VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF RADIO ELECTRONICS

POKROČILÉ ALGORITMY PRO ANALÝZY DATOVÝCH SEKVENCÍ V MATLABU

ADVANCED ALGORITHMS FOR THE ANALYSIS OF DATA SEQUENCES IN MATLAB

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE Bc. Tomáš Gotthans

AUTHOR

VEDOUCÍ PRÁCE Ing. Jiří Petržela, Ph.D.

SUPERVISOR

BRNO, 2010

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta elektrotechniky a komunikačních technologií
Ústav radioelektroniky

Diplomová práce

magisterský navazující studijní obor
Elektronika a sdělovací technika

Student: Bc. Tomáš Götthans ***ID:*** 78218
Ročník: 2 ***Akademický rok:*** 2009/10

NÁZEV TÉMATU:

Pokročilé algoritmy analýzy datových sekvencí v Matlabu

POKYNY PRO VYPRACOVÁNÍ:

Detailně se seznamte s možnostmi programu Matlab z hlediska analýzy autonomních deterministických dynamických systémů. Navrhněte algoritmus umožňující specifikovat chování systému na základě znalosti příslušných diferenciálních rovnic.

Vytvořte v programu Matlab program pro výpočet Ljapunovových exponentů na základě vstupní datové sekvence. Ověřte správnou funkci programu dostatečným množstvím odlišných simulačních procesů.

Využijte výstupů z předchozích etap projektu k vytvoření programu pro vyhledávání řešení dynamického systému citlivého na počáteční podmínky variací jeho parametrů. Využijte přitom vhodnou optimalizační metodu.

DOPORUČENÁ LITERATURA:

[1] CARROL, T., PECORA, L. Nonlinear Dynamics in Cirucits. World Scientific Publishing, 1995.

[2] WYK, M. A., STEEB, W. H. Chaos in Electronics. Kluwer Academic Publishers, 1997.

[3] SPROTT, J. C. Chaos and Time Series Analysis. Oxford University Press, 2003.

Termín zadání: 8.2.2010 ***Termín odevzdání:*** 21.5.2010

Vedoucí práce: Ing. Jiří Petržela, Ph.D.
Konzultanti diplomové práce:

Raida

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ Fakulta elektrotechniky a komunikačních technologií Ústav radioelektroniky

prof. Dr. Ing. Zbyněk Raida
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Bc. Tomáš Götthans

Bytem: Bří. Mrštíků 1412/6, Šlapanice, 66451

Narozen/a (datum a místo): 26. ledna 1985 v Brně

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií

se sídlem Údolní 53, Brno, 602 00

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

prof. Dr. Ing. Zbyněk Raida, předseda rady oboru Elektronika a sdělovací technika

(dále jen „nabyvatel“)

Čl. 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

   ☐ disertační práce

   ☒ diplomová práce

   ☐ bakalářská práce

   ☐ jiná práce, jejíž druh je specifikován jako .... (dále jen VŠKP nebo dílo)

Název VŠKP: Pokročilé algoritmy analýzy datových sekvencí v Matlabu

Vedoucí/ školitel VŠKP: Ing. Jiří Petržela, Ph.D.

Ústav: Ústav radioelektroniky

Datum obhajoby VŠKP: __________________

VŠKP odevzdal autor nabyvateli*:

☒ v tištěné formě – počet exemplářů: 2

☒ v elektronické formě – počet exemplářů: 2

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.

3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.

4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

---

* hodící se zaškrtněte

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizovaní výpisů, opisů a rozmnoženin.

2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.

3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti

   - [x] ihned po uzavření této smlouvy
   - [ ] 1 rok po uzavření této smlouvy
   - [ ] 3 roky po uzavření této smlouvy
   - [ ] 5 let po uzavření této smlouvy
   - [ ] 10 let po uzavření této smlouvy (z důvodu utajení v něm obsažených informací)

4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.

2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.

3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.

4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 21. května 2010

………………………………………..	…………………………………………
Nabyvatel	Autor

ABSTRAKT

Cílem této práce je se seznámení s možnostmi programu Matlab z hlediska detailní analýzy deterministických dynamických systémů. Jedná se především o analýzu časové posloupnosti a o nalezení Lyapunových exponentů. Dalším cílem je navrhnout algoritmus umožňující specifikovat chování systému na základě znalosti příslušných diferenciálních rovnic. To znamená, nalezení chaotických systémů.

KLÍČOVÁ SLOVA

Chaos, atraktor, dynamické systémy, Lyapunovy exponenty, genetický algoritmus, optimalizace, multikriteriální optimalizace, fraktál, fraktální dimenze, PSO, metoda roje částic, časová posloupnost

ABSTRACT

This work aims to familiarize with the possibilities of Matlab in terms of detailed analysis of deterministic dynamical systems. This is essentially a analysis of time series and finding Lyapunov exponents. Another objective is to design an algorithm allowing to specify the system behavior based on knowledge of the relevant differential equations. That means finding chaotic systems.

KEYWORDS

Chaos, attractor, dynamical systems, Lyapunov exponents, genetic algorithm, optimization, multicriteria optimization, fractal, fractal dimension, PSO, particle swarm method, timing

Bibliografická citace

GÖTTHANS, T. *Pokročilé algoritmy analýzy datových sekvencí v Matlabu.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 55 s. Vedoucí semestrální práce Ing. Jiří Petržela, Ph.D.

Prohlášení

Prohlašuji, že svůj semestrální projekt na téma Pokročilé algoritmy analýzy datových sekvencí v Matlabu jsem vypracoval samostatně pod vedením vedoucího semestrálního projektu a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedeného semestrálního projektu dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 21. května 2010 ....

podpis autora

Poděkování

Děkuji vedoucímu semestrálního projektu Ing. Jiřímu Petrželovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mého semestrálního projektu.

V Brně dne 21. května 2010 ....

podpis autora

LIST OF PICTURES

CONTENTS

1 INTRODUCTION TO NONLINEAR DYNAMICS

The essence of science is the assumption that all the experiments are carried out predictable and repeatable. It was therefore surprising when a simple deterministic systems (under certain circumstances), were not predictable or repeatable. Instead, it showed a phenomenon known as chaos. Under the term deterministic chaos we understand a system which is extremely sensitive to initial conditions. As a result of this sensitivity to the behavior of these physical systems exhibiting chaos, seems to be random, although the model system is the 'deterministic' in the sense that it is well defined and contains no random parameters. The roots of chaos theory can be dated back to 1900, Henri Poincare in studies on the problem of movement of objects to 3 mutual gravitational force, the problem of three objects. Poincare discovered that there may be orbits which are non-periodical, and which are not constantly increasing or close to a fixed point. Later studies, also on the topic of nonlinear differential equations, were implemented GD Birkhoff, AN Kolmogorov, ML Cartwright, J.E. Littlewood, and Stephen Smale. In addition Smale, perhaps the first pure mathematician studied nonlinear dynamics, these studies were all directly inspired by physics: the problem of three bodies in the case of Birkhoff, turbulence and astronomical problems in the case Kolmogorov, and radio technology in the case Cartwright and Littlewood. Although the chaotic movement of planets has been observed, experimenters encountered turbulence in the movement of liquids and non-periodical oscillations in the radio circuit, without the support of the theory that would explain their observations.

Chaos theory quickly progressed in the middle of last century, when it became clear to some scientists that linear theory, the prevailing theory of systems in this period, simply cannot explain the observed behavior in certain experiments, such as the logistic map. The main catalyst for the development of chaos theory was the electronic computer. Most of the mathematical theory of chaos involves a simple re-iteration, the development is impractical to test manually. Electronic computer research in such systems facilitate highly. One of the first electronic computers, ENIAC, was used to study simple models of weather forecasts. One of the first pioneer this theory was Edward Lorenz, whose interest in chaos arose randomly during his work on weather prediction in 1961. Lorenz used the computer Royal McBee LPG-30 to calculate its model simulating weather. I see again the sequence, and to save time, the simulation began brokering. It is printed data from previous simulation and is entered as input data to your model.

To his surprise the weather forecast was quite different than its original model. Lorenz examined why, and discovered the cause of their group. Report rounded variable to 3 decimal places, while the computer worked with 5 decimal places. This difference is small and should not have to deal with practical impact. However, Lorenz discovered that small changes in initial conditions lead to large changes in output in the long term.

The concept of chaos, as used in mathematics, was established applied mathematician James A. Yorke. Moore Act and the availability of cheaper computers have extended the possibility of examining the theory of chaos. Currently, continuing very actively exploring this theory.

Systems that exhibit mathematical chaos are in a sense, complex manner. This is the meaning of the word in mathematics and physics in non-compliance with the usual understanding of the word chaos as total disarray. The origin of this word can be found in Greek mythology.

2 EXAMPLES OF DYNAMICAL SYSTEMS

2.1 Astronomