*, September 2014.

[65] Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115, 2017.

[66] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[67] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, December 2014.

[68] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986.

[69] Alan Hastings, Carole L. Hom, Stephen Ellner, Peter Turchin, and H. Charles J. Godfray. Chaos in ecology: is mother nature a strange attractor? *Annual Review of Ecology and Systematics*, 24(1):1–33, 1993.

[70] Rui Wang and Balakumar Balachandran. Extreme wave formation in unidirectional sea due to stochastic wave phase dynamics. *Physics Letters A*, 382 (28):1864–1872, July 2018. ISSN 0375-9601.

[71] David A. Hsieh. Chaos and nonlinear dynamics: application to financial markets. *The Journal of Finance*, 46(5):1839–1877, 1991.

[72] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, April 2016. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1517384113.

[73] Steven L. Brunton, Bingni W. Brunton, Joshua L. Proctor, Eurika Kaiser, and J. Nathan Kutz. Chaos as an intermittently forced linear system. *Nature Communications*, 8(1):19, May 2017. ISSN 2041-1723.

[74] Jaideep Pathak, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach. *Physical Review Letters*, 120(2):024102, January 2018.

[75] Nikolai A. Kudryashov. Exact solutions of the generalized Kuramoto-Sivashinsky equation. *Physics Letters A*, 147(5-6):287–291, 1990.

[76] Pantelis R. Vlachas, Wonmin Byeon, Zhong Y. Wan, Themistoklis P. Sapsis, and Petros Koumoutsakos. Data-Driven Forecasting of High-Dimensional Chaotic Systems with Long-Short Term Memory Networks. *arXiv:1802.07486 [nlin, physics:physics]*, February 2018.

[77] Sepp Hochreiter and Jrgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[78] A. Karimi and Mark R. Paul. Extensive chaos in the Lorenz-96 model. *Chaos: An interdisciplinary journal of nonlinear science*, 20(4):043105, 2010.

[79] E. N. Lorenz. Predictability: a problem partly solved. In: Proc. Seminar on Predictability, Volume 1. European Centre for Medium-Range Weather Forecast, Shinfield Park, Reading, Berkshire, United Kingdom. 1996.

[80] Edward Ott, Brian R. Hunt, Istvan Szunyogh, Aleksey V. Zimin, Eric J. Kostelich, Matteo Corazza, Eugenia Kalnay, D. J. Patil, and James A. Yorke. A local ensemble Kalman filter for atmospheric data assimilation. *Tellus A*, 56(5):415–428, 2004.

[81] D. J. Patil, Brian R. Hunt, Eugenia Kalnay, James A. Yorke, and Edward Ott. Local low dimensionality of atmospheric dynamics. *Physical Review Letters*, 86(26):5878, 2001.

[82] Eugenia Kalnay. *Atmospheric modeling, data assimilation and predictability*. Cambridge University Press, 2003.

[83] Zoltan Toth and Eugenia Kalnay. Ensemble forecasting at NCEP and the breeding method. *Monthly Weather Review*, 125(12):3297–3319, 1997.

[84] Julien Clinton Sprott and Julien C. Sprott. *Chaos and time-series analysis*, volume 69 of *Physics*. Oxford University Press, 2003. ISBN 0-19-850840-9.

[85] Divakar Viswanath. Lyapunov Exponents from Random Fibonacci Sequences to the LorenzEquations. Technical report, Cornell University, 1998.

[86] Luca Dieci, Michael S. Jolly, and Erik S. Van Vleck. Numerical Techniques for Approximating Lyapunov Exponents and Their Implementation. *Journal of Computational and Nonlinear Dynamics*, 6(1):011003, 2011. ISSN 15551423.

[87] P Manneville. Macroscopic Modelling of Turbulent Flows. In *Proceedings of a Workshop held at INRIA*, volume Lecture Notes in Physics, pages 319–326. (Springer-Verlag, Berlin, Sophia-Antipolis, France, 1985.

[88] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv:1609.04747 [cs]*, September 2016.

[89] James Douglas Hamilton. *Time series analysis*, volume 2. Princeton university press Princeton, NJ, 1994.

[90] Solomon Kullback and Richard A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[91] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[92] Alex Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*. Studies in Computational Intelligence. Springer-Verlag, Berlin Heidelberg, 2012. ISBN 978-3-642-24796-5.

[93] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[94] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[95] Kyunghyun Cho, Bart Van Merrinboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[96] Trieu H. Trinh, Andrew M. Dai, Minh-Thang Luong, and Quoc V. Le. Learning Longer-term Dependencies in RNNs with Auxiliary Losses. *arXiv:1803.00144 [cs, stat]*, February 2018.

[97] Julian Georg Zilly, Rupesh Kumar Srivastava, Jan Koutnk, and Jrgen Schmidhuber. Recurrent Highway Networks. *arXiv:1607.03474 [cs]*, July 2016.

[98] Felix A. Gers, Nicol N. Schraudolph, and Jrgen Schmidhuber. Learning precise timing with LSTM recurrent networks. *Journal of machine learning research*, 3(Aug):115–143, 2002.

[99] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective Approaches to Attention-based Neural Machine Translation. *arXiv:1508.04025 [cs]*, August 2015.

[100] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.

[101] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[102] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. pages 770–778, 2016.

[103] Rupesh Kumar Srivastava, Klaus Greff, and Jrgen Schmidhuber. Highway Networks. *arXiv:1505.00387 [cs]*, May 2015.

[104] I. Aleksandr and Akovlevich Khinchin. *Mathematical foundations of statistical mechanics*. Courier Corporation, 1949.

[105] Georges Lematre. Evolution of the expanding universe. *Proceedings of the National Academy of Sciences of the United States of America*, 20(1):12, 1934.

[106] Stephen W. Hawking and George Francis Rayner Ellis. *The large scale structure of space-time*, volume 1. Cambridge University Press, 1973.

[107] Steven M. Reppert and David R. Weaver. Coordination of circadian timing in mammals. *Nature*, 418(6901):935, 2002.

[108] Otto E. Rssler. An equation for continuous chaos. *Physics Letters A*, 57(5): 397–398, 1976.

[109] Georg Duffing. *Erzwungene Schwingungen bei vernderlicher Eigenfrequenz und ihre technische Bedeutung*. Number 41-42. F. Vieweg & sohn, 1918.

[110] Ivana Kovacic and Michael J. Brennan. *The Duffing equation: nonlinear oscillators and their behaviour*. John Wiley & Sons, 2011.

[111] Philip Holmes. A nonlinear oscillator with a strange attractor. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 292(1394):419–448, 1979.

[112] F. C. Moon and Philip J. Holmes. A magnetoelastic strange attractor. *Journal of Sound and Vibration*, 65(2):275–296, 1979.

[113] Aly-Khan Kassam and Lloyd N. Trefethen. Fourth-order time-stepping for stiff PDEs. *SIAM Journal on Scientific Computing*, 26(4):1214–1233, 2005.

[114] Leslie M. Smith and Fabian Waleffe. Generation of slow large scales in forced rotating stratified turbulence. *Journal of Fluid Mechanics*, 451:145–168, 2002.

[115] Leslie M. Smith and Fabian Waleffe. Transfer of energy to two-dimensional large scales in forced, rotating three-dimensional turbulence. *Physics of Fluids*, 11(6):1608–1622, 1999.

[116] Paul A. Milewski and Esteban G. Tabak. A pseudospectral procedure for the solution of nonlinear wave equations with examples from free-surface flows. *SIAM journal on scientific computing*, 21(3):1102–1114, 1999.

[117] Yvon Maday, Anthony T. Patera, and Einar M. Rnquist. An operator-integration-factor splitting method for time-dependent problems: application to incompressible fluid flow. *Journal of Scientific Computing*, 5(4):263–292, 1990.

[118] Steven M. Cox and Paul C. Matthews. Exponential time differencing for stiff systems. *Journal of Computational Physics*, 176(2):430–455, 2002.

[119] Richard A. Friesner, Laurette S. Tuckerman, Bright C. Dornblaser, and Thomas V. Russo. A method for exponential propagation of large systems of stiff nonlinear differential equations. *Journal of Scientific Computing*, 4(4): 327–354, 1989.

[120] Nicholas J. Higham. *Accuracy and stability of numerical algorithms*, volume 80. Siam, 2002.