# BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

## FACULTY OF MECHANICAL ENGINEERING

FAKULTA STROJNÍHO INŽENÝRSTVÍ

## INSTITUTE OF MATHEMATICS

ÚSTAV MATEMATIKY

# SYNCHRONIZATION OF CHAOTIC DYNAMICAL SYSTEMS

SYNCHRONIZACE CHAOTICKÝCH DYNAMICKÝCH SYSTÉMŮ

MASTER'S THESIS

DIPLOMOVÁ PRÁCE

AUTHOR
AUTOR PRÁCE
Bc. Ondřej Borkovec

SUPERVISOR
VEDOUCÍ PRÁCE
doc. Ing. Petr Tomášek, Ph.D.

BRNO 2019

# Zadání diplomové práce

| | |
|---|---|
| Ústav: | Ústav matematiky |
| Student: | **Bc. Ondřej Borkovec** |
| Studijní program: | Aplikované vědy v inženýrství |
| Studijní obor: | Matematické inženýrství |
| Vedoucí práce: | **doc. Ing. Petr Tomášek, Ph.D.** |
| Akademický rok: | 2018/19 |

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

## Synchronizace chaotických dynamických systémů

**Stručná charakteristika problematiky úkolu:**

Některé dynamické systémy vykazují "zvláštní" chování, pro které se ujal název (deterministický) chaos. Ten se vyznačuje tím, že "většina" trajektorií systému je přitahována množinou se složitou geometrickou strukturou, tzv. podivným atraktorem. Základní vlastností chaotických systémů je vysoká citlivost na poruchy v počátečních podmínkách, přesněji, ukazuje se, že (alespoň krátce) je rychlost separace dvou trajektorií začínajících blízko sebe exponenciální. Tento jev může svádět k závěru, že synchronizace dvou (ať už stejných nebo rozdílných) chaotických systémů bude obtížná, ne–li nemožná. Opak je však pravdou, již v roce 1990 L. M. Pecora a T. L. Carroll navrhli poměrně jednoduchý způsob spárování dvou systémů tak, aby chybový signál (ten je dán rozdílem mezi stavy uvažovaných systémů) konvergoval k nule (pro libovolnou počáteční podmínku). Toho pak lze využít v mnoha aplikacích, např. při znemožnění odposlechu akustických signálů.

**Cíle diplomové práce:**

Rešeršní část: Studium teorie nelineárních dynamických systémů se zaměřením na chaotické systémy. Synchronizační metody pro dva chaotické systémy.

Praktická část: Analýza synchronizace konkrétních chaotických systémů. Vytvoření algoritmu v prostředí MATLAB a následné testování teoretických výsledků.

**Seznam doporučené literatury:**

PECORA, L.M. and CARROLL, T. L. Synchronization of chaotic systems, Chaos 25 (2015), 13 pp.

PIKOVSKY, A., ROSEMBLUM, M. and KURTHS, J. Synchronization: A Universal Concept in Nonlinear Sciences, Cambridge University Press, 2001.

STROGATZ, S. H. Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering, 2nd ed., Westview Press, 2014.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2018/19

V Brně, dne

L. S.

| | |
|---|---|
| prof. RNDr. Josef Šlapal, CSc.<br>ředitel ústavu | doc. Ing. Jaroslav Katolický, Ph.D.<br>děkan fakulty |

**Abstract**
The master's thesis deals with the basic notion of chaotic dynamical systems with the special focus on their synchronization. The process of synchronization is then applied using two different methods: complete synchronization on two Lorenz systems and negative-feedback method on two Rössler systems. The possible application of synchronization of chaotic systems on the field of private communication is investigated and it is complemented by the algorithms in MATLAB.

**keywords**
chaotic dynamical systems, Lyapunov exponents, complete synchronization, negative-feedback synchronization, private communication, signal encryption

**Abstrakt**
Diplomová práce pojednává o chaotických dynamických systémech se zvláštním zaměřením na jejich synchronizaci. Proces synchronizace je aplikován použitím dvou různých metod, a to - metodou úplné synchronizace na dva Lorenzovy systémy a metodou negativní zpětné vazby na dva Rösslerovy systémy. Dále je prozkoumána možná aplikace synchronizace chaotických systémů v oblasti soukromé komunikace, která je doplněná algoritmy v prostředí MATLAB.

**klíčová slova**
chaotické dynamické systémy, Ljapunovovy exponenty, úplná synchronizace, synchronizace negativní zpětnou vazbou, šifrování signálu

BORKOVEC, Ondřej.: *Synchornization of chaotic dynamical systems*, Brno University of Technology, Faculty of Mechanical Engineering, 2019. 57 pages. Supervised by doc. Ing. Petr Tomášek, Ph.D.

I declare that I have written my master's thesis titled *Synchornization of chaotic dynamical systems* independently, under the direction of my supervisor doc. Ing. Petr Tomášek, Ph.D. and with the use of literature and other sources of information which are all cited in the thesis and detailed in the list of literature at the end of the thesis.

Bc. Ondřej Borkovec

I would like to thank to my supervisor doc. Ing. Petr Tomášek, Ph.D. for help, great patience throughout the year and the advice he gave me, especially for the the practical part of the thesis.

Ondřej Borkovec

# Contents

# Introduction

Dynamics is a part of physics that deals with the change of objects evolving in time. First work in this field began in 16th century by the Isaac Newton, who defined laws of motion and invented the calculus along with the differential equations. In general, dynamical system is a differential equation or a system of them, describing a motion of a point in a geometrical space. It is prescribing rules under which the system behaves in time.

Nowadays we use the dynamical systems for modelling the phenomena in various fields including physics, biology, chemistry or economics. In the nature, most of the systems are non-linear and they are very difficult or nearly impossible to solve analytically. It is then no surprise that the era of deeper investigations has raised with the progress in computing technology. On top of that, it has been discovered that some systems possessed an unpredictable chaotic behaviour, when the initial data for the system have been changed.

Lorenz, in 1963, was the first who presented a model that exhibited this behaviour along with some possible explanations and also showed the existence of a strange attractor - a complicated set of solutions of the model with unexpected properties. This discovery then became quickly well known. Among other things, its popularity raised also because of the characteristic shape of the strange attractor that looks like a wings of a butterfly. Since then, the study of chaos theory attracted many scientists to explore and brought interesting results.

The idea of synchronization of the dynamical systems does not have to be strictly bounded to chaos. In fact many mathematicians believed that due to the nature of the system, it internally defies the synchronization. In the 90's, multiple mathematicians in different parts of the world independently published papers, where they showed that two chaotic systems starting from different initial conditions can be synchronized. Their methods were often using different approaches. Further investigations on this topic continue even nowadays. Many methods of the synchronization of chaotic dynamical systems and the potential usage of it have been found. However, in the chaos theory there are still phenomena that have not been fully understood and the explanations for their occurrence are yet to be discovered.

The goal of the first part of this work is to bring a brief outline for the theory of dynamical systems and discuss their chaotic behaviour, together with several examples. The goal of the second part is to investigate the methods of synchronization for two specific chaotic dynamical systems both theoretically and numerically. Moreover, it is desired to create an algorithm in MATLAB environment for that synchronization and show some possible application of this phenomena on the field of private communications. The work is organised as follows:

Chapter 1 presents an overview of basic notions and properties related to dynamical systems and their qualitative description.

Chapter 2 deals with the introduction to chaos theory and the methods of measurement of chaos such as the notion of the Lyapunov exponents. Then a concrete examples of chaotic dynamical systems with their attractors are showed, particularly and the Lorenz system, the Rössler system.

Chapter 3 is devoted to the basic overview of methods of synchronization of chaotic dynamical system. The acquired knowledge in the chapter is then applied to the synchronization of two Lorenz systems and two Rössler systems via presented methods.

Finally, chapter 4 discusses possible applications of synchronization of chaotic dynamical systems. The specific type of encryption of data with the chaotic signal and the decryption using the synchronization is presented. Then the theoretical results are demonstrated on examples of encryption and decryption on two types of data, e. g. digital binary valued bit-streams and analog waveform signal such as an image.

# 1 Dynamical systems

The study of dynamical systems can be divided into two groups with respect to the perception of time. If it is discrete, therefore it attains only integer values, we call it discrete dynamical system. This can be referred to the change of cattle population in the herd, where each month we count the pieces and we observe how the value changes over these periods. If we measure the time continuously, meaning that $t \in \mathbb{R}$, we call the system a continuous dynamical system.

Our work will be focused on continuous autonomous dynamical systems, which means that they are time-invariant. These systems do not explicitly depend on time and they have a property that the current state of the system is fully determined by its previous state. From a mathematical point of view, our subject of work will be a dynamical system of dimension $n$ given by the ordinary differential equation

$$\begin{aligned} \dot{\boldsymbol{x}} &= f(\boldsymbol{x}), \\ \boldsymbol{x}(0) &= \boldsymbol{x}_0, \end{aligned} \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable function not depending on $t$.

Our aim is now to define more precisely the theory of dynamical systems, which will support the conclusions later in this work. This chapter is mainly consisted from the definitions and theorems in [1], [2], [3], [4].

## 1.1 Rigorous definition of a dynamical system

In this section we define the most important concepts, which we discuss in the next chapters.

The definition comes intuitively from our interpretation of a point $\boldsymbol{x}$ moving in time. A map $\Phi_t : \mathbb{R}^n \to \mathbb{R}^n$ takes a point $\boldsymbol{x}$ into $\boldsymbol{x}_t$ and is defined for all $t$, it is also reasonable to expect that $\Phi_t$ has $\Phi_{-t}$ as its inverse. Then, $\Phi_0$ should be the identity function and $\Phi_t(\Phi_s(\boldsymbol{x})) = \Phi_{t+s}(\boldsymbol{x})$ is also a natural condition. We formalize all of this in the following definition

**Definition 1.1.** Dynamical system on $\Omega$ is a continuously differentiable function

$$\Phi : \mathbb{R} \times \Omega \to \Omega,$$

where $\Omega$ is an open subset of $\mathbb{R}^n$ called *phase space* and a function $\Phi$ satisfies

1. $\Phi_0(\boldsymbol{x}) = \boldsymbol{x}$ for all $\boldsymbol{x} \in \Omega$,
2. $\Phi_t \circ \Phi_s(\boldsymbol{x}) = \Phi_{t+s}(\boldsymbol{x})$ for all $t, s \in \mathbb{R}$ and for all $\boldsymbol{x} \in \Omega$.

If the condition of continuous differentiability holds, we call the system *smooth*.

As we see in the next section, this condition guarantees us the existence and uniqueness of the initial value problem for most of the dynamical systems.

## 1.2 The Existence and Uniqueness theorem

Let us consider the system of differential equations

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}),$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$. Solution of this system is a function $\boldsymbol{x} : J \to \mathbb{R}^n$ defined on some interval $J \subset \mathbb{R}$, such that, for all $t \in J$,

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t)).$$

If we look at the system geometrically, then $\boldsymbol{x}(t)$ is a curve in $\mathbb{R}^n$ with a tangent vector $\dot{\boldsymbol{x}}(t)$ that exists for all $t \in J$ and equals $f(\boldsymbol{x}(t))$. We think of this vector as being based at $\boldsymbol{x}(t)$ so the map $f$ defines a vector field on $\mathbb{R}^n$. We add an initial condition $\boldsymbol{x} : J \to \mathbb{R}^n$ which is in the form $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$, where $t_0 \in J$ and $\boldsymbol{x}_0 \in \mathbb{R}^n$. Now the main task is to find the solution of any initial value problem, but unfortunately some non-linear differential equations may have no solutions that satisfy certain initial condition.

Consider first-order differential equation

$$\dot{x} = \begin{cases} 1 & \text{if } x < 0, \\ -1 & \text{if } x \geq 0. \end{cases}$$

The vector field in the phase portrait on $\mathbb{R}$ points to the left when $x \geq 0$ and to the right of $x < 0$. Consequently, there is no such solution that would satisfy the initial condition $x(0) = 0$ due to discontinuity of the vector field at 0. Another problem arises when there are more solutions for one initial condition. The differential equation

$$\dot{u} = 3u^{\frac{2}{3}}$$

for an initial condition $u(0) = 0$ has as a solution a zero function $u : \mathbb{R} \to \mathbb{R}$ given by $u(t) \equiv 0$, but also $u_0(t) = t^3$ is a solution satisfying this initial condition. These problems are caused because the function $u(t)$ is not differentiable at point $u(0) = 0$.

Therefore we must apply a condition of continuous differentiability on the function $f(\boldsymbol{x})$ so we would guarantee both existence and solution. Following theorem summarizes knowledge we acquired so far.

**Theorem 1.1** (Existence and Uniqueness)**.** *Consider initial value problem*

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}), \boldsymbol{x}(t_0) = \boldsymbol{x}_0,$$

*where $\boldsymbol{x}_0 \in \mathbb{R}^n$. Suppose that $f : \mathbb{R}^n \to \mathbb{R}^n$ is $C^1$. Then there exists a solution of this initial value problem and it is the only such solution. More precisely, there exists an $a > 0$ and a unique solution*

$$\boldsymbol{x} : (t_0 - a, t_0 + a) \to \mathbb{R}^n,$$

*of this differential equation satisfying the initial condition $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$.*

*Proof.* The proof of this theorem can be found in [2]. □

## 1.3 Geometrical point of view on dynamical systems

We now look at the system from a geometrical view point.

**Definition 1.2.** A *phase portrait* of a system of differential equations such as (1.1) with $\boldsymbol{x} \in \mathbb{R}^n$ is the set of all solution curves of (1.1) in the phase space $\mathbb{R}^n$.

**Definition 1.3.** Let the initial point $\boldsymbol{x}_0$ be fixed and let $J = J(\boldsymbol{x}_0)$, then the mapping $\Phi(\cdot, \boldsymbol{x}_0) : J \to \Omega$ defines a solution curve or *trajectory* of the system (1.1) through the point $\boldsymbol{x}_0 \in \Omega$.

Geometrically, the dynamical system describes the motion of the points in phase space along the solution curves defined by the system of differential equations. From algebraic point of view, the system of differential equations represents the vector field. It dictates the velocity vector $\dot{\boldsymbol{x}}$ at each $\boldsymbol{x}$ and the vector of directions is called the flow. We give more precise definition.

**Definition 1.4.** Let $\Omega \in \mathbb{R}^n$ and $f \in C^1(\Omega)$. Let $\Phi(t, \boldsymbol{x}_0)$ be the solution of (1.1) defined on its maximal interval $J(\boldsymbol{x}_0)$, $x_0 \in \Omega$. Then for $t \in J(\boldsymbol{x}_0)$, the family of evolution operators $\Phi_t$ defined by

$$\Phi_t(\boldsymbol{x}_0) = \Phi(t, \boldsymbol{x}_0),$$

is called the *flow* of the system (1.1).

The fact that initial value problem (1.1) has a solution defined for each $\boldsymbol{x}_0 \in \Omega$ on the maximal interval $J = (\alpha, \beta)$ is a conclusion of a theorem that can be found in [1]. It is important to say that the theorem 1.1 has an important corollary: solution curves in the phase space never intersect each other. The visual appearance of the phase portrait is controlled by the fixed points which we also call the equilibria.

**Definition 1.5** (Fixed point)**.** A point $\boldsymbol{x}^* \in \Omega$ is called a *fixed point* (equilibrium point, critical point) of the system (1.1) if

$$f(\boldsymbol{x}^*) = \mathbf{0}.$$

Moreover, for any trajectory starting in $\boldsymbol{x}^*$, i. e. $\boldsymbol{x}(0) = \boldsymbol{x}^*$, is $\boldsymbol{x}(t) = \Phi_t(\boldsymbol{x}^*) \equiv \boldsymbol{x}^*$ for any $t \in \mathbb{R}$.

In general, trajectories of the solution $\boldsymbol{x}(t)$ can be divided into 3 main categories:

1. Fixed point - the solution $\boldsymbol{x}(t)$ is constant, i. e. trajectory stays in the fixed point for all time.
2. Cycle, periodic orbit - the solution $\boldsymbol{x}(t)$ is periodic, i. e. the trajectory forms a closed curve and stays on this curve for all time.
3. Open curve - the trajectory is an injective map never intersecting itself.

The equilibria then can be distinguished in terms of stability, which is discussed later.

## 1.4 Linear system

A special case of (1.1) is when it is linear, i. e. the function $f : \mathbb{R}^n \to \mathbb{R}^n$ consists of only linear terms. Then we can express the dynamical system in the form

$$\begin{aligned} \dot{\boldsymbol{x}} &= A\boldsymbol{x} \\ \boldsymbol{x}(0) &= \boldsymbol{x}_0, \end{aligned} \tag{1.2}$$

where $\boldsymbol{x}$ is a a vector in $\mathbb{R}^n$ and $A$ is $n \times n$ matrix. The name linear system comes from the sense, that if $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ are solutions, then also their linear combination $c_1\boldsymbol{x_1} + c_2\boldsymbol{x_2}$ is a solution. Also $\dot{\boldsymbol{x}} = \mathbf{0}$ when $\boldsymbol{x} = \mathbf{0}$, therefore $\boldsymbol{x}^* = \mathbf{0}$ is always a fixed point for linear system for any regular matrix $A$.

It can be shown that the following theorem holds:

**Theorem 1.2** (The Fundamental theorem for linear systems)**.** *Let $A$ be $n \times n$ matrix. Then for given $\boldsymbol{x}_0 \in \mathbb{R}^n$ the initial value problem* (1.2) *has a unique solution for all $t \in \mathbb{R}$ given by*

$$\boldsymbol{x}(t) = e^{At}\boldsymbol{x}_0, \tag{1.3}$$

*where the exponential of $A$ is defined as*

$$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!}.$$

*Proof.* See [1]. □

This theorem gives us an elegant way how to to deal with the solution of the linear system. The version of the theorem for non-autonomous systems, where $A = A(t)$ and its proof can be found in [2].

In theory of dynamical systems, we are often interested in qualitative analysis. We restrict our attention to the planar linear systems and discuss their behaviour. The analysis of $n$-dimensional linear systems is thoroughly studied in [1] or [2].

**Classification of fixed points in the plane** The planar linear system is defined as follows:

$$\dot{\boldsymbol{x}} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \boldsymbol{x}. \tag{1.4}$$

The planar systems are then divided into two main categories: hyperbolic and non-hyperbolic.

**Definition 1.6.** A planar linear system (1.4) is called *hyperbolic* if no eigenvalues of matrix $A$ have 0 real part. In the opposite case it is called *non-hyperbolic.*

In the next part, the qualitative behaviour of planar systems will be discussed.

**Definition 1.7.** Suppose that matrix $A$ has two real eigenvalues $\lambda_1 < \lambda_2$. Then the fixed point is called:
(i) *Saddle* if $\lambda_1 < 0 < \lambda_2$,
(ii) *Sink* if $\lambda_1 < \lambda_2 < 0$,
(iii) *Source* if $0 < \lambda_1 < \lambda_2$.
Furthermore, suppose that matrix $A$ has two complex eigenvalues $\lambda_{1,2} = a \pm ib$ Then the fixed point is called:
(i) *Center* if $a = 0, b \neq 0$,
(ii) *Spiral sink* if $a < 0, b \neq 0$,
(iii) *Spiral source* if $a > 0, b \neq 0$.

If one or both eigenvalues of $A$ are zero, i. e. $\det A = 0$, the fixed point is called *degenerate.* These points appear in non-hyperbolic systems. In literature the sinks, sources sometimes are called stable and unstable nodes and their spiral versions are stable and unstable foci respectively. There is an elegant method on categorizing the systems by its behaviours.

From the characteristic equation

$$\lambda^2 - (a+d)\lambda + (ad - bc) = 0,$$

it is easy to see that the coefficient by the linear term is the trace of A denoted tr$A$ and the latter term is the determinant det$A$. Then the eigenvalues satisfy

$$\lambda^2 - (\mathrm{tr}A)\lambda + \mathrm{det}A = 0,$$

and are given by

$$\lambda_{1,2} = \frac{1}{2}\left(\mathrm{tr}A \pm \sqrt{(\mathrm{tr}A)^2 - 4\,\mathrm{det}A}\right).$$

The trace and the determinant tells us everything about the geometry of solutions of the planar system. We can then visualize the behaviour of the system in *trace-determinant plane* (fig. 1). In the picture, the matrix with trace Tr$A$ and determinant Det$A$ corresponds to the point with coordinates $(\mathrm{tr}A, \mathrm{det}A)$. The location of the point in this plane then determines the geometry of the phase portrait. The regions are bounded by the two lines and a parabola that corresponds to the case when the term in square root is equal to zero.

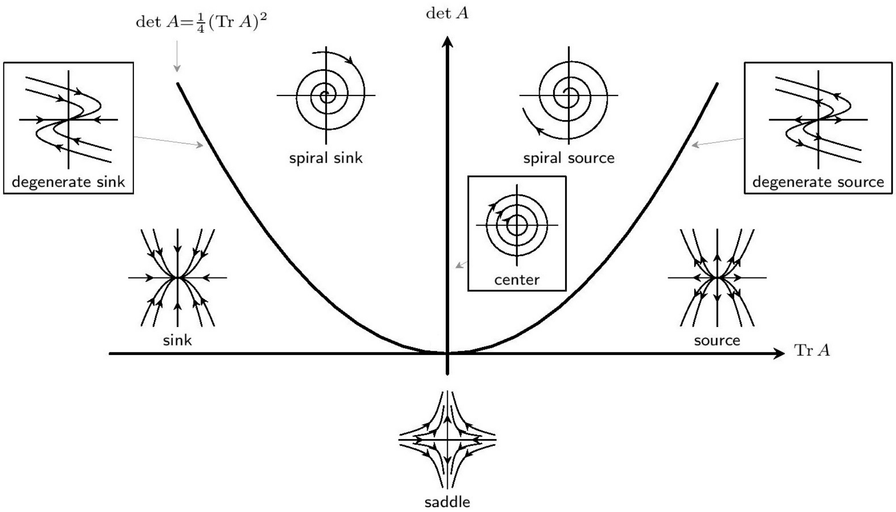



Figure 1: trace-determinant plane [17]

## 1.5 Non-linear systems

In previous section we summarized basics of the linear systems. If the function $f(\boldsymbol{x})$ of (1.1) has besides the linear part one or more non-linear terms, the behaviour of the system may change rapidly. Unfortunately, the analytical tools for the non-linear systems are limited. For solving most of the non-linear systems, we have to use the numerical methods. It is then no surprise that the biggest progress in studying these systems came with the invention of computers. In following section we show one of the techniques on how to analyse these systems.

**Linearisation of non-linear systems** Suppose we have system (1.1). A good place to start in analysing this non-linear system is to determine its equilibrium points and then describe the behaviour in their neighbourhood.

Consider the system (1.1) and suppose that $\boldsymbol{x}^*$ is its fixed point, i. e. $f(\boldsymbol{x}^*) = \mathbf{0}$. Let us a denote small disturbance around the fixed point as

$$\boldsymbol{u} = \boldsymbol{x} - \boldsymbol{x}^*.$$

We are now interested whether the disturbance with time decays or grows. We examine the derivative:

$$\begin{aligned}
\dot{\boldsymbol{u}} &= \dot{\boldsymbol{x}} && \dots \text{ since } \boldsymbol{x}^* \text{ is constant,} \\
&= f(\boldsymbol{x}^* + \boldsymbol{u}) && \dots \text{ by substitution,} \\
&= f(\boldsymbol{x}^*) + \boldsymbol{u}\,\frac{\partial f(\boldsymbol{x}^*)}{\partial \boldsymbol{x}} + O(\boldsymbol{u}^2) && \dots \text{ Tayor series expansion,} \\
&= \boldsymbol{u}\,\frac{\partial f(\boldsymbol{x}^*)}{\partial \boldsymbol{x}} + O(\boldsymbol{u}^2) && \dots \text{ since } f(\boldsymbol{x}^*) = \mathbf{0}.
\end{aligned}$$

Note, that the partial derivative in the last term is being evaluated in the fixed point, therefore it is a number and not a function. By $O(\boldsymbol{u}^2)$ we denote quadratic and higher order terms in $\boldsymbol{u}$. If we consider small perturbations, this remainder will be extremely small and we neglect it. Hence the disturbance evolves by the equation

$$\begin{pmatrix} \dot{u}_1 \\ \dot{u}_2 \\ \vdots \\ \dot{u}_n \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1(\boldsymbol{x}^*)}{\partial x_1} & \cdots & \frac{\partial f_1(\boldsymbol{x}^*)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n(\boldsymbol{x}^*)}{\partial x_1} & \cdots & \frac{\partial f_n(\boldsymbol{x}^*)}{\partial x_n} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}. \tag{1.5}$$

The matrix of partial derivatives is a Jacobian and we denote it by $J = Df(\boldsymbol{x}^*)$. Combined with the classification of the fixed points in previous section, the equation (1.5) gives us as a good approximation of the system. It is natural question to ask how much accurate it is. The next theorem gives us a connection between the equilibrium of non-linear system and its linearised version. We first state few preliminaries and then the theorem.

**Definition 1.8.** An equilibrium point $\boldsymbol{x}^*$ of the system (1.1) is called hyperbolic if none of the eigenvalues of the Jacobian matrix $J = Df(\boldsymbol{x}^*)$ has zero real part. Otherwise, the equilibrium point is called non-hyperbolic.

**Definition 1.9.** Let

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}), \tag{1.6}$$
$$\dot{\boldsymbol{x}} = J\boldsymbol{x}, \tag{1.7}$$

be systems of differential equations. The systems are said to be *topologically equivalent* in a neighbourhood of the origin if there is a homeomorphism $H$ mapping the open set $U$ containing the origin onto an open set $V$ containing the origin and maps trajectories of (1.6) in $U$ onto trajectories of (1.7) in $V$ and preserves their orientation by time. Moreover, if the homeomorphism $H$ also preserves the parametrization by time, then the systems (1.6) and (1.7) are said to be *topologically conjugate* in a neighbourhood of the origin.

**Theorem 1.3** (Hartman-Grobman)**.** *Let $\Omega$ be open subset of $\mathbb{R}^n$ containing the origin. Let $f \in C^1(\Omega)$. Suppose that $f(\mathbf{0}) = \mathbf{0}$ and that the matrix $J = Df(\mathbf{0})$ has no eigenvalue with zero real part. Then there exists a homeomorphism $H$ that maps trajectories of* (1.6) *near the origin onto trajectories of* (1.7) *near the origin and preserves the parametrization by time.*

*Proof.* See [1]. □

The corollary of this theorem is that in a close neighbourhood of hyperbolic fixed point (1.6) it qualitatively acts as its linearised version (1.7). For the cases of non-hyperbolic fixed points see [3].

**Closed orbits and limit sets** Unlike their linear counterpart, non-linear systems posses also other type of solution such as closed orbits or periodic solution. A periodic solution occurs for (1.1) if we have a non-equilibrium point $\boldsymbol{x}$ and time $\tau > 0$ for which $\Phi_\tau(\boldsymbol{x}) = \boldsymbol{x}$. It follows that $\Phi_{\tau+t}(\boldsymbol{x}) = \Phi_\tau(\boldsymbol{x})$ for all $t$, so $\Phi_\tau$ is a periodic function.

**Definition 1.10.** A point $\boldsymbol{p} \in \Omega \subset \mathbb{R}^n$ is called an $\omega$-limit point for the solution through $\boldsymbol{x}$ if there is a sequence of times $t_n \to \infty$ such that

$$\lim_{n\to\infty} \Phi_{t_n}(\boldsymbol{x}) = \boldsymbol{p}.$$

Similarly, if there is a sequence of times $t_n \to -\infty$ such that

$$\lim_{n\to\infty} \Phi_{t_n}(\boldsymbol{x}) = \boldsymbol{p},$$

then $p$ is called $\alpha$-limit point. By a limit set we mean a set of the form $\omega(\boldsymbol{x})$ or $\alpha(\boldsymbol{x})$.

Following theorem gives us an important corollary about all the possible limiting behaviours in planar systems.

**Theorem 1.4** (Poincaré-Bendixon)**.** *Suppose that $\Omega$ is a non-empty, closed and bounded limit set of planar system of differential equations that contains no equilibrium point. Then $\Omega$ is closed orbit.*

*Proof.* See [2]. □

The theorem is one of the crucial assertions of non-linear dynamics. It says that the dynamical possibilities in phase plane are very limited: if a trajectory is confined to a closed, bounded region that contains no fixed points, then the trajectory must eventually approach the closed orbit. Nothing more complicated is possible. In the dimension $n > 2$ this theorem is no longer valid and the trajectories may also wander around forever in closed region without settling down into a fixed point or a closed orbit. That allows the existence of strange attractors and chaotic behaviour of dynamical systems, which we discuss in the following sections.

## 1.6 Stability

A study of equilibria plays a central role in ordinary differential equations and their applications. An equilibrium point, however, must satisfy a certain stability criterion to be physically significant. From this point of view, we divide the equilibria into three categories with respect to the sensitivity to initial data perturbation. Intuitively, an equilibrium point is stable, if nearby solutions stay nearby for all future time. We now give more precise definition.

**Definition 1.11.** Suppose $\boldsymbol{x}^* \in \mathbb{R}^n$ is an equilibrium point for (1.1). Then $\boldsymbol{x}^*$ is *stable*, if for every neighbourhood $\mathcal{O}$ of $\boldsymbol{x}^*$ there is a neighbourhood $\mathcal{O}_1$ of $\boldsymbol{x}^*$ in $N$, such that every solution $\boldsymbol{x}(t)$ with $\boldsymbol{x}(\mathbf{0}) = \boldsymbol{x}_0$ in $\mathcal{O}_1$ is defined and remains in $\mathcal{O}$ for all $t > 0$. Moreover, if $\mathcal{O}_1$ can be chosen so that $\boldsymbol{x}^*$ is stable and we have

$$\lim_{t\to\infty} \boldsymbol{x}(t) = \boldsymbol{x}^*,$$

then it is *asymptotically stable*. An equilibrium is *unstable* if it is not stable.

Sources and saddles are examples of unstable equilibria. An example of fixed point that is stable, but not asymptotically stable, is center. Also, $\omega$-limit set is asymptotically stable and $\alpha$-limit set is unstable. The basic tool for stability analysis of equilibrium points is provided by the linearisation method. The stability of a fixed point can be determined from the sign of real parts of eigenvalues $\lambda$ of the Jacobian matrix. The following theorem holds.

**Theorem 1.5.** *Let $J = Df(\boldsymbol{x}^*)$ be the Jacobian matrix for the system* (1.1) *evaluated at the fixed point $\boldsymbol{x}^*$ and let $\lambda_i$ be its eigenvalues.*

*(i) If $Re(\lambda_i) < 0$ for all $\lambda_i$ then a fixed point $\boldsymbol{x}^*$ is asymptotically stable.*
*(ii) If $Re(\lambda_i) > 0$ at least for one $\lambda_i$ then a fixed point $\boldsymbol{x}^*$ is unstable.*
*(iii) If $Re(\lambda_i) = 0$ at least for one $\lambda_i$ then a fixed point non-hyperbolic and its stability cannot be determined by the linearisation method.*

*Proof.* See [1]. □

Unfortunately, many important equilibrium points that arise in applications are non-hyperbolic. There are no universal global techniques on determining the stability except by actually finding all solutions of the system, which is usually difficult if not impossible. Nevertheless, there are some techniques that allow us to determine stability in certain special cases. One of them is determination of stability by Lyapunov theorem.

**Theorem 1.6.** *(Lyapunov Function). Suppose the non-linear system* (1.1) *with an equilibrium point $\boldsymbol{x}^*$, $\boldsymbol{x}^* \in \Omega$, where $\Omega$ is an open subset in $\mathbb{R}^n$. Suppose that there exists a function $V \in C^1(\Omega)$ satisfying $V(\boldsymbol{x}^*) = \mathbf{0}$ and $V(\boldsymbol{x}^*) > 0$ if $\boldsymbol{x} \neq \boldsymbol{x}^*$. Then*

*a) if $\dot{V}(\boldsymbol{x} \leq$ for all $\boldsymbol{x} \in \Omega$, then $\boldsymbol{x}^*$ is stable.*
*b) if $\dot{V}(\boldsymbol{x} < 0$ for all $\boldsymbol{x} \in \Omega\backslash\{\boldsymbol{x}^*\}$, then $\boldsymbol{x}^*$ is asymptotically stable.*
*c) if $\dot{V}(\boldsymbol{x} > 0$ for all $\boldsymbol{x} \in \Omega\backslash\{\boldsymbol{x}^*\}$, then $\boldsymbol{x}^*$ unstable.*

*The function $V$ is called the Lyapunov function and the term $\dot{V}(\boldsymbol{x}) = DV(\boldsymbol{x})f(\boldsymbol{x})$, where $DV = \left(\frac{\partial V}{\partial x_1}, \ldots, \frac{\partial V}{\partial x_n}.\right)$*

*Proof.* See [2]. □

## 1.7 Attractors

Besides equilibrium points and periodic orbits, a dynamical system can have other attractors. Intuitively, by the term attractor we mean a set to which all the trajectories of the dynamical system converge, e. g. stable fixed points and stable limit cycles. So far the mathematicians have not been able to agree on what the exact definition should be. For the purposes of this work we define strange attractor as in [3].

**Definition 1.12.** An *attractor* is a closed set $A$ with the following properties:

- $A$ is an invariant set, i. e. any trajectory $\boldsymbol{x}(t)$ that starts in $A$ stays in $A$.
- $A$ attracts an open set of initial conditions, i. e. there is an open set $U$ containing $A$ s. t. if $\boldsymbol{x}(0) \in U$, then the distance from $\boldsymbol{x}(t)$ to $A$ tends to 0 as $t \to \infty$. That means that $A$ attracts all trajectories that start sufficiently close to it. The largest $U$ is called *basin of attraction* of $A$
- $A$ is minimal, i. e. there is no proper subset of $A$ that satisfies previous two conditions.

There are three types of so-called non-strange attractors, namely fixed point attractor, limit cycle attractor and torus attractor. A particular attractor, which was observed in chaotic systems is called a strange attractor and its definition leads to fractal theory. For the purposes of this work it is enough to say that a *strange attractor* is an attractor that also exhibits a sensitive dependence on initial conditions.

# 2 Chaotic dynamical systems

The theorem 1.4 in previous chapter clearly determined the behaviour of planar dynamical systems. In higher dimension the situation is not that clear. One of the most astonishing observations in this field of study was that there are systems, which are under some conditions extremely sensible towards the change of the initial data. First work on this topic was the formulation of a system done by E. N. Lorenz in 1963. The system was derived from a vastly oversimplified model of atmospheric convection.

This sensitivity makes the system act chaotic, because even the slight change in the initial condition can make the solution follow completely different trajectory. No definition of the term *chaos* has been widely agreed on yet, but in the literature there are some matching concepts which we now state.

**Definition 2.1.** *Chaos* is aperiodic long-term behaviour in a deterministic system, that exhibits sensitive dependence on initial conditions.

Aperiodic long-term behaviour means, that there are which do not settle down to fixed points, or periodic orbits as $t \to \infty$. In this context, deterministic means that the system has no noisy inputs or parameters. Its irregular behaviour follows strictly from the non-linearity, rather from the noisy driving forces. A sensitive dependence on initial conditions means that the trajectories separate separate exponentially fast. Naturally, a tool for measuring the sensitivity is desired. We introduce the notion o Lyapunov exponents.

## 2.1 Lyapunov exponents

The Lyapunov exponent is a measure of the sensitivity to a small changes in initial conditions and it is primarily used for identifying chaos. Suppose $\boldsymbol{x}(t)$ is a point on the solution curve at time t and consider a nearby point $\boldsymbol{x}(t)+d(t)$, where $d$ is tiny separation vector of the initial length $d_0$. We want to compare the distances of these two points at times $t_0$ and $t$, possibly on the logarithmic scale. The sketch is in fig. 2. Then the idea

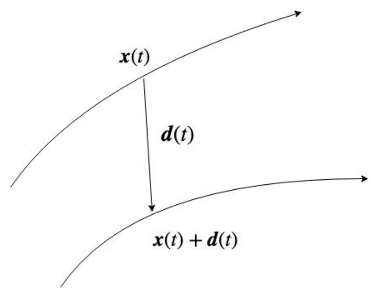



Figure 2: scheme for Lyapunov exponents

can be written as

$$\frac{d(t)}{d_0} = e^{\Lambda(t-t_0)}. \tag{2.1}$$

The variable $\Lambda$ is the Lyapunov exponent. We can see that if $\Lambda > 0$, then $\frac{d(t)}{d_0}$ grows exponentially with the passage of time; if $\Lambda < 0$ then the fraction shrinks to zero. If $\Lambda = 0$, then the time change in displacement over time is non-exponential. Note that if the system is continuous, we use base $e$. If the system is a discrete mapping, we use base 2. Solving the equation (2.1) for $\Lambda$ we obtain

$$\Lambda = \frac{1}{t - t_0} \ln \left| \frac{d(t)}{d_0} \right| .$$

This equation only provides a means for calculating the exponent for two specific neighbouring points over a specific interval of time. The approximation for the entire dynamical system is then the average of many different neighbourhoods.

**Definition 2.2.** If the displacement between the $i$-th point and a neighbouring point at time $t_i$ is $d_i$, and the initial displacement between the two points is $d_{0i}$ at time $t_{0i}$, then the Lyapunov exponent is defined as

$$\Lambda = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{t_i - t_{0i}} \ln \left| \frac{d_i}{d_{0i}} \right| .$$

This definition gives us a convenient formula to approximate $\Lambda$ for any dynamical system using computer. By randomly choosing a large number of neighbouring pairs of points and observing their relative movement to one another, the value $\Lambda$ can be easily calculated. However there also other definitions, varying on the literature.

Most dynamical systems have more than one Lyapunov exponent. If a system is $n$-dimensional, then it has $n$ Lyapunov exponents, one for each dimension. All combined, the set of exponents make up Lyapunov spectrum. In order to find if, we must calculate $\Lambda_i$ for small displacements in the i-th dimension for each $1 \leq i \leq$ n.

**Definition 2.3.** If $\Lambda_1 \geq \Lambda_2 \geq \ldots \geq \Lambda_n$ are Lyapunov exponents for dynamical system in $\mathbb{R}^n$, then the Lyapunov spectrum is the set $\{\Lambda_1, \Lambda_2, \ldots, \Lambda_n\}$.

Thanks to the definition, we can now identify chaos with following criterion:

$$\begin{aligned} \Lambda_1 > 0 &\iff \text{system is chaotic,} \\ \Lambda_1 \leq 0 &\iff \text{system is non-chaotic.} \end{aligned}$$

When the system has a positive maximal Lyapunov exponent, then there is a time horizon, beyond which the prediction breaks down, as shown schematically in fig. 3.

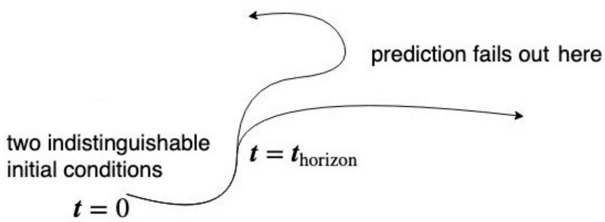



Figure 3: scheme for time horizon

Suppose that we measure initial conditions of experimental system very accurately. Because no measurement is perfect, there is always a some error $\|\delta_0\|$ between our estimate and true initial state. After a time $t$, the discrepancy grows to $\|\delta(t)\| \approx \|\delta_0\| e^{\Lambda t}$. Let $a$ be a measure of our tolerance, hence the prediction is tolerable if evolved state is within the distance $a$ of the true state. The value $a$ becomes intolerable if $\|\delta(t)\| \geq a$. That occurs after the time

$$t_{\text{horizon}} \approx O\left(\frac{1}{\Lambda} \ln \frac{a}{\|\delta_0\|}\right).$$

The logarithmic dependence on $|\delta_0|$ makes the system so much unpredictable. No matter how small the initial measurement error is, we cant predict longer than few multiples of $\frac{1}{\Lambda}$. It is easy to make a calculation that if we reduced our initial measurement error for given tolerance $a = 10^{-3}$ from $10^{-7}$ to $10^{-13}$, we would be able to predict the behaviour for only $2,5$ times longer. More examples can be found in [3].

## 2.2 Examples of chaotic systems

In this section a brief background to several chaotic dynamical systems will be given. Remark, that just the fact that we call a system chaotic does not mean, that it exhibits chaotic behaviour at all times. The solution curves of the dynamical systems depend on the parameters and chaos occurs for a special choice of them. The progress in investigating these systems dramatically increased with the invention of computers, however the analysis of the chaotic behaviour of particular systems such as Lorenz system is usually extremely difficult. There are some methods, that are based on the geometric models for particular differential equations, rather than on studying the equations themselves. This approach was used for example in [2]. In this work we will not give a very detailed analysis of dynamical systems but we focus on preparing an outline for the following chapter which is devoted to applications. For details one can study [2], [3] or [4].

### 2.2.1 The Lorenz system

In 1963, Edward Norton Lorenz attempted to set up a system of differential equations that would explain some unpredictable behaviour of the weather. Most viable models for weather prediction involve partial differential equations, but Lorenz sought a much simpler and system that could be easier to analyse.

Basic idea for the model is as follows. Imagine a planet with an atmosphere as a single layer composed of only one type of fluid particle. As on earth, this particle is heated from below and rises. When it gets high enough it gets cooled and then falls back down. It is natural to ask whether it is possible to predict a weather on such planet. The answer is no, but this model still gives us at least a basic handle for it. Lorenz looked at two-dimensional fluid cell that was heated and cooled as described above. The fluid motion can be described by infinitely many variables. One of the most drastic approximations Lorenz made was the assumption, that all but three of those variables remained constant. The remaining independent variables were transferred as convective "overturning" and then the horizontal and vertical temperature variation. The three-dimensional model then involved three parameters: the Prandtl number $\sigma$, the Rayleigh number $\rho$ and another parameter $\beta$ that is related to the physical size of the system. After all these simplifications the model was represented by the system of differential equation given by:

$$\begin{aligned}
\dot{x} &= \sigma(y - x), \\
\dot{y} &= x(\rho - z) - y, \\
\dot{z} &= xy - \beta z,
\end{aligned}$$

where all three parameters are positive and also $\sigma > \beta + 1$. Lorenz then discovered that this simple-looking deterministic system could have extremely erratic dynamics: over a wide range of parameters, the solutions oscillate irregularly, never exactly repeating but always remaining in a bounded region of phase space, which we now called the strange attractor. From this observation, the name The Butterfly Effect came from, because of attractors shape. The most famous phase portrait corresponds to the choice of parameters $\sigma = 10, \rho = 28$ and $\beta = 8/3$. Figure 4 corresponds to the phase portrait for these parameters.

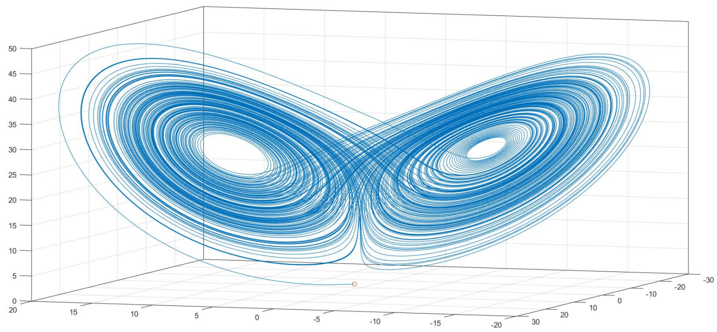



Figure 4: the Lorenz model

One of the iconic properties that the system has high sensitivity towards the small change of the initial condition is shown in the following pictures 5. For acquiring the pictures an algorithm 6.3 was used.

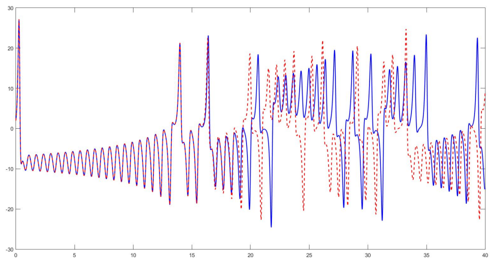
We can see in 5a and 5b that even though the change of initial condition is minimal, the trajectories approximately after time $t = 15$ start diverge from each other. The pictures 5c and 5d confirm that the trajectories are trapped inside the strange attractor. For thorough analysis of the system one can study [5] or [6], where the behaviour for various coefficients is examined.

In reality, the Lorenz system does not describe the weather, however it has still some applications. Same equations appear in models for lasers, electric circuits or dynamos.

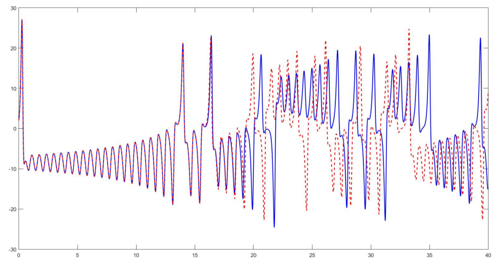
### 2.2.2 The Rössler system

In the 1970's a german biochemist Otto Rössler published a series of papers in which he studied prototype systems of ordinary differential equations in three-dimensional phase spaces. Among them was today's most famous one, called Rössler system. It was a successful attempt to find a system that is qualitatively similar, but is not that complicated. This system is in comparison to Lorenz system simpler, because it contains only one nonlinearity in the form of quadratic term. Furthermore, the solvability is its big advantage. It is also considered as minimal among all the chaotic dynamical systems, because of its form. The applications of the model are in the field of chemistry. The system in its original form is given by the equations:

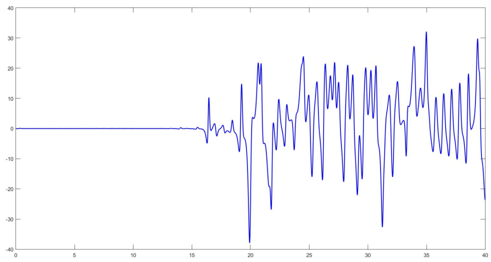
$$\begin{aligned} \dot{x} &= -y - z, \\ \dot{y} &= x + ay, \\ \dot{z} &= b + z(x - c). \end{aligned} \tag{2.2}$$



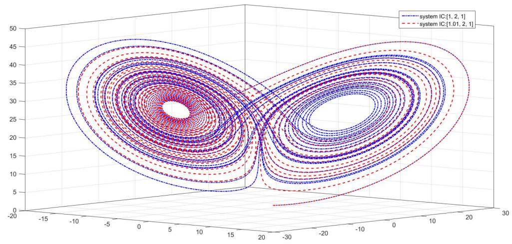
(a) the difference in $y$ variable



(b) the difference in $x$ variable



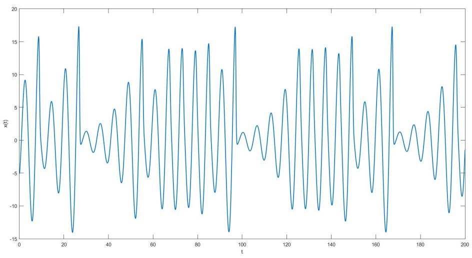
(c) the distance of trajectories in $y$ variable



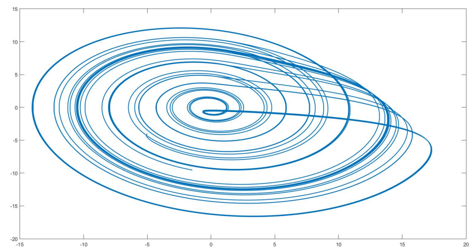
(d) 3D phase portrait of the systems

Figure 5: the trajectories for two nearby IC: $P_1 = [1, 2, 1]$ and $P_2 = [1.01, 2, 1]$

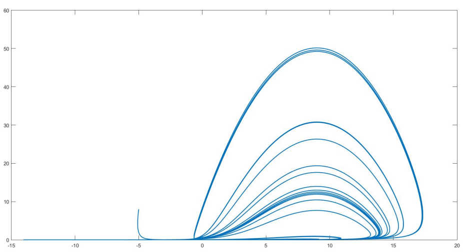
The original coefficients have values $a = 0.2$, $b = 0.2$ and $c = 5.7$. The phase portrait is shown in figure 6.



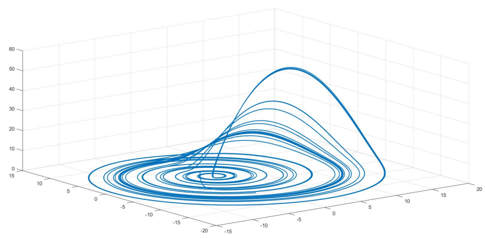
(a) The variable $x(t)$



(b) phase portrait on $xy$-plane



(c) phase portrait on $xy$-plane



(d) 3D phase portrait

Figure 6: the Rossler system for $a = 0.2$,$b = 0.2$ and $c = 5.7$

As we can see, the phase portrait consisted of only one blob, unlike from the previous Lorenz system that had two. For given choice of parameters the system acts chaotic. Roughly speaking, the point on the trajectory of Rössler system spirals in the $xy$-plane and once a while it enters the third dimension to perform a movement in the shape of a handle and then returns back to the plane. It can be shown that the linear terms in first two equations of the system create an oscillation. If we increase the value of parameter a, these oscillation will get amplified. The motion is then ruled by third equation, containing the non-linear term which makes reinjection back to the beginning of the spiraling-out motion. Precise analysis of the system including the numerical analysis for varying coefficient $c$ can be found in [4]. For acquiring the pictures an algorithm 6.3 was used.

### 2.2.3 Dynamical system of Chua's circuit

Among the many fields of science where the chaotic dynamical systems can be applied is the category of non-linear electronic circuits. In 1983, Leon O. Chua invented a simple electronic circuit whose dynamics exhibited a chaotic behaviour. It is a non-periodic oscillator and the the strange attractor is widely know as "the double scroll". The RLC circuit consists of four linear elements (two capacitors, inductor and linear resistor) and one non-linear element (non-linear resistor $N_r$). The scheme is in fig. 7. The model for

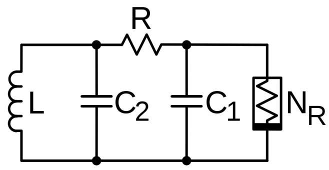



Figure 7: the scheme of the Chua's circuit

the system can be acquired by applying Kirchhoff's laws on the circuit:

$$\begin{aligned} C_1 \frac{dv_{C_1}}{dt} &= G\left(v_{C_2} - v_{C_1}\right) - f\left(v_{C_1}\right), \\ C_2 \frac{dv_{C_2}}{dt} &= G\left(v_{C_1} - v_{C_2}\right) + i_L, \\ L \frac{di_L}{dt} &= v_{C_2}, \end{aligned} \tag{2.3}$$

where $i$ is a current, $G = 1/R$ is conductance of the resistor, $C_1, C_2$ are the capacity and finally $L$ is inductance. The non-linear term is given by the function $f\left(v_{C_1}\right)$given by:

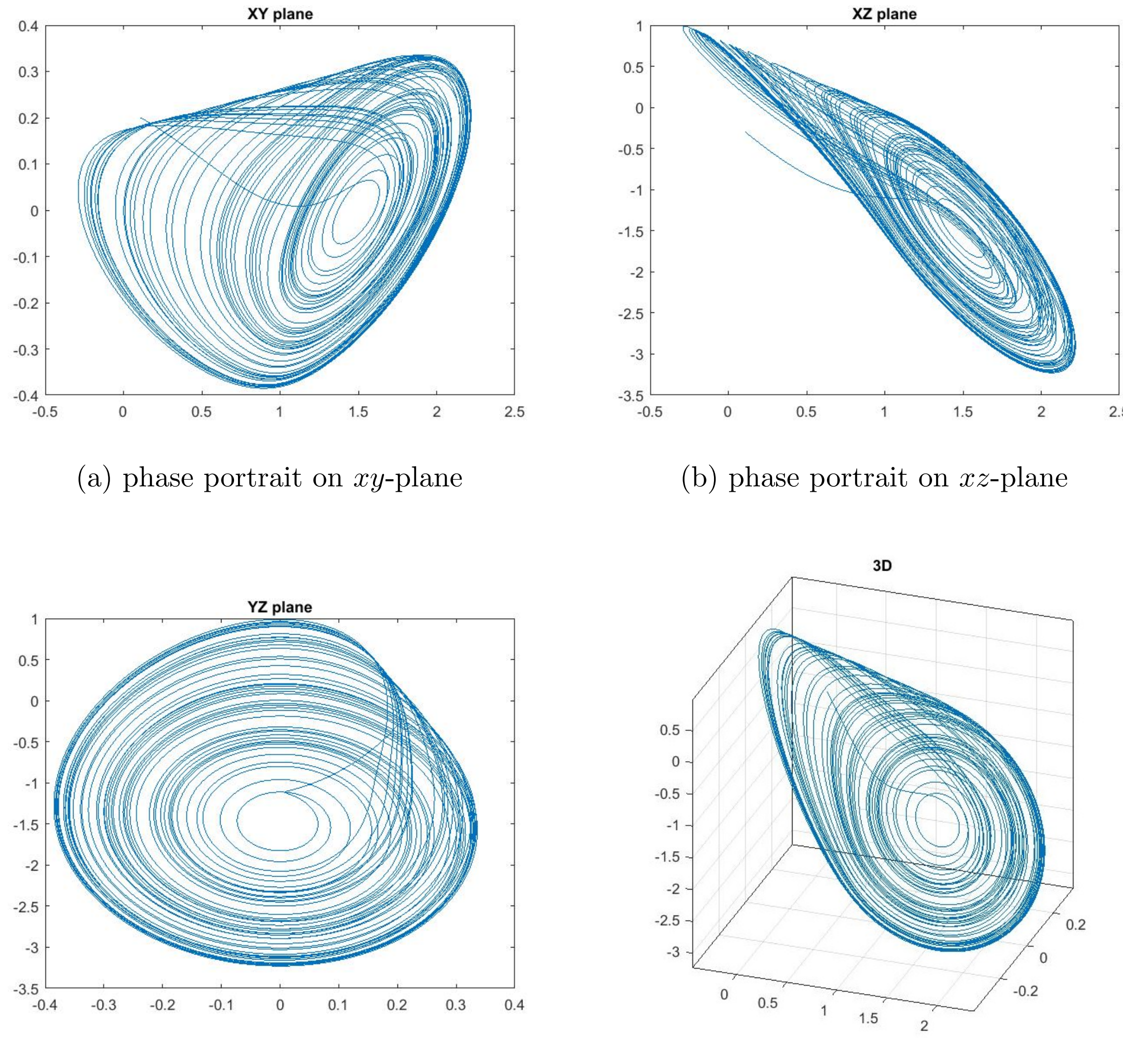
$$f\left(v_{C_1}\right) = m_0 v_{C_1} + \frac{1}{2}\left(m_1 - m_0\right)\left[\left|v_{C_1} + B_p\right| - \left|v_{C_1} - B_p\right|\right].$$

From a mathematical point of view, this form of the system is not suitable for analysis, therefore it is often showed in the form:

$$\begin{aligned} x' &= \alpha[y - x - f(x)], \\ y' &= x - y + z, \\ z' &= -\beta y, \end{aligned} \tag{2.4}$$

where $x = v_{C1}/B_p$, $y = v_{C_2}/B_p$, $z = i/B_pG$, $\alpha = C_2/C_1$, $\beta = C_2/G^2L$ and $f(x) = bx + \frac{1}{2}(a-b)[|x+1| - |x-1|]|$ with $a = m_1/G$, $b = m_0/G$.

In previous cases we experienced one type of strange attractor in the system for given all sets of parameters. Dynamical system of Chua's circuit produces different kinds of attractors for various combinations of parameters, namely simple attractor (fig. 8) and double-scroll attractor (fig. 9). One of the implementations of the Chua's circuit in applications can be as a generator of pseudo-random signals, since the chaotic behaviour causes non-periodic of the outputs. The analysis of the system can be found in [7]. For acquiring the pictures an algorithm 6.1 was used.



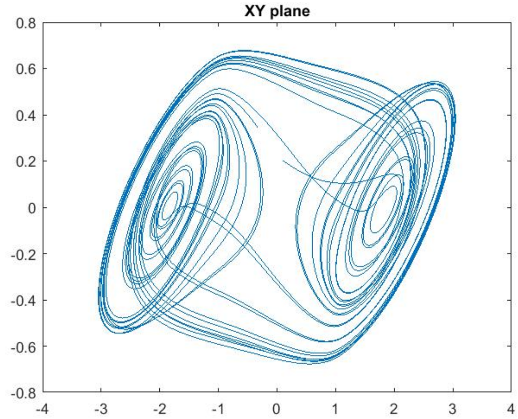

(a) phase portrait on $xy$-plane

(b) phase portrait on $xz$-plane

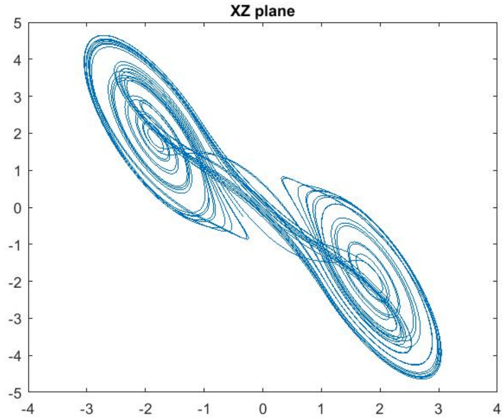
(c) phase portrait on $yz$-plane

(d) 3D phase portrait

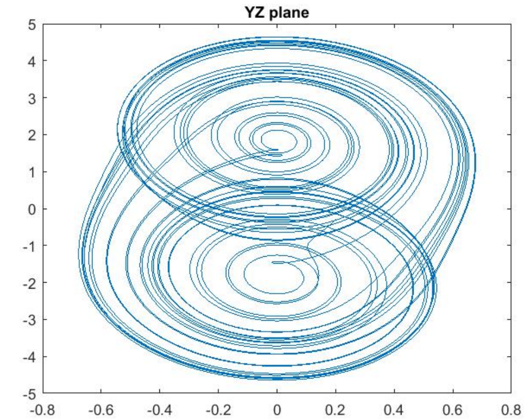
Figure 8: Chua's circuit system for $\alpha = 9.4$, $\beta = 16$, $m_1 = -1.14$, $m_0 = -0.71$



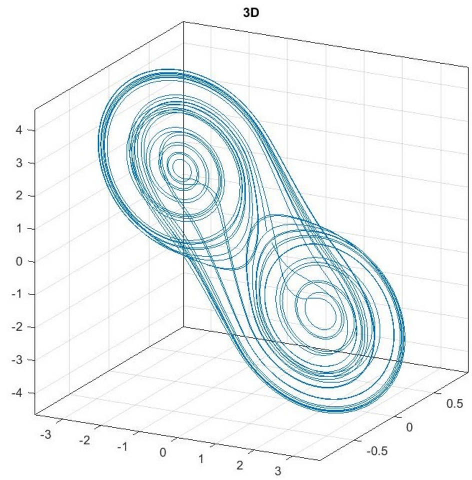

(a) phase portrait on $xy$-plane




(b) phase portrait on $xz$-plane




(c) phase portrait on $yz$-plane




(d) 3D phase portrait

Figure 9: Chua's circuit system for $\alpha = 10$, $\beta = 15$, $m_1 = -1.27$,$m_0 = -0.68$

# 3 Synchronization of chaos

In previous chapters we mentioned that the sensitivity towards the change of initial conditions is crucial in terms of behaviour of some systems. The two points will separate from each other exponentially fast, but remain trapped in the complicated structure called the strange attractor. The phase portraits then cover wide range of extraordinary shapes.

By synchronization we mean agreement or correlation of distinct processes in time. Historically, the analysis of synchronization phenomena in the evolution of dynamical systems has been a subject of active investigation since the earlier days of physics. In the 17th century the Huygens found that two very weakly coupled pendulum clocks, that were hanging at the same beam, became synchronized in phase. In the last century, the search for synchronization has moved to chaotic dynamical systems. Due to the nature of the system one would say that they intrinsically defy the synchronization. That is because even two identical systems starting from slightly different initial conditions would evolve in time in an unsynchronized manner, therefore there is no correlation. This is a relevant practical problem. The initial conditions observed experimentally are never known perfectly.

The definition of synchronization of chaos is slightly generalized. We will refer to synchronization of chaos as a process where two (or many) chaotic systems (either equivalent or non-equivalent) adjust a given property of their motion to a common behaviour, due to coupling or forcing. This ranges from complete agreement of trajectories to locking of phases.

There are differences in the process leading to synchronized states, depending upon the particular coupling configuration. Namely, two main cases should be distinguished: unidirectional coupling and bidirectional coupling.

In the former case, a global system is formed by two subsystems that realize a drive–response (or master–slave) configuration. That results are that one subsystem evolves freely and drives the evolution of the other. As a result, the response system is slaved to follow the dynamics of the drive system, even though the drive systems is acting chaotic. On the other hand, the bidirectional coupling involves subsystems being coupled with each other, which induces a mutual synchronization behaviour. This situation typically occurs in physiology, e.g. between cardiac and respiratory systems or in non-linear optics, e.g. coupled laser systems with feedback.

In the past few decades, many different synchronization states have been studied. Namely: complete synchronization (CS), generalized synchronization (GS), phase and lag synchronization and more.

Among all, the CS was the first one being discovered and it is the simplest form of synchronization in chaotic systems. The main feature is the perfect hooking of the chaotic trajectories of two systems which is achieved by means of a coupling signal, in such a way that they remain in step with each other over the course of the time.

GS in the generalization of CS. Main goal is to synchronize completely different systems and associating the output of one system to a given function of the output of the other system.

The other ways of synchronization do not explicitly require the trajectories to converge to each other completely but only to some certain level. They are also more suitable for synchronization of arbitrary systems.

To achieve the phase synchronization, the weak coupling of systems is used. During the process, the phases become locked on some value and provide the trajectories to be in phase with each other, while the amplitudes may differ completely.

In the case of lag synchronization, the states of the two systems become nearly identical due to a stronger coupling, but the trajectories are shifted in time. For an extended overview, one should study [8].

In the following part, deeper investigation of the unidirectional coupling method is provided, namely: complete synchronization and synchronization using negative-feedback. The results are then applied on the Lorenz system and the Rössler system, described in previous chapter. At the end, both methods are compared.

## 3.1 Complete Synchronization

One of the first papers [10], in which the synchronization of two chaotic systems was investigated was published in 1990 by Louis Pecora and Thomas Carroll. They were working in Naval Research Lab in Washington D. C. and they were asking a question, whether it is possible to synchronize two chaotic systems in any way. From outer perspective it would seem that chaotic behaviour cannot be tamed by any manner, but they discovered that certain chaotic dynamical systems have a self-synchronization property. That means that if we take two identical systems, both starting with different initial condition, after some time they will synchronize and the trajectories will coincide.

The method belongs to the category of Drive-Response ("Master and Slave") configurations. First, a global system is formed and then it is divided into two subsystems. Next, one of those is acting freely and the second one has no choice but following it's "master's" evolution and synchronize itself with it. We now discuss this method more precisely.

Consider an autonomous $n$-dimensional dynamical system

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}).$$

We arbitrarily divide the system into two subsystems $\boldsymbol{x} = (x_1, \ldots, x_n) = (\boldsymbol{d}, \boldsymbol{r})$,

$$\begin{aligned} \dot{\boldsymbol{d}} &= f_d(\boldsymbol{d}, \boldsymbol{r}), \\ \dot{\boldsymbol{r}} &= f_r(\boldsymbol{d}, \boldsymbol{r}), \end{aligned}$$

where

$$\boldsymbol{d} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}, f_d = \begin{pmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \\ \vdots \\ f_m(\boldsymbol{x}) \end{pmatrix}, \boldsymbol{r} = \begin{pmatrix} x_{m+1} \\ x_{m+2} \\ \vdots \\ x_n \end{pmatrix}, f_r = \begin{pmatrix} f_{m+1}(\boldsymbol{x}) \\ f_{m+2}(\boldsymbol{x}) \\ \vdots \\ f_n(\boldsymbol{x}) \end{pmatrix}.$$

We now create a new subsystem $\boldsymbol{r'}$, which is identical to the system $\boldsymbol{r}$. We replace the set of variables in $\boldsymbol{x}$ for the corresponding $\boldsymbol{r'}$ in the function $\boldsymbol{f_r}$ and augment the the previous equations. After this, we remain with two systems, one of them is the original and the second is the induced one.

$$\begin{aligned} \dot{\boldsymbol{d}} &= f_d(\boldsymbol{d}, \boldsymbol{r}), \\ \dot{\boldsymbol{r}} &= f_r(\boldsymbol{d}, \boldsymbol{r}), \end{aligned} \qquad \begin{aligned} \dot{\boldsymbol{d}} &= f_d(\boldsymbol{d}, \boldsymbol{r}), \\ \dot{\boldsymbol{r}} &= f_r(\boldsymbol{d}, \boldsymbol{r}), \\ \dot{\boldsymbol{r}}' &= f_r(\boldsymbol{d}, \boldsymbol{r}'). \end{aligned}$$

Our aim was to split the system into two types of variables, one of which will serve as a drive and the other as a response.

The systems synchronize completely, if we require that the difference in the response variable $\Delta \boldsymbol{r} = \boldsymbol{r} - \boldsymbol{r'}$ goes to 0 as $t \to \infty$. In the infinitesimal limit this leads to the variational equation for the subsystem

$$\dot{\boldsymbol{\xi}} = D_r \boldsymbol{f_r}(\boldsymbol{d}(t), \boldsymbol{r}(t))\boldsymbol{\xi},$$

where the $D_r f_r$ is the Jacobian of the $\boldsymbol{r}$ subsystem vector field with respect to $\boldsymbol{r}$. The behaviour of this equation depends on the Lyapunov exponents of the response $\boldsymbol{r}$ subsystem. If we assume, that the two systems have the same parameters, then the condition for the two subsystems under which they synchronize is that the Lyapunov exponents of the $\boldsymbol{r}$ subsystem are all negative. To see this, we show on a concrete example of the Lorenz system that this phenomena of synchronization occurs.

### 3.1.1 Synchronization of the Lorenz system

We work with Lorenz equations given by:

$$\begin{aligned} \dot{x} &= \sigma(y - x), \\ \dot{y} &= x(\rho - z) - y, \\ \dot{z} &= xy - \beta z. \end{aligned} \tag{3.1}$$

Using a procedure which we described above, we create two systems and examine their synchronization. In the driving system, we choose $\boldsymbol{d} = x$ as our driving variable and $\boldsymbol{r} = (y, z)$ is the response. The driving system is then given by (3.1) and we obtain the response system given by the equations:

$$\begin{aligned} x_r &\equiv x(t), \\ \dot{y_r} &= x(t)(\rho - z_r) - y_r, \\ \dot{z_r} &= x(t)y_r - \beta z_r. \end{aligned} \tag{3.2}$$

We can think of the driving variable as a signal, which we put into the response system. In this case the system is then formed by 5 equations (3 for drive and 2 for response). We claim that the $(y, z)$ and $(y_r, z_r)$ subsystems synchronize. Assume that the parameters $\sigma, \rho, \beta$ match. Considering that the $x$ variable remains unchanged, the two systems will synchronize if $|y - y_r| \to 0$ and $|z - z_r| \to 0$ as $t \to 0$. In 1992, He and Vaidya [12] published a paper with a simple proof of the synchronization. Moreover, the two subsystems synchronize exponentially fast (see [15]). We present their method here.

**Proof of synchronization for Lorenz system** We create corresponding system of differential equations that will represent the difference (error) between the pairs of variables. Let

$$\begin{aligned} \dot{e_x} &= \dot{x} - \dot{x}_r, \\ \dot{e_y} &= \dot{y} - \dot{y}_r, \\ \dot{e_z} &= \dot{z} - \dot{z}_r, \end{aligned} \tag{3.3}$$

then

$$\begin{aligned} e_x &\equiv 0 \\ \dot{e_y} &= -(y - y_r) - x(t)(z - z_r) = -e_y - x(t)e_z, \\ \dot{e_z} &= x(t)(y - y_r) - \beta(z - z_r) = x(t)e_y - \beta e_z, \end{aligned} \tag{3.4}$$

There is no error in $x$ variable because the exact signal $x(t)$ is substituted into the response system. We get linear two-dimensional non-autonomous system for the errors. Next it is shown that $e_y$ and $e_z$ go to 0 as $t \to \infty$. The goal is to eliminate the $x(t)$ term, because it is explicitly time dependent and it is possibly also the source of chaotic behaviour. To achieve this, we multiply the second equation by $e_y$ and the third equation by $e_z$ and add them together. We get this equation:

$$\begin{aligned} \dot{e_y} e_y + \dot{e_z} e_z &= - e_y^2 - x(t) e_y e_z + x(t) e_y e_z - \beta e_z^2, \\ &= - e_y^2 - \beta e_z^2. \end{aligned} \tag{3.5}$$

Notice that on the left-hand side the equation is in the form of differential and it can be rewritten in the form

$$\frac{d}{dt}\left(\frac{1}{2}[e_y^2 + e_z^2]\right) = -e_y^2 - \beta e_z^2.$$

We can take the term in the derivative as a candidate for Lyapunov function $V(t)$. Let

$$V(t) := \frac{1}{2}(e_y^2 + e_z^2).$$

This function is certainly positive definite since it is a sum of squares and its derivative, which is given by 3.5, is $\dot{V}(t) < 0$ except at the origin $(0, 0)$, therefore it is negative definite. By the Lyapunov theorem it is the Lyapunov function and $V(t) \to 0$ as $t \to 0$ and $\boldsymbol{e} = (e_x, e_y, e_z) = \mathbf{0}$ is asymptotically stable. That implies that $e_y \to 0$ and $e_z \to 0$ as $t \to 0$ and the system synchronizes with the drive. We now show that the errors decay exponentially fast.

Let $V(t) = \frac{1}{2}(e_y^2 + e_z^2)$. Then $V$ decays exponentially fast, because

$$\dot{V} = -e_y^2 - \beta e_z^2 \leq -kV,$$

for any $k = \min(2, 2b)$. Integration then yields $0 \leq V(t) \leq V_0 e^{-kt}$ so that $e_y(t) \leq (V_0)^{\frac{1}{2}} e^{\frac{-kt}{2}}$. Similarly, $e_z(t) \leq O(e^{\frac{-kt}{2}})$, thus all components $\boldsymbol{e} = (e_x, e_y, e_z)$ decay at least exponentially fast.

**Geometry of the phase-space attractor** To give a geometrical perspective on the synchronization, we describe the behaviour of the systems in so called *synchronization manifold*. We work with the variables $x, y$ and $y_r$. Since after some time $y = y_r$, we see that the motion remains on the plane defined by this equality. Similarly, the motion must remain on the plane defined by $z = z_r$. Such equalities define a hyperplane in the five-dimensional state space. Its projection in three-dimensional space can be seen in figure 10. The constraint of motion to a hyperplane and the existence of complete synchronization are really one and the same as it is shown below.

Pecora and Carrol in 1997 [13] described that if the motion, given by the equations, is continually confined to a hyperplane in phase space, then there is an identical synchronization in any system, whether it is chaotic or not. If we change the coordinates with a constant linear transformation, the geometry remains the same. This change just represents a change of variables in the the equations of motion. Then we can assume, that the hyperplane contains the origin of the coordinates since this is just a simple translation that also maintains the geometry. These observations give the following result: the space orthogonal to the synchronization manifold, called the *transverse* space, has coordinates

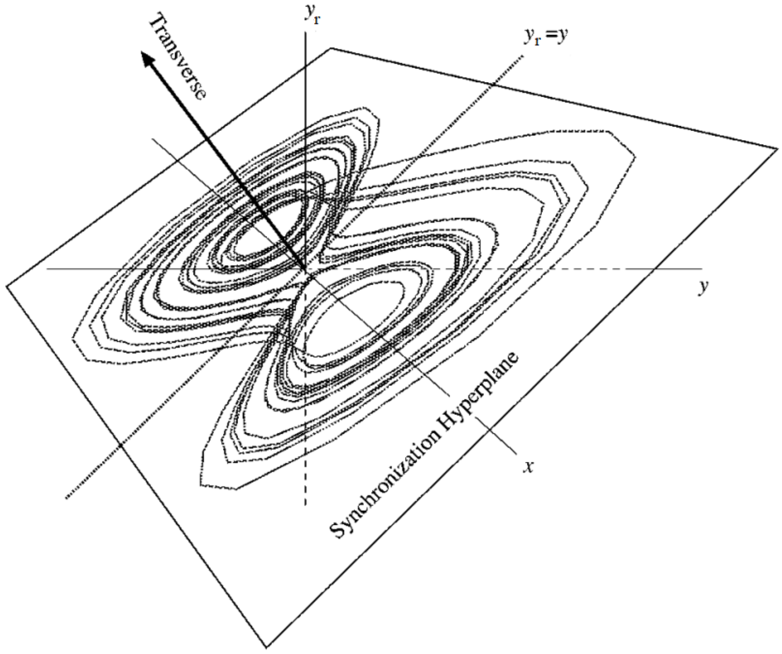



Figure 10: the synchronization manifold for Lorenz system [13]

that will be zero when the motion is on the synchronization manifold. Also as it has been described in [9], if the synchronization manifold is stable, the systems will synchronize.

We now apply this approach on example of two Lorenz systems. We showed that the differences among $y, y_r$ and $z, z_r$ tend to 0 as $t \to \infty$. This occurs because the synchronization manifold is stable. To see this, we transform the system to a new set of coordinates: $x$ stays the same and we let

$$\begin{aligned} y_\perp &= y - y_r, \\ y_\parallel &= y + y_r, \\ z_\perp &= z - z_r, \\ z_\parallel &= z + z_r. \end{aligned}$$

We have transformed the system to a new set of coordinates in which three coordinates are on the synchronization manifold $(x, y_\parallel, z_\parallel)$ and two are on the transverse manifold $(y_\perp, z_\perp)$. Now, at the very least for the subsystems to synchronize, we need to have $y_\perp$ and $z_\perp$ go to zero as $t \to 0$. Thus, the zero point $(0, 0)$ in the transverse manifold must be a fixed point within that manifold.

This requires the dynamical subsystem $(y_\perp, dz_\perp)$ to be stable at the origin. In the limit of small perturbations (represented by those variables $y_\perp$ and $z_\perp$) we end up with variational equations for the response. Next, we approximate the differences in the vector fields by the Jacobian, the matrix of partial derivatives of the right-hand side of the $(y, z)$-response system.

More precisely, let $\boldsymbol{H}$ be the two-dimensional function that is the right-hand side of the response system (3.2). We assume small perturbations. Then we have:

$$\begin{pmatrix} \dot{y}_\perp \\ \dot{z}_\perp \end{pmatrix} = \boldsymbol{H}(y, z) - \boldsymbol{H}(y_r, z_r) \approx D\boldsymbol{H} \begin{pmatrix} y_\perp \\ z_\perp \end{pmatrix} = \begin{pmatrix} -1 & -x \\ x & -b \end{pmatrix} \begin{pmatrix} y_\perp \\ z_\perp \end{pmatrix}. \tag{3.6}$$

The solution of these equations tell us, whether the $y_\perp$ and $z_\perp$ on the transverse manifold shrink and systems will synchronize or they grow and the opposite case will happen. Pecora and Carroll showed that in general the minimal condition for stability and for the systems to synchronize is to have the Lyapunov exponents, associated with the equation (3.6), negative for the transverse subsystem. We observe that this is the same as requiring the response subsystem $(y_r, z_r)$ to have negative Lyapunov exponents. That is, we treat the response as a separate dynamical system driven by variable $x$ with its own Lyapunov exponents. These exponents will depend on $x$ and for that reason they are referred as *conditional* Lyapunov exponents. The algorithms for numerical computing the conditional Lyapunov exponents (CLE) are beyond the scope of this work. See [**?**] or [**?**] for further information. We now show some concrete examples of synchronization.

### 3.1.2 Examples of complete synchronization

In this subsection we show that for certain choice of driving variable the Lorenz systems will synchronize. The occurrence of complete synchronization depends on the conditional Lyapunov exponents of the subsystem. We show a table 1 of the associated exponents for various subsystems and demonstrate the trajectories in graphs.

| **System** | **Drive signal** | **Response system** | **Conditional Lyapunov exponents** |
|---|---|---|---|
| | x | (y,z) | (-1,81 ; -1,86) |
| Lorenz | y | (x,z) | (-2,67 ; 9,99) |
| | z | (x,y) | (0,0108 ; -11,01) |
| | x | (y,z) | (0,2 ; -8,89) |
| Rössler | y | (x,z) | (-0,056 ; -8,81) |
| | z | (x,y) | (0,1 ; 0,1) |

Table 1: CLE for Lorenz system ($\sigma = 10$, $\rho = 28$, $\beta = 8/3$) and Rössler system ($a = b = 0,2$; $c = 9$)

For the computation in this work we used an approximated version of the differential equations of the systems. We acquired the results using a procedure ode45 in software MATLAB, which is based on an explicit Runge-Kutta (4,5) formula. In eligible cases the Euler's method with time step $h = 0.01$ was used. The phase portrait of the Lorenz system is depicted in figure 11.

**y-drive** As we can see in figure 12, for the y-drive the systems will completely synchronize after few steps. The error $e_y$ is identically 0 since we are comparing the same data set.

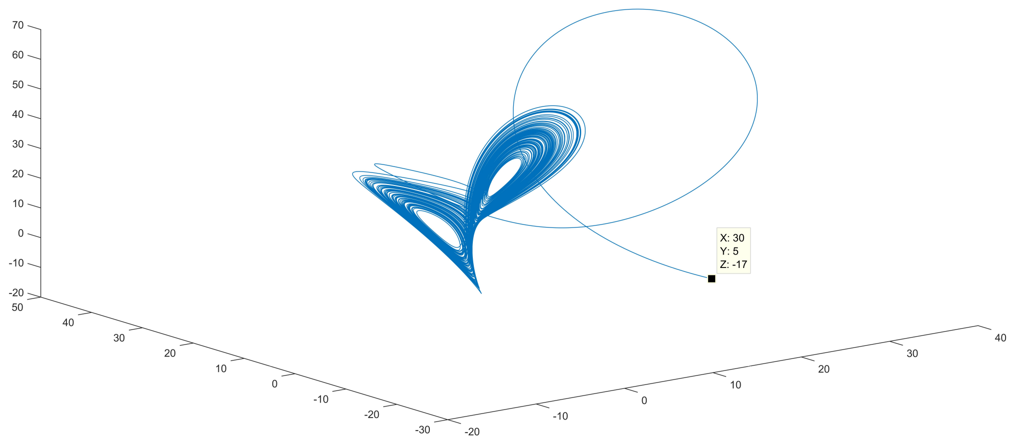
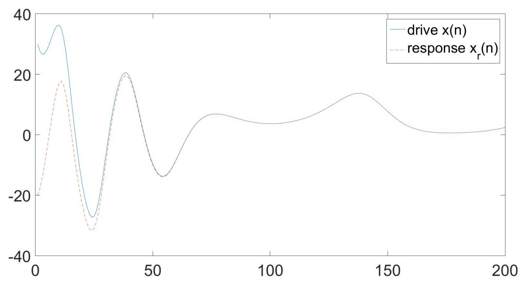
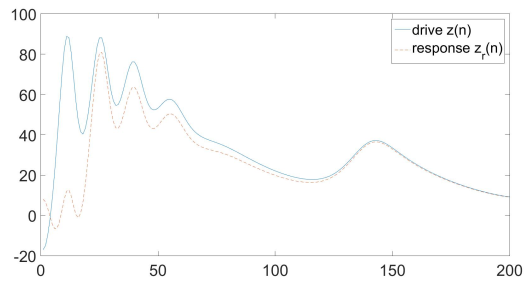


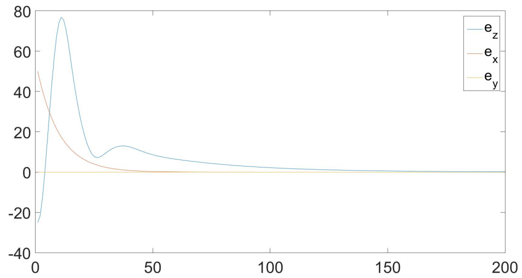

Figure 11: the Lorenz system



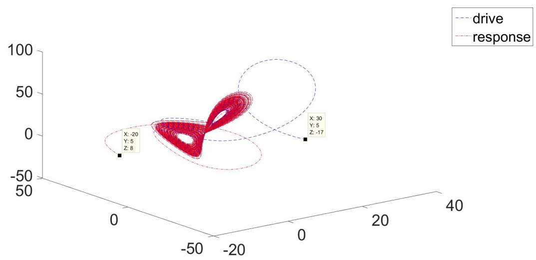

(a) difference in variable $x$




(b) difference in variable $z$



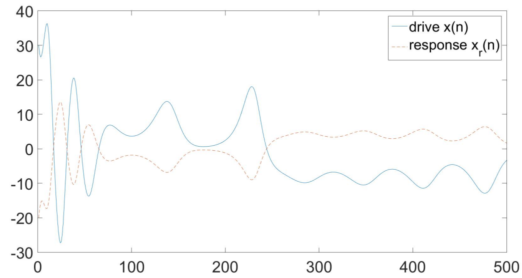

(c) errors in the system



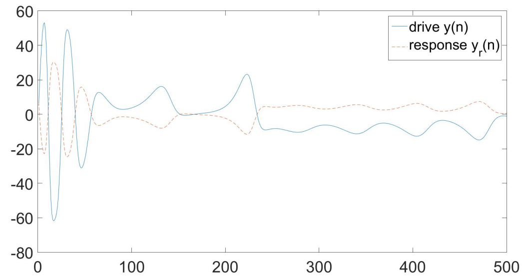

(d) drive and response system in 3D

Figure 12: the Lorenz system with driving variable $y$

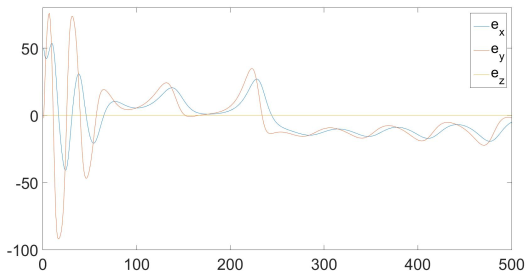
**z-drive** For our case of the Lorenz system, if $z$ is taken as a driving variable, the necessary condition for the conditional exponents to be negative is not satisfied. That makes the two subsystems to not synchronize. In picture 13d we can see that even though the trajectories are trapped inside the strange attractor, they will never synchronize with each other like in previous example.



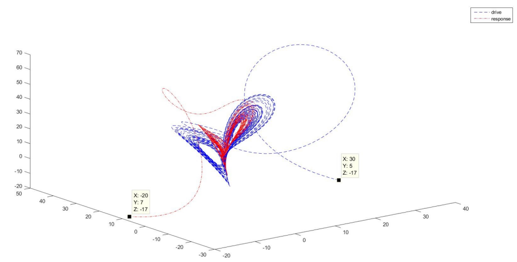

(a) difference in variable $x$




(b) difference in variable $y$




(c) errors in the system



(d) drive and response system in 3D

Figure 13: the Lorenz system with driving variable $z$

For acquiring the pictures an algorithm 6.2 was used.

## 3.2 Negative-feedback control

Another approach, for which is the process of complete synchronization a special case, is *diffusive* coupling of systems, often called negative-feedback control. It also belongs to the group of "drive-response" methods of synchronization. We describe the process below.

Consider two autonomous $n$-dimensional dynamical systems with the same parameters

$$\begin{aligned}\dot{\boldsymbol{x_1}} &= f(\boldsymbol{x_1}),\\ \dot{\boldsymbol{x_2}} &= f(\boldsymbol{x_2}).\end{aligned}$$

One of them is drive and the other is response. We connect these two systems by adding a damping term to the response system, which consists of the difference between the drive and response variables. We have:

$$\begin{aligned}\dot{\boldsymbol{x_1}} &= f(\boldsymbol{x_1}),\\ \dot{\boldsymbol{x_2}} &= f(\boldsymbol{x_2}) + \alpha \boldsymbol{E}(\boldsymbol{x_1} - \boldsymbol{x_2}),\end{aligned}$$

where $\alpha$ is a positive coefficient of coupling strength and $\boldsymbol{E}$ is $n$-dimensional square matrix that determines the linear combination of $x$ components in the coupling. It was observed in [13] that to achieve the synchrony between the two systems, the parameter $\alpha$ must be chosen in the way that the largest Lyapunov exponent (denoted $\lambda_{\perp}^{max}$) of the response system must be negative. We use this method to synchronize two Rössler systems.

### 3.2.1 Example of negative-feedback synchronization

Consider two Rössler systems with parameters $a = 0,2,\ b = 0,2,\ c = 9,0$:

$$\begin{aligned} \dot{x_1} &= -y_1 - z_1, & \dot{x_2} &= -y_2 - z_2 + \alpha(x_1 - x_2), \\ \dot{y_1} &= x_1 + ay_1, & \dot{y_2} &= x_2 + ay_2, \\ \dot{z_1} &= b + z_1(x_1 - c), & \dot{z_2} &= b + z_2(x_2 - c). \end{aligned}$$

In this case we have chosen

$$\boldsymbol{E} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

We now follow the similar procedure as in 3.1.1. We express the variational equation of the system in the variables $(x_\perp, y_\perp, z_\perp)$ lying on the transverse manifold. From that it is possible to estimate the Lyapunov exponents. In fact it is sufficient to compute only the maximal Lyapunov exponent $\lambda_\perp^{max}$.

$$\begin{pmatrix} \dot{x}_\perp \\ \dot{y}_\perp \\ \dot{z}_\perp \end{pmatrix} \approx \begin{pmatrix} -\alpha & -1 & -1 \\ 1 & a & 0 \\ z & 0 & x - c \end{pmatrix} \begin{pmatrix} x_\perp \\ y_\perp \\ z_\perp \end{pmatrix}.$$

In this case, it is dependent on the value of $\alpha$. It was shown in [14] that the function expressing the dependency of $\lambda_\perp^{max}$ on $\alpha$ can be estimated. It tells us if these perturbations will damp out or not and, hence whether the synchronization state is stable or not.

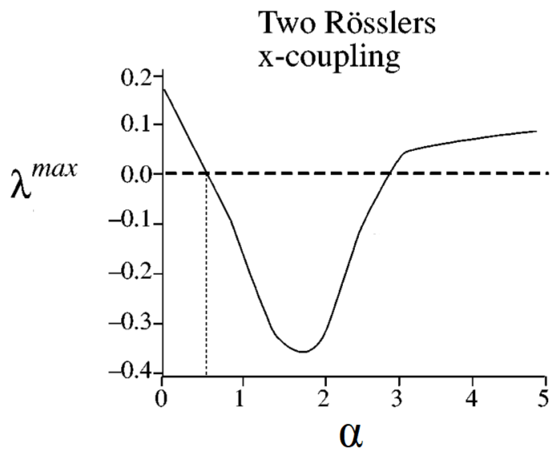
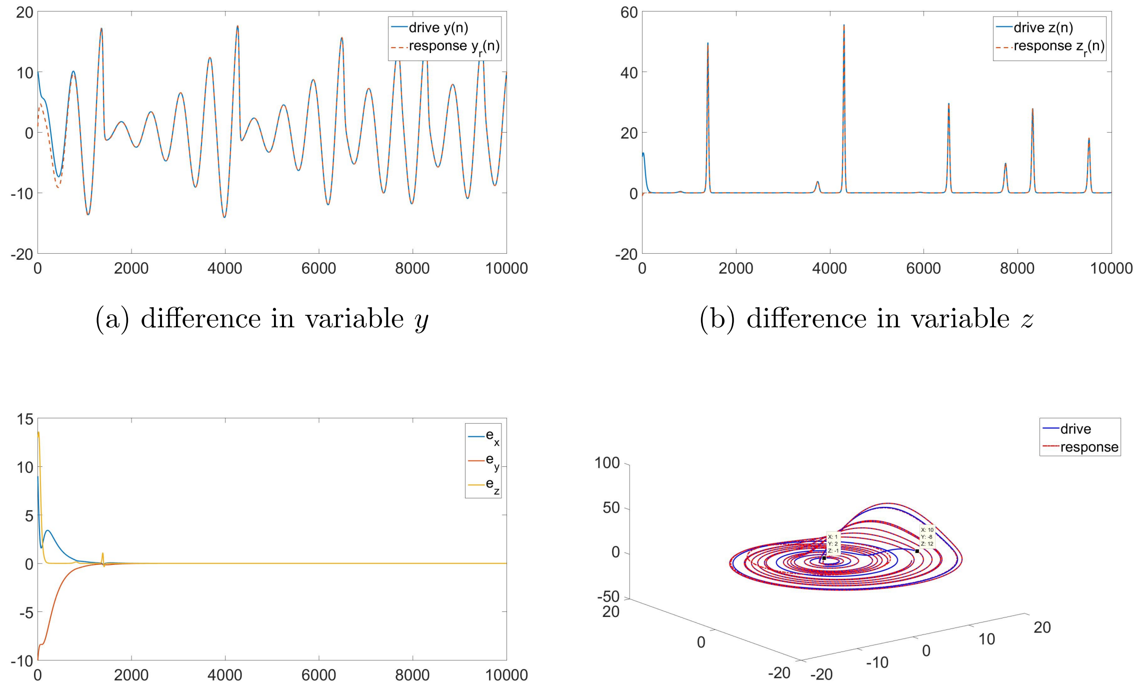



Figure 14: The maximum transverse Lyapunov exponent $\lambda_\perp^{max}$ as a function of coupling strength $\alpha$ in the Rössler system.

As we can see in picture 14 for the case of two Rössler systems, the systems start being more connected and the $\lambda_\perp^{max}$ is decreasing with increasing coupling strength $\alpha$. At some intermediate value of $\alpha$, we will synchronize the systems. For larger values of $\alpha$ the desynchronization occurs. However, it was observed that at very large values of $\alpha$ we slave the $x_2$ to $x_1$. We would achieve this enslavement, if we replaced all the $x_2$ variables in the response system to $x_1$. The method of complete synchronization is then a limit case of negative-feedback control when $\alpha \to \infty$. It is an open problem to find the value of $\alpha$ for which the function $\lambda_{\perp}^{max} = \lambda_{\perp}^{max}(\alpha)$ becomes negative again.

In pictures 15 and 16 there are two synchronized systems if coupled with the parameter $\alpha = 2$ and $\alpha = 100$. The pictures were acquired using algorithm 6.4.




(a) difference in variable $y$

(b) difference in variable $z$

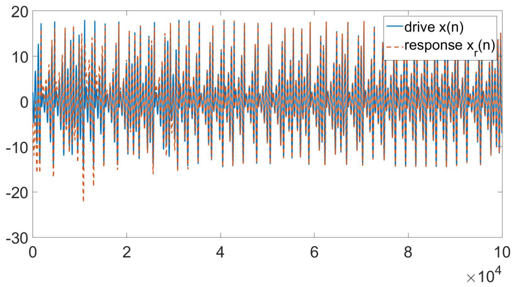
(c) errors in the system

(d) drive and response system in 3D

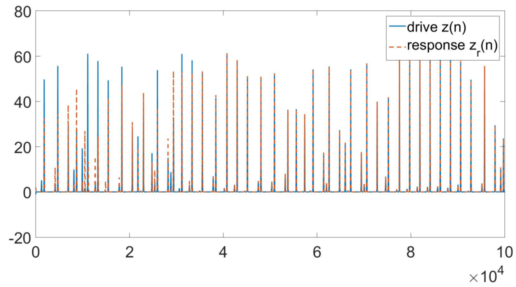
Figure 15: synchronization using negative-feedback method with $\alpha = 2$, $y$-drive

Next we can see, that for large parameters $\alpha$ (chosen $\alpha = 100$), the negative-feedback method and complete synchronization method coincide. Note, that in order for the systems to synchronize completely, the coupling must be added to those variables that can serve as a drives, otherwise complete synchronization does not occur.

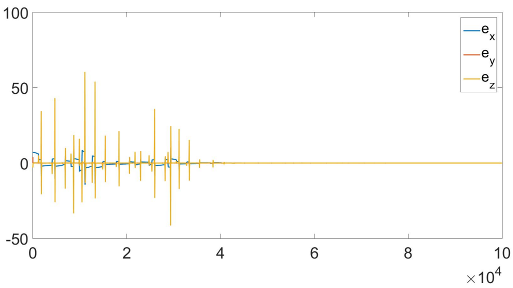
To demonstrate that for large $\alpha$ the methods start having the same effect, we can look at graph 17 that shows us the difference between errors in synchronization by complete synchronization (1) and by negative-feedback (2). For example, the difference in $x$ variable of the synchronization by CS $e_{x_1} = |x - x_r^{CS}|$ and by NF $e_{x_2} = |x - x_r^{NF}|$ goes to zero as $t \to \infty$. The comparison and the picture were done by algorithm 6.5.



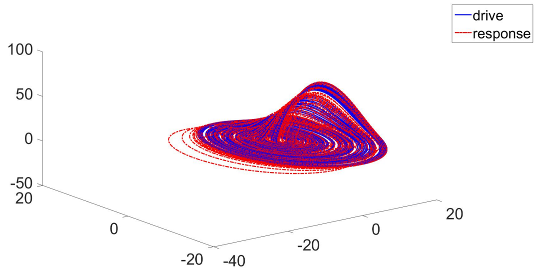

(a) difference in variable x




(b) difference in variable z




(c) errors in the system




(d) drive and response system in 3D

Figure 16: Synchronization using negative-feedback method with $\alpha = 100$, y-drive

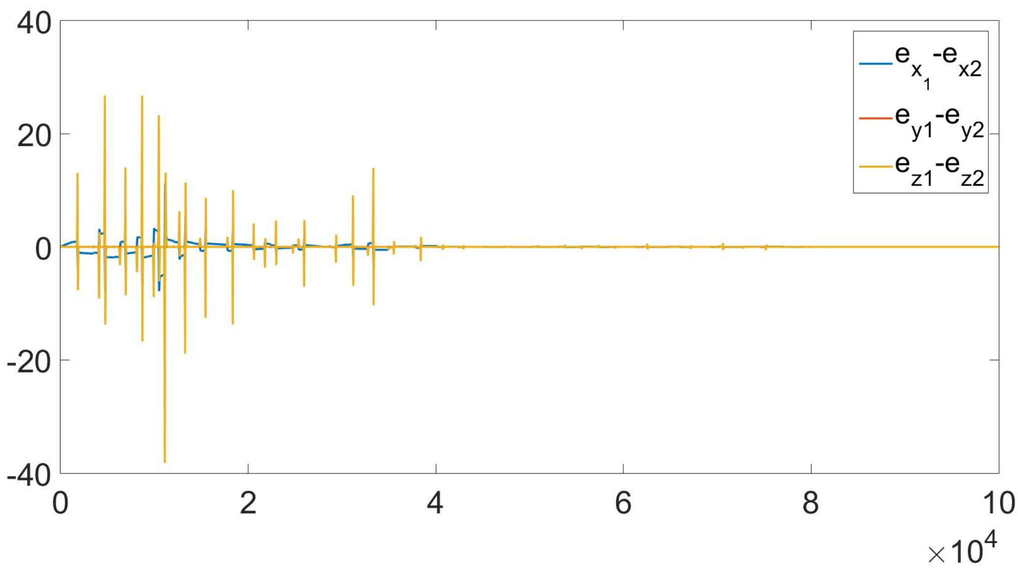



Figure 17: difference between NF and CS method

# 4 Applications of the synchronization

In control theory, or some other fields, we normally want to avoid chaotic behaviour of the system. If possible, we try remove it by the change of the parameters etc. Since last forty years, mathematicians and engineers have been thinking about the question, whether the chaos can be useful in some way. One of the interesting applications of it involve private communication. In 1992, Cuomo and Oppenheim [11] brought in a new view point on the applications of synchronization, expanding the work on Pecora and Carroll.

Suppose that you have an information in analog or digital form that is private and you want to send it to through a public channel. Our desire is, when somebody intercepts it, he will not be capable of extracting the information. For example in the early days of cellphones, when the communication was not ciphered, there were cases, when the calls have been intercepted by third parties and the information have been taken advantage of. Naturally, you want to mask the information. In fact, you can use the self-synchronizing property od chaotic dynamical systems for that purpose.

You can think of a the information as a signal function of $t$, which you mask into the much louder chaotic signal and you send it via public channel. If someone intercepts it and tries to read it or listen to it, he will recognize only the noise generated by the chaotic system. If the receiver has a device, which is in sync with the sender's transmitter, then it is possible to filter the chaos away using synchronization and you are left with the original message. In this chapter show two possible approaches on how to implement this idea.

For our experiment we use the Lorenz system

$$\begin{aligned}\dot{x} &= \sigma(y-x),\\ \dot{y} &= x(\rho-z)-y,\\ \dot{z} &= xy-\beta z,\end{aligned} \tag{4.1}$$

where $\sigma = 10$, $\rho = 28$ and $\beta = 8/3$. In this work, there were attempts to use also the Rössler system described in chapter 2.2. Unfortunately it was much more difficult task to perform the procedure of unmasking, due to the "stronger" chaotic behaviour. It is an open problem to find an optimal way on how to implement Rössler system to serve this purpose.

We will operate with two Lorenz systems, whose dynamics are prescribed by (4.1). One of them will serve as a transmitter and the other as a receiver (we denote the variables with the index $r$). Each will start from a different initial condition. We denote

$$\begin{aligned}&\boldsymbol{d}(t) = (x(t), y(t), z(t)) && \ldots \quad \text{state of the transmiter (drive)},\\ &\boldsymbol{r}(t) = (x_r(t), y_r(t), z_r(t)) && \ldots \quad \text{state of the receiver (response)},\\ &\boldsymbol{e}(t) = \boldsymbol{d}(t) - \boldsymbol{r}(t) && \ldots \quad \text{error signal}.\end{aligned}$$

In chapter 3.1 we showed that the system can be decomposed into two subsystems with drive signals $x$ or $y$ and responses $(y, z)$ or $(x, z)$ respectively, which will synchronize exponentially fast. It is possible to use this property to regenerate two full-dimensional systems that become synchronized.

We take the drive system with arbitrary initial condition and evolve its dynamics. Then we use the signal $x(t)$ to drive the response $(y_r, z_r)$-subsystem by substituting the third variable $x_r$ with $x(t)$. It was shown that this $(y_r, z_r)$-subsystem will be forced to synchronize with the drive, so $y_r \to y$ and $z_r \to z$ with any initial conditions. Now, take the signal $y_r(t)$ and substitute it into the drive system. Because it can serve also as a driving variable, this signal also makes the $(x, z)$ subsystem to synchronize with $(x_r, z_r)$. Therefore $x \to x_r$ and the systems become synchronized. We have

$$\begin{aligned}\dot{x} &= \sigma(y-x), & & & \dot{x}_r &= \sigma(y_r - x_r),\\ \dot{y} &= x(\rho-z)-y, & &\xrightarrow{x(t)} & \dot{y}_r &= x(t)(\rho - z_r) - y_r,\\ \dot{z} &= xy-\beta z, & & & \dot{z}_r &= x(t)y_r - \beta z_r.\end{aligned} \tag{4.2}$$

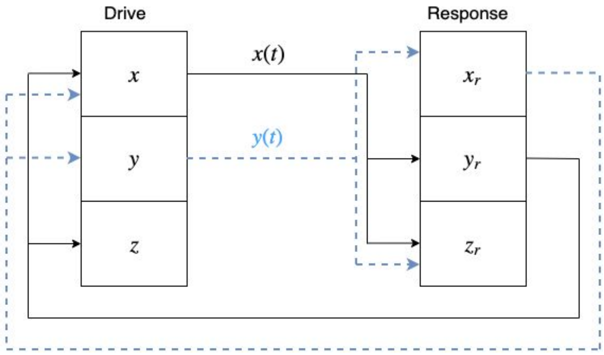
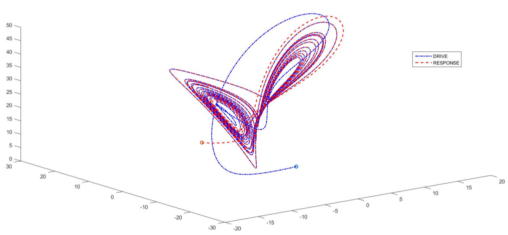


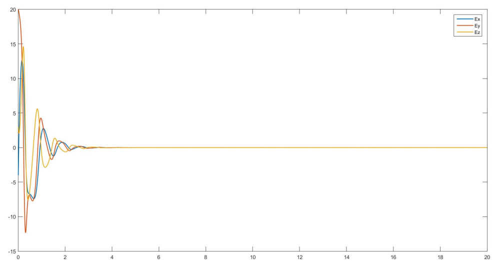

Figure 18: connection scheme for the Lorenz system for x-drive and y -drive

The scheme illustrating the nesting of the system is in picture 18

Similarly we could nest two systems using signal $y(t)$. Note, that some systems cannot be synchronized this way in all dimensions, e. g. in Rössler system presented in previous chapter there is only one possible driving variable, therefore two systems cannot be connected in both directions and it will not synchronize in all three dimensions.




(a) 3D phase portrait



(b) errors

Figure 19: two full-dimensional Lorenz systems synchronized

The conclusion of the ability two systems being connected gives an interesting result. Suppose that the sender and receiver both agree on the parameters of their systems. Then in order for two systems to synchronize, only one piece of information is needed and that is the driving signal. We are able to completely reconstruct the dynamics of the driving system in the response system without the knowledge of the initial conditions. This gives an interesting idea on how to use this property in the following applications.

## 4.1 Transmission and recovery of binary-valued bit streams

One of the potential applications is the ability to secretly send digital, binary valued messages. We use the property that if for some short time the driving signal is slightly changed and the response system desynchronizes, then the return back to the synchronous state occurs exponentially fast. In this case the basic idea is to modulate one of the transmitter coefficients in the driving system with the information-bearing waveform and then to transmit this chaotic drive signal to the receiver. The coefficient modulation then produces a synchronization error between the received signal and the signal that the receiver regenerated in his own system. The amplitude of error will depend on the modulation, we choose. If we then look into the error, we can detect the original message. We illustrate the process shown in the scheme 20 concretely below.

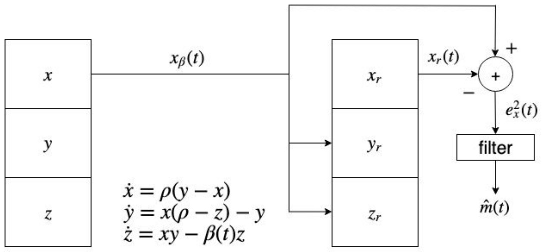



Figure 20: scheme and the driving equations

Suppose that the parameter $\beta$, which is normally a constant, is now time-dependent function $\beta = \beta(t)$. Let $m(t)$ be a binary-valued bit stream (Fig. 21). We embed it to the

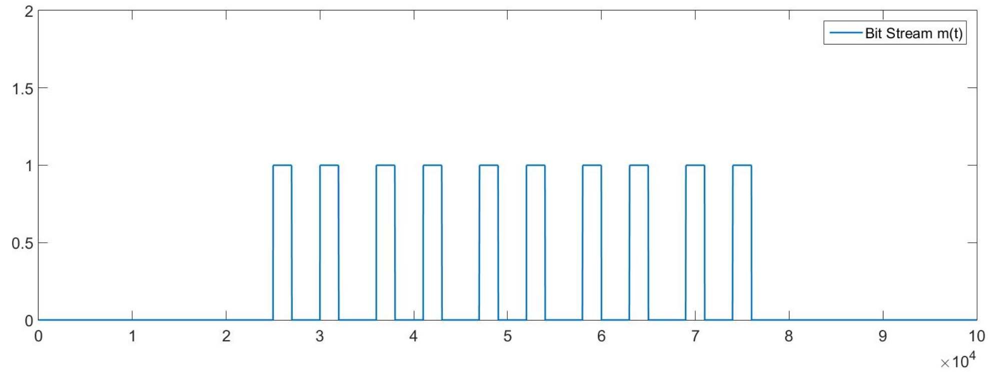



Figure 21: binary valued function $m(t)$

parameter function $\beta(t)$ and get a new function $s(t) = \beta(t) + km(t)$, where $k$ is arbitrary constant modulation parameter. Next, we evolve the transmitter system and generate drive signal $x_\beta(t)$, which we send via possibly unsecured channel.

In the 22 we can see the comparison of signals $x(t)$ and $x_\beta(t)$ created in the system with the constant parameter and with time-dependent parameter that carries the message. Notice, that in the moment of when the change occurred, the system completely changed its behaviour. That is good for the purpose of masking. The interceptor can only see the noise, but cannot read the values of the bit stream from it.

The receiver then evolves his system with the signal $x_\beta(t)$ as described in equation (4.2). We have

$$\begin{aligned}\dot{x}_r &= \sigma(y_r - x_r)\\ \dot{y}_r &= x_\beta(t)(\rho - z_r) - y_r\\ \dot{z}_r &= x_\beta(t)y_r - \beta z_r.\end{aligned}$$

The receiver system evolved its own version of $x_r(t)$. We examine the error $e_x(t) = x_\beta(t) - x_r(t)$. If there was no message modulated, the error would be equal to 0. This would be the result of complete synchronization of the systems. However, at times $t$, where the message is located, the receiver system desynchronizes, revealing the position of the one-valued part of stream (fig. 23). Then, for the zero-valued part it synchronizes again. It is then easy to apply a filter that transforms the square of the error function

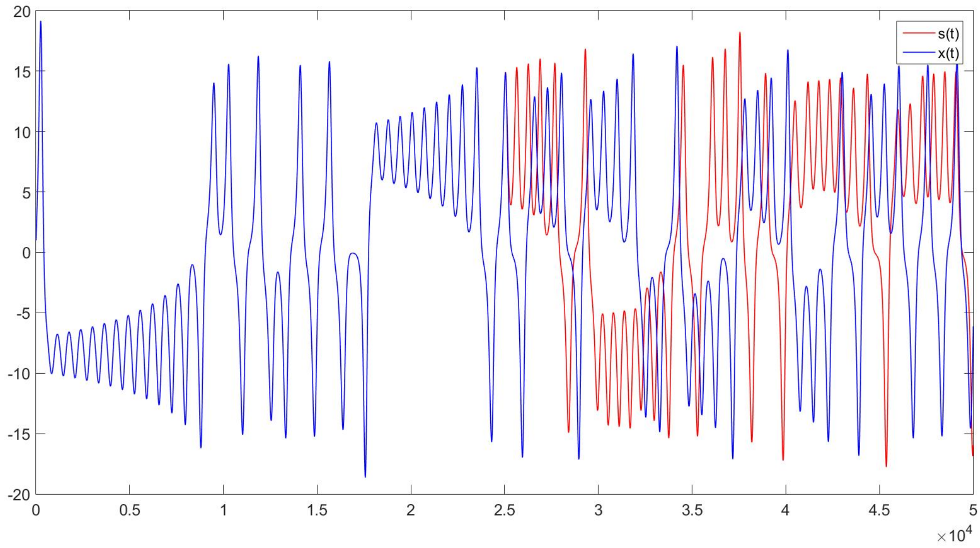



Figure 22: comparison of signal with and without message $m(t)$

$e_x^2(t)$ back to the bit stream $m(t)$. Remark, that we are able to recover the position of one-valued part exactly because the synchronization happens exponentially fast. If that did not occur, we could not determine where that part ends. The algorithm for this application can be found in section 6.6.

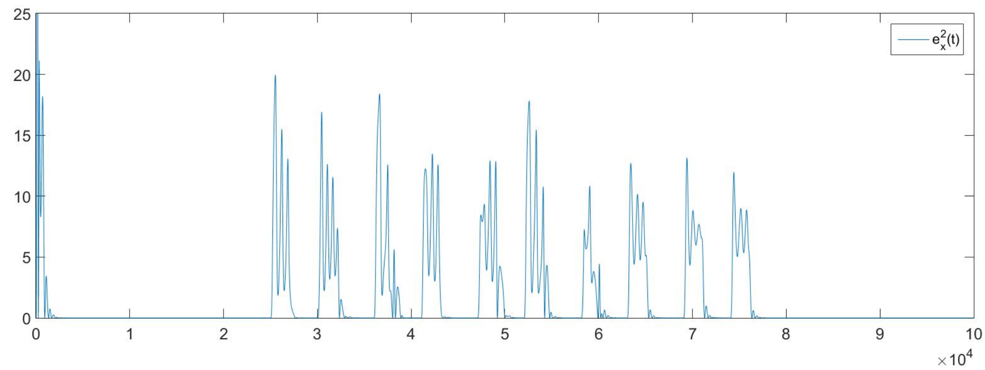



Figure 23: recovered bit stream

## 4.2 Using synchronized chaos to send secret messages

Another potential approach to communications applications is based on signal masking and recovery. In signal masking, a noiselike masking signal is added at the transmitter to the information-bearing signal $m(t)$ and at the receiver the masking is removed.

In previous section we modulated one of the parameters in the driving system and then we recovered the message from the position of the desynchronized part. The approach enables us to send and recover whole waveforms with information. The basic idea is to mask our message into a driving signal, send it via public channel and then use the signal to regenerate the masking signal at the receiver and subtract it to recover $m(t)$. All this can be done with the synchronizing receiver circuit since the ability to synchronize is robust, i.e., not highly sensitive to perturbations in the drive signal (as shown in [16]) and thus, can be done with the masked signal.

Suppose we have a waveform message $m(t, f)$ that we want to transmit, where $t$ is time and $f$ is frequency. We calculate with this extra parameter, because it was observed that it has direct impact on the ability to recover the signal. In transmitter we evolve dynamics of a driving signal $x(t)$ that will carry the message. We denote

$$s(t) = x(t) + m(t, f)$$

as the signal containing the information and we send it to the receiver.

We drive the response system by this signal in the $(y_r, z_r)$-subsystem and obtain a signal $x_r(t)$. As mentioned before, if the parameters of the message are well selected, the receiver system recognizes it as a noise and it will not change the its behaviour and synchronizes with the driver. Then the regenerated message is obtained by

$$\hat{m}(t) = s(t) - x_r(t) = m(t, f) + x(t) - x_r(t) \approx m(t, f).$$

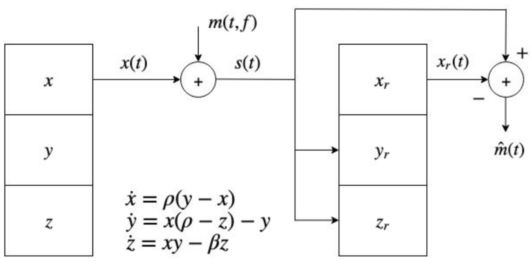



Figure 24: scheme for the message masking and driving equations

Using this approach, we run several experiments using algorithm 6.7, which will be now discussed. In the picture 25 it is shown how the added information changes the signal.

At time $t = 10$ the position of message in the signal can be observed, but the only way for the interceptor to read the message is to know the exact initial conditions of the drive system in order to reproduce uncorrupted signal and subtract the message. We have observed how the value of frequency affects the reconstruction of the output signal.

As a model example was chosen the function of the form

$$m(t, f) = \cos(2\pi t f).$$

We notice that the amplitude does not have much effect for this shape of message. It only does when the frequency is not well chosen. The comparison of how does the frequency $f$ affects the reconstructibility of the original message is in the picture 26.

It was demonstrated that the low frequencies on the interval $(0, 2)$ give bad result. For frequencies higher than 2 the ability to read the message rapidly increases and for very high frequencies the messages are almost identical. In [15] it was shown that the power spectrum of human speech is also overlapping with the chaotic masking spectrum of the Lorenz system. Therefore it is possible to recover a message containing human speech in reasonable quality.

Next attempt was to test this method to transmit a message containing an image. For this purpose 8-bit grayscale picture was used (fig. 27). Each pixel of this figure represents

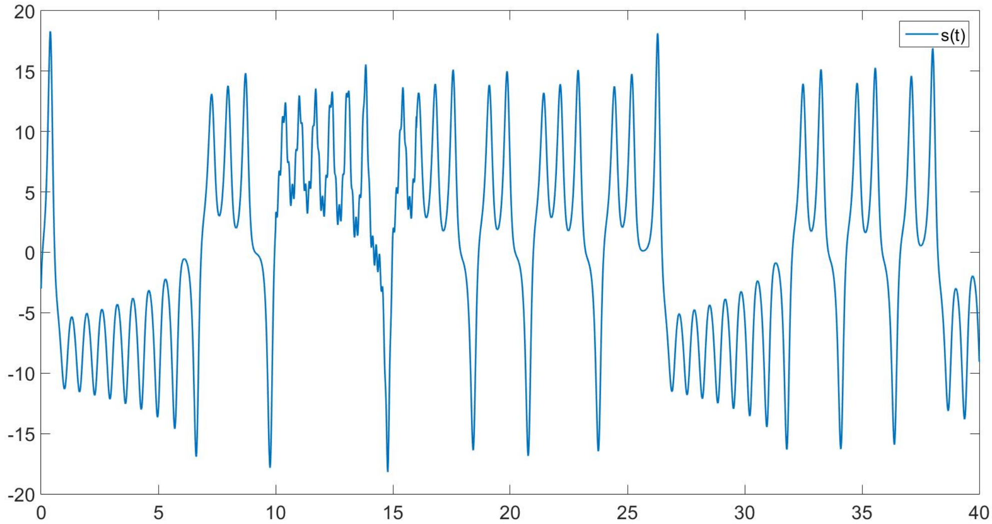
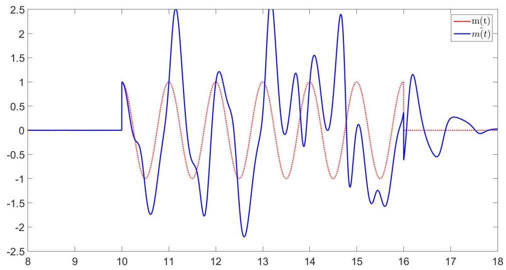
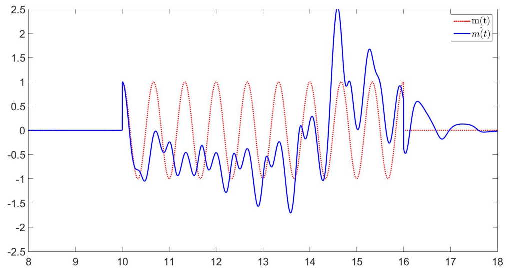


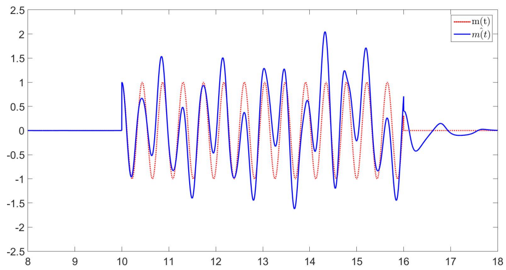

Figure 25: signal $s(t)$ carrying message



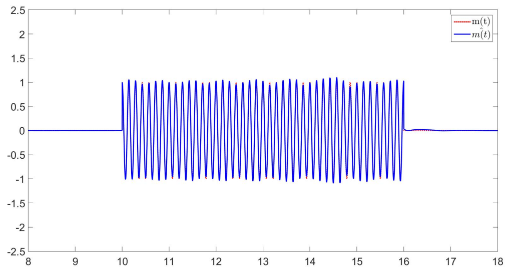

(a) $f = 1$




(b) $f = 1.5$




(c) $f = 2.3$




(d) $f = 7$

Figure 26: comparison of effect of frequencies on unmasking

a value on the integer interval $[0; 255]$. We obtained the wave-form of the picture by concatenation of the columns into the string of values interpolated at time $t$ (fig. 28). In this process, also the values of the pixels were downscaled and whole spectrum was shifted down by its mean. This gives better quality on the output.

We then generate the mask using drive signal and encrypt the image by it. In the picture 29 it is clear that the information is hidden in the mask and it is completely undetectable. Unlike from previous encryption with digital signal, if we look how the

Figure 27: original 100x100 8-bit grayscale picture used in experiment

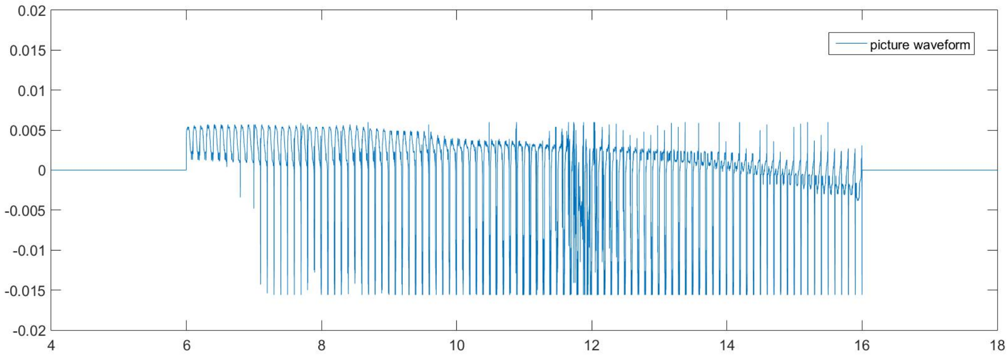



Figure 28: image transformed into waveform

message encryption changed the carrying signal, we will not see any difference (fig. 30. In the picture 32 there is a recovered waveform. It can be observed that the values of the pixels were stretched for about two times the distance between the maximal and minimal value. Therefore the contrast of the recovered picture is higher than in the original. In figure 31b we can see the recovered image. It is quite an exciting result that even though the recovered spectrum is corrupted, after its visualization it still bears the original piece of information and the picture is recognizable. The stripes in the recovered image are the side effect of the concatenation of the individual columns into one-dimensional array. One can notice, that each "local maximum" or "minimum" in the graph of recovered waveform corresponds to the light or the dark stripe in the picture.

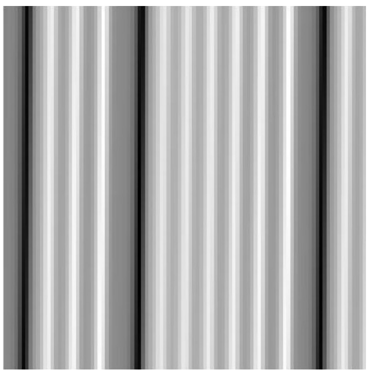


Figure 29: message hidden in the chaotic mask

To increase the quality of the decrypted picture there were attempts to decrease scale down the amplitude of the waveform, but as it was confirmed by the test with $\cos(t)$ function, it does not have much effect on it. Next step for obtaining better result could be

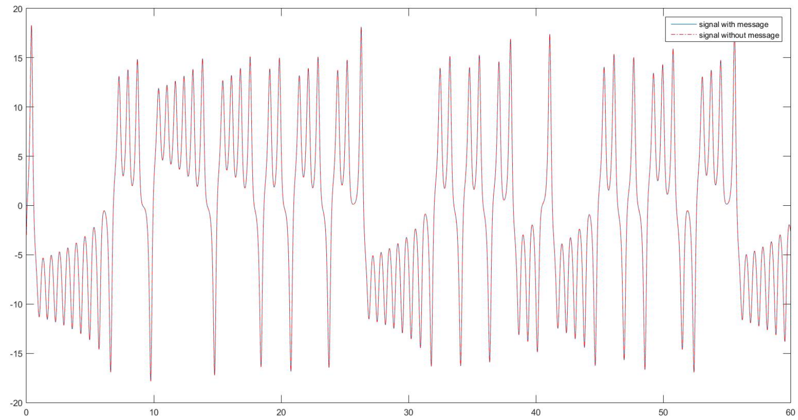
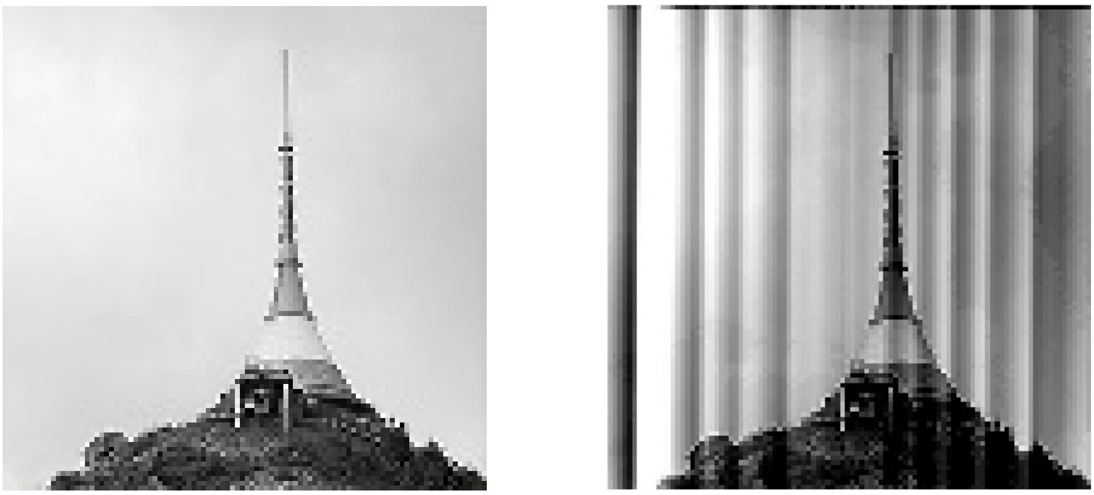



Figure 30: comparison of in the signals with and without the message



(a) (b)

Figure 31: comparison of the original and recovered message

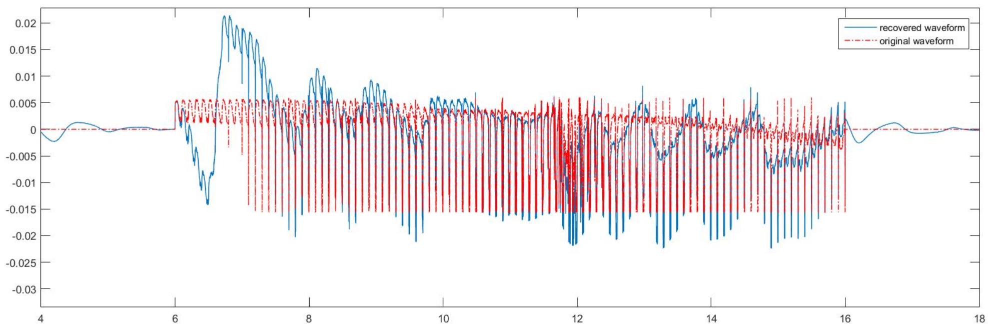



Figure 32: waveform of recovered image

increasing or reducing the frequency of the original waveform by interpolating the pixel values on a larger or a smaller time array. We leave this procedure as an open problem for expanding this work.

We discuss couple of remarks. It is important to say that this approach for masking the messages is just a basic illustration. For broader usage of this method a more sophisticated algorithm for encryption of the message into the mask needs to be used.

Next problem for this type of encryption is its speed. If the set of values we want to encrypt is large, we have to evolve the system for sufficient amount of time, so the drive signal could bear whole length of message. In our experiments, the computations took quite long time due to required precision, therefore it is not usable for everyday applications working real-time. On the other hand there were successful attempts by K. Cuomo [11] to build this masking device using RLC circuit run by the Lorenz equations.

Last, but not least, it is good to realize that this type of encryption will never be fully secure but only private. That means an interceptor can still decrypt the message if he is somehow able to acquire the initial conditions and parameters of the transmitter system. However, for less motivated interceptor the encryption is sufficient.

# 5 Conclusion

The goal of this work was to give an introduction to the theory of dynamical systems, deal with the notion of the chaotic behaviour and manifest it on specific examples of chaotic dynamical systems. Then, to give a summary of possible methods of synchronization of two chaotic dynamical systems and demonstrate it on concrete examples of two systems. These theoretical results were then supposed to be numerically confirmed by the algorithm created in MATLAB environment on the examples previously observed. The last goal was to demonstrate the applications of synchronization of chaotic dynamical systems inspired by the approach of Cuomo and Oppenheim in [11].

In the first part of the work the fundamental theory of chaotic dynamical systems was presented. The second part was devoted to the methods of synchronization of these types of systems. It was given a basic outline and then the two of them, complete synchronization and negative-feedback were deeper investigated. The complete synchronization was numerically performed on two Lorenz systems. The behaviour of the systems under this act was clarified and then the criteria for successful synchronization performance were generalized for arbitrary couple of systems. The synchronization by negative-feedback was applied and discussed on two Rössler systems. The conclusion of the chapter was that under certain conditions the complete synchronization is a special case of the negative-feedback synchronization.

In the last chapter, some possible applications of synchronization on the field of private communications was shown. We showed that the chaotic signal can serve for the encryption as a mask and then it is possible to use the properties of synchronized systems to reverse the process of masking to acquire the message that was encrypted.

This demonstration was done using the synchronization of Lorenz system for two types of data. Particularly we analysed the digital bit-stream and a grayscale picture. We showed that the quality of results is significantly influenced by the frequency of the data and we suggested possible approach to achieve better results. However, the observations showed that even though these applications can be put in work, a wider usage of these methods is not suitable due to the high computational demand and not being secure enough.

# References


[1] PERKO, Lawrence. Differential equations and dynamical systems. 3rd ed. New York: Springer, c2001. ISBN 03-879-5116-4.

[2] HIRSCH, Morris W., Stephen SMALE a Robert L. DEVANEY. Differential equations, dynamical systems, and an introduction to chaos. 3rd ed. Waltham, MA: Academic Press, c2013. ISBN 978-0-12-382010-5.

[3] STROGATZ, Steven H., Stephen SMALE a Robert L. DEVANEY. Nonlinear dynamics and Chaos: with applications to physics, biology, chemistry, and engineering. 3rd ed. Reading, Mass.: Addison-Wesley Pub., c1994. ISBN 02-015-4344-3.

[4] KHŮLOVÁ, Jitka: Stability and chaos in nonlinear dynamical systems: master's thesis. Brno: Brno University of Technology, Faculty of Mechanical Engineering, Institute of Mathematics, 2018. 74 p. Supervised by doc. RNDr. Jan Čermák, CSc.

[5] OBORNÁ, E. Stabilní a chaotické chování Lorenzova systému. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2015. 36 s. Vedoucí doc. RNDr. Jan Čermák, CSc

[6] TESAŘ, L. Nelineární dynamické systémy a chaos. Brno: Vysoké učení technické v Brně, Fakulta strojniho inženýrství, 2018.57 s. Vedoucí diplomové práce doc. Ing.Luděk Nechvátal, Ph.D.

[7] Tancjurová, J.: Analýza nelineárních dynamických systémů vykazující chaotické chování s atraktorem typu "dvojtý svitek", Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2017. 35 s. Vedoucí doc. Ing. Luděk Nechvátal, Ph.D.

[8] BOCCALETTI, S., J. KURTHS, G. OSIPOV, D.L. VALLADARES and C.S. ZHOU. The synchronization of chaotic systems. Physics Reports [online]. 2002, 366(1-2), 1-101 [cit. 2019-05-24]. DOI: 10.1016/S0370-1573(02)00137-0. ISSN 03701573. Available from: http://linkinghub.elsevier.com/retrieve/pii/S0370157302001370

[9] TRESSER, Charles, Patrick A. WORFOLK a Hyman BASS. Master-slave synchronization from the point of view of global dynamics. Chaos: An Interdisciplinary Journal of Nonlinear Science. 1995, 5(4), 693-699. DOI: 10.1063/1.166101. ISSN 1054-1500. Avalible from: http://aip.scitation.org/doi/10.1063/1.166101

[10] PECORA, Louis M. a Thomas L. CARROLL. Synchronization of chaotic systems. Chaos: An Interdisciplinary Journal of Nonlinear Science. 2015, 25(9). DOI: 10.1063/1.4917383. ISSN 1054-1500. Avalible from: http://aip.scitation.org/doi/10.1063/1.4917383

[11] CUOMO, Kevin M. a Alan V. OPPENHEIM. Circuit implementation of synchronized chaos with applications to communications. Physical Review Letters. 1993, 71(1), 65-68. DOI: 10.1103/PhysRevLett.71.65. ISSN 0031-9007. Avalible from: https://link.aps.org/doi/10.1103/PhysRevLett.71.65

[12] HE, Rong a P. G. VAIDYA. Analysis and synthesis of synchronous periodic and chaotic systems. Physical Review A. 1992, 46(12), 7387-7392. DOI: 10.1103/PhysRevA.46.7387. ISSN 1050-2947. Avalible from: https://link.aps.org/doi/10.1103/PhysRevA.46.7387

[13] PECORA, Louis M., Thomas L. CARROLL, Gregg A. JOHNSON, Douglas J. MAR a James F. HEAGY. Fundamentals of synchronization in chaotic systems, concepts, and applications. Chaos: An Interdisciplinary Journal of Nonlinear Science. 1997, 7(4), 520-543. DOI: 10.1063/1.166278. ISSN 1054-1500. Avalible from: http://aip.scitation.org/doi/10.1063/1.166278

[14] PECORA, Louis M., Thomas L. CARROLL, Gregg A. JOHNSON, Douglas J. MAR a James F. HEAGY. Synchronization of chaotic systems. Chaos: An Interdisciplinary Journal of Nonlinear Science. 2015, 25(9), 520-543. DOI: 10.1063/1.4917383. ISSN 1054-1500. Avalible from: http://aip.scitation.org/doi/10.1063/1.4917383

[15] CUOMO, K.M., A.V. OPPENHEIM, S.H. STROGATZ, Douglas J. MAR a James F. HEAGY. Synchronization of Lorenz-based chaotic circuits with applications to communications. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing. 2015, 40(10), 626-633. DOI: 10.1109/82.246163. ISSN 10577130. Avalible from: http://ieeexplore.ieee.org/document/246163/

[16] KOCAREV, Lj., K. S. HALLE, K. ECKERT, L. O. CHUA a U. PARLITZ. EXPERIMENTAL DEMONSTRATION OF SECURE COMMUNICATIONS VIA CHAOTIC SYNCHRONIZATION. International Journal of Bifurcation and Chaos. 1992, 02(03), 709-713. DOI: 10.1142/S0218127492000823. ISSN 0218-1274. Avalible from: http://www.worldscientific.com/doi/abs/10.1142/S0218127492000823

[17] Free picture avalible from: https://i.stack.imgur.com/duPPi.png

# 6 Apendix

The following sections contain the algorithms used for numerical computations created in MATLAB environment.

## 6.1 Algorithm for the phase portrait of Chua's circuit

```
function chua
%Function for the phase portairt of Chuas dynamical system
clear all
%constants
        r=1;
        ga = -1.14;
        gb = -0.71;
        c1 = 10.0d-09;
        c2 = 100.0d-09;
        bind = 20.0d-03;
        a = r*ga;        b = r*gb;
        alpha = 9.4;
        beta= 16;
%time step and initial condition
        tspan = 0:0.01:100;
        x10 = 0.11; x20 = 0.2; x30 = -0.3;
        y0 = [x10; x20; x30];

%computation
        [t,y] = ode45(@(t,x) f(t,x,a,b,alpha,beta),tspan,y0);
        x1=y(:,1); x2=y(:,2); x3=y(:,3);
%plots
   plot(x1,x2)
   title('XY plane')
   figure
   plot(x1,x3)
   title('XZ plane')
   figure
   plot(x2,x3)
   title('YZ plane')
   figure
   plot3(x1,x2,x3)
   title('3D')
   figure
    plot(t,x1)

 function dy = f(t,y,a,b,alpha,beta)
   x1 = y(1);     x2 = y(2);     x3 = y(3);
        hx=b*x1 + 0.5*(a-b)*(abs(x1+1.0)-abs(x1-1.0));
        dx1=alpha*(x2-x1-hx);
        dx2=x1-x2+x3;
        dx3=-beta*x2;
     dy = [dx1; dx2; dx3];
```

## 6.2 Algorithm for the synchronization of two Lorenz systems by CS

```
function LorSYNC_test
%%input parameters and settings
tStart = 0;
tEnd = 40;
tStep=0.001;
PP1 = [1, 2, 1];
PP3 = [1.01, 2, 1];
PP2 = [-8, 2];
opts = odeset('RelTol',1e-7,'AbsTol',1e-7);
%% DRIVE
[t,y] = ode45(@LorDRIVE,[tStart:tStep:tEnd],PP1,opts);
[t2,y2] = ode45(@LorDRIVE,[tStart:tStep:tEnd],PP3,opts);
%graph of DRIVE system
figure
plot3(y(:,1),y(:,2),y(:,3),y(1,1),y(1,2),y(1,3),'o')
driveX=(y(:,1));
t1=t;
%% RESPONSE
[T,yy] = ode45(@(tt,in) LorRESPONSE(tt,in,t1,driveX),[tStart:tStep:tEnd],PP2,opts);
%graph of RESPONSE system
figure
plot3(y(:,1),yy(:,1),yy(:,2),'b-.',y(:,1),y(:,2),y(:,3),'r--')
legend('DRIVE','RESPONSE')
figure
plot(t,(y(:,2)-yy(:,1)),t,(y(:,3)-yy(:,2)))
legend('Ey','Ez')
figure
plot(t,y(:,2),'b-',t,y2(:,2),'r--','LineWidth',1.5)
figure
plot(t,y(:,2)-y2(:,2),'b-','LineWidth',1.5)
figure
plot3(y(:,1),y(:,2),y(:,3),'b-.',y2(:,1),y2(:,2),y2(:,3),'r--','LineWidth',1.5)
legend('system IC:[1, 2, 1]','system IC:[1.01, 2, 1]')
```

```
end

function [out] = LorDRIVE( t,in )

%pocatecni podminky (ze syntaxu ODE45)
x = in(1);
y = in(2);
z = in(3);

%parametry
sigma  = 10;
ro     = 28;
beta   = 8/3;

%rovnice
xdot = sigma*(y-x);
ydot = x*(ro-z) - y;
zdot  = x*y-beta*z;

%vystup
out = [xdot ydot zdot]';

end

function [out] = LorRESPONSE( tt,in,t1,driveX)
driveX = interp1(t1,driveX,tt);
%ridici promena X

yr = in(1);
zr = in(2);


%původní nastavené hodnoty
sigma  = 10;
ro   = 28;
beta     = 8/3;

%rovnice
ydot =  driveX.*(ro-zr) - yr;
zdot =  driveX.*yr-beta*zr;

out = [ydot zdot]';

end
```

## 6.3 Algorithm for the synchronization of two Rössler systems by CS

```
function RosSYNC_test
%%input parameters and settings
tStart = 0;
tEnd = 200;
tStep=0.001;
PP1 = [-5, -4, 8];
PP3 = [-3, 4, 2];
PP2 = [10, 13];
opts = odeset('RelTol',1e-5,'AbsTol',1e-5);
%% DRIVE
[t,y] = ode45(@RosDRIVE,[tStart:tStep:tEnd],PP1,opts);
[t,y2] = ode45(@RosDRIVE,[tStart:tStep:tEnd],PP3,opts);
%graph of DRIVE system
figure
plot3(y(:,1),y(:,2),y(:,3),y(1,1),y(1,2),y(1,3),'o')
driveY=(y(:,2));
t1=t;
%% RESPONSE
[T,yy] = ode45(@(tt,in) RosRESPONSE(tt,in,t1,driveY),[tStart:tStep:tEnd],PP2,opts);
%graph of RESPONSE system
figure
plot3(yy(:,1),y(:,2),yy(:,2),'b-.',y(:,1),y(:,2),y(:,3),'r--')
legend('DRIVE','RESPONSE')
figure
plot(t,(y(:,1)-yy(:,1)),t,(y(:,3)-yy(:,2)))
legend('Ex','Ez')
figure
plot3(y(:,1),y(:,2),y(:,3),'LineWidth',1.5)
figure
plot(y(:,1),y(:,2),'LineWidth',1.5)
figure
plot(y(:,1),y(:,3),'LineWidth',1.5)
end
function [out] = RosDRIVE( t,in )
%Y DRIVE
%pocatecni podminky (ze syntaxu ODE45)
x = in(1);
y = in(2);
z = in(3);

%parametry

a = 0.2;
b = 0.2;
c = 9;
%soustava nelineárních ODR
```

```

xdot = -y-z;
ydot = x+a*y;
zdot = b+z*(x-c);

%rovnice

%vystup
out = [xdot ydot zdot]';

end

function [out] = RosRESPONSE( tt,in,t1,driveY)
driveY = interp1(t1,driveY,tt);
%ridici promena Y

xr = in(1);
zr = in(2);


%parametry

a = 0.2;
b = 0.2;
c = 9;

%rovnice
xdot = -driveY-zr;
zdot = b+zr*(xr-c);


out = [xdot zdot]';

end
```

## 6.4 Algorithm for the synchronization of two Rössler systems by NF

```
function rossler_discrete_NF
%function for synchronizazion by Negative Feedback for Rossler system
PP=[10 -8 12 1 2 -1];
n=1;
N=10000;
format long
h=0.01;
ALPHA=4;
X=zeros(1,N)';
Y=zeros(1,N)';
Z=zeros(1,N)';
Xr=zeros(1,N)';
Yr=zeros(1,N)';
Zr=zeros(1,N)';
SOL=zeros(N,7);
X(1)=PP(1);
Y(1)=PP(2);
Z(1)=PP(3);
Xr(1)=PP(4);
Yr(1)=PP(5);
Zr(1)=PP(6);
%system1
for n=1:N
    X(n+1)=X(n)+h*-(Y(n)+Z(n));
    Y(n+1)=Y(n)+h*(X(n)+0.2*Y(n));
    Z(n+1)=Z(n)+h*(0.2+Z(n)*(X(n)-9));
end;
for n=1:N
    Xr(n+1)=Xr(n)+h*-(Yr(n)+Zr(n)+ALPHA*(Xr(n)-X(n)));
    Yr(n+1)=Yr(n)+h*(Xr(n)+0.2*Yr(n));
    Zr(n+1)=Zr(n)+h*(0.2+Zr(n)*(Xr(n)-9));
end
for n=1:N
    SOL(n,:)=[n X(n) Y(n) Z(n) Xr(n) Yr(n) Zr(n)];
end
figure();
plot3(SOL(:,2),SOL(:,3),SOL(:,4),'b-',SOL(:,5),SOL(:,6),SOL(:,7),'r-.');
legend('drive','response')
end
```

## 6.5 Algorithm for the comparison of CS and NF method

```
function comparison_CS_NF
PP=[2 3 -1 -5 -1 2];
n=1;
N=10^5;
format long
h=0.01;
%coupling coefficient
ALPHA=100;
X=zeros(1,N)';
Y=zeros(1,N)';
Z=zeros(1,N)';
Xr=zeros(1,N)';
```

```
Xr2=zeros(1,N)';
Yr=zeros(1,N)';
Zr=zeros(1,N)';
Zr2=zeros(1,N)';
SOL=zeros(N,10);
X(1)=PP(1);
Y(1)=PP(2);
Z(1)=PP(3);
Xr(1)=PP(4);
Yr(1)=PP(5);
Zr(1)=PP(6);
Xr2(1)=PP(4);
Zr2(1)=PP(6);
for n=1:N
    X(n+1)=X(n)+h*-(Y(n)+Z(n));
    Y(n+1)=Y(n)+h*(X(n)+0.2*Y(n));
    Z(n+1)=Z(n)+h*(0.2+Z(n)*(X(n)-9));
end;
for n=1:N
    Xr(n+1)=Xr(n)+h*-(Yr(n)+Zr(n));
    Yr(n+1)=Yr(n)+h*(Xr(n)+0.2*Yr(n)+ALPHA*(Y(n)-Yr(n)));
    Zr(n+1)=Zr(n)+h*(0.2+Zr(n)*(Xr(n)-9));
end
for n=1:N
    Xr2(n+1)=Xr2(n)+h*-(Y(n)+Zr2(n));
    Zr2(n+1)=Zr2(n)+h*(0.2+Zr2(n)*(Xr2(n)-9));
end
for n=1:N
    SOL(n,:)=[n X(n) Y(n) Z(n) Xr(n) Yr(n) Zr(n) Xr2(n) Y(n) Zr2(n)];
end
figure()
plot(SOL(:,1),SOL(:,2),'-',SOL(:,1),SOL(:,5),'--','LineWidth',2);
legend('drive x(n)','response x_r(n)'),
set(gca,'FontSize',30)
figure()
plot(SOL(:,1),SOL(:,4),'-',SOL(:,1),SOL(:,7),'--','LineWidth',2);
legend('drive z(n)','response z_r(n)')
set(gca,'FontSize',30)
figure();
plot3(SOL(:,2),SOL(:,3),SOL(:,4),'b-',SOL(:,5),SOL(:,6),SOL(:,7),'r-.','LineWidth',2);
legend('drive','response')
set(gca,'FontSize',30)
figure();
plot(SOL(:,1),SOL(:,2)-SOL(:,5),SOL(:,1),SOL(:,3)-SOL(:,6),SOL(:,1),SOL(:,4)-SOL(:,7),'LineWidth',2);
legend('e_x','e_y','e_z')
set(gca,'FontSize',30)
figure();
plot3(SOL(:,2),SOL(:,3),SOL(:,4),'b-',SOL(:,5),SOL(:,6),SOL(:,7),'r-.','LineWidth',2);
legend('drive','response')
figure();
plot(SOL(:,1),abs(SOL(:,2)-SOL(:,5)),'LineWidth',2)
figure();
plot3(SOL(:,2),SOL(:,3),SOL(:,4),'b-',SOL(:,8),SOL(:,9),SOL(:,10),'r-.','LineWidth',2);
legend('drive','response')
figure();
plot(SOL(:,1),(SOL(:,5)-SOL(:,8)),SOL(:,1),(SOL(:,6)-SOL(:,9)),SOL(:,1),(SOL(:,7)-SOL(:,10)),'LineWidth',2)
legend('e_{x_1}-e_x_2','e_y_1-e_y_2','e_z_1-e_z_2')
set(gca,'FontSize',30)
end
```

## 6.6 Algorithm for the transmission and recovery of a bit-stream

```
function bitstream_discrete
%function to generate bitstream, encrypt it to BETA(t)
%and decrypt to see the square error e_x
PP=[1 6 3];
format long
N=10^5;
h=0.001;
int=(linspace(0,N,N))';
X=zeros(1,N)';
Y=zeros(1,N)';
Z=zeros(1,N)';
X(1)=PP(1);
Y(1)=PP(2);
Z(1)=PP(3);
X2=zeros(1,N)';
Y2=zeros(1,N)';
Z2=zeros(1,N)';
X2(1)=PP(1);
Y2(1)=PP(2);
Z2(1)=PP(3);
n=500; %zeros after n digits
k=5; %repear k times
%%GENERATE THE STREAM
%stream=repmat(reshape([ones(n,1) ones(n,1) zeros(n,1)  zeros(n,1)
%                        zeros(n,1) ones(n,1) ones(n,1) zeros(n,1)
%                        zeros(n,1) zeros(n,1)],[10*n 1]),k,1);
%Lstream=length(stream);
%Beta=[zeros(N/4,1);stream;zeros(N-Lstream-N/4,1)];
Beta=(8/3)*ones(N,1);
%plot the generated bit stream
figure
plot(int,Beta,'LineWidth',1.5)
legend('Bit Stream m(t)')
for n=1:N-1
```

```
    X(n+1)=X(n)+h*(10*(Y(n)-X(n)));
    Y(n+1)=Y(n)+h*(-X(n)*Z(n)+28*X(n)-Y(n));
    Z(n+1)=Z(n)+h*(X(n)*Y(n)-Beta(n)*Z(n));
end
for n=1:N-1
    X2(n+1)=X2(n)+h*(10*(Y2(n)-X2(n)));
    Y2(n+1)=Y2(n)+h*(-X2(n)*Z2(n)+28*X2(n)-Y2(n));
    Z2(n+1)=Z2(n)+h*(X2(n)*Y2(n)-(8/3)*Z2(n));
end
A=N/2;
%plot difference of signals
figure
plot(int(1:(A)),X(1:(A)),'r',int(1:(A)),X2(1:(A)),'b','LineWidth',1)
legend('s(t)','x(t)');
%%
PP2=[4 -8 -5];
Xr=zeros(1,N)';
Yr=zeros(1,N)';
Zr=zeros(1,N)';
Xr(1)=PP2(1);
Yr(1)=PP2(2);
Zr(1)=PP2(3);
for n=1:N-1
    Xr(n+1)=Xr(n)+h*(10*(Yr(n)-Xr(n)));
    Yr(n+1)=Yr(n)+h*(-X(n)*Zr(n)+28*X(n)-Yr(n));
    Zr(n+1)=Zr(n)+h*(X(n)*Yr(n)-(8/3)*Zr(n));
end
figure();
plot3(Xr,Yr,Zr,Xr(1),Yr(1),Zr(1),'o',X,Y,Z,'-.',X(1),Y(1),Z(1),'o')
%Square Error
plot(int,(X-Xr).^2)
legend('e_x^2(t)');
end
```

## 6.7 Algorithm for the encryption and decryption of an image

```
function image_cypher
%%input parameters and settings
tStart = 0;
tEnd = 60; %delitelne 4
tStep=0.001;
PP1 = [-3, 5, 7];
PP2 = [1, 1, 1];
opts = odeset('RelTol',1e-5,'AbsTol',1e-5);
%% DRIVE GENERATION
[t,y] = ode45(@LorDRIVE,[tStart:tStep:tEnd],PP1,opts);
%graph of DRIVE system
%figure
%plot3(y(:,1),y(:,2),y(:,3),y(1,1),y(1,2),y(1,3),'o')
driveX=(y(:,1));
t1=t;
%% MESSAGE
%modulace
M = double(imread('jested.bmp'));
figure
imshow(M,[0 255])
[p,q]=size(M);
M = reshape(M,[p*q 1]);
prum=round(mean(M));
M=(M-prum)/10000;
Mask=driveX(6001:16000)+M;
Mask2=reshape(Mask, [100 100]);
figure
imshow(Mask2,[min(Mask) max(Mask)]);
%přidávání nul - NEFUNGUJE
m = 1; % po kolika ta nula
k = 0; %number of zeros
n = length(M);
OUTh = reshape([reshape(M,m,[]);zeros(k,n/m)],[],1);
OUT = [zeros(6000,1);OUTh;zeros(length(t)-length(OUTh)-6000,1)];
%test funkce
%fceE=@(t) .01*cos(20*(t)*pi); % FUNGUJE fceE=@(t) .01*cos(10*(t)*pi) na poli o t=0..500
%tau=[tStart:tStep:tEnd];
%fceE=fceE(t);
%newTest=[zeros(10000,1);fceE(1:301);zeros(29700,1)];
%fceE=fceE';
figure
plot(t,OUT,'b','LineWidth',0.5)
ylim([-0.02 0.02])
xlim([6 16])
set(gca,'FontSize',20)
%informaci davam na k-tou pozici
%fceElong=zeros(param*hust*k,1);
%fceElong=zeros(length(IMG)*k,1);
%fceElong(1:k:end,1)=fceE;
%% MESSAGE MASKING
K=(length(t)-1)/5; %position in mask
St=driveX+OUT; % [zeros(K,1);fceElong;zeros(length(driveX)-length(fceElong)-K,1)]; %message masked
%figure %graf jak vypada signal **
plot(t,St)
figure %graf umisteni zpravy ve grafu ***
plot(t,St-driveX)
%% DECRYPTION
t1D=t;
[TD,yyD] = ode45(@(ttD,in) LorRESPONSE_St(ttD,in,t1D,St),[tStart:tStep:tEnd],PP2,opts);

```

```
%figure %errory
%plot(t,y(:,1)-yyD(:,1),t,y(:,2)-yyD(:,2),t,y(:,3)-yyD(:,3))
%legend('Ex','Ey','Ez')

fceDhelp=St-yyD(:,1);
figure %graf po odecteni, chci dostat ***
plot(t,fceDhelp)
mess=fceDhelp(6000:(6000+(length(OUTh)-1)));
mess=reshape(mess,[p q]);
mess2=mess*10000+prum;
figure
imshow(mess2,[0 255]);

%fceD2=fceDhelp(K:k:(K+length(fceElong)-1));
%figure %graf zpravy, chci dostat **
%plot(tau,fceD2)
%fceD=St(K:k:K+length(tau)-1)-yyD(K:k:K+length(tau)-1,1);
%figure
%plot(tau,fceD)
end
function [out] = LorDRIVE( t,in )

%pocatecni podminky (ze syntaxu ODE45)
x = in(1);
y = in(2);
z = in(3);

%parametry
sigma  = 10;
ro    = 28;
beta  = 8/3;

%rovnice
xdot = sigma*(y-x);
ydot = x*(ro-z) - y;
zdot  = x*y-beta*z;

%vystup
out = [xdot ydot zdot]';

end

function [out] = LorRESPONSE_St(ttD,in,t1D,St)
%message St
St= interp1(t1D,St,ttD);

xr = in(1);
yr = in(2);
zr = in(3);

%parametry
sigma  = 10;
ro   = 28;
beta     = 8/3;

%rovnice
xdot =  sigma*(yr-xr);
ydot =  St.*(ro-zr) - yr;
zdot =  St.*yr-beta*zr;

%vystup
out = [xdot ydot zdot]';
end
```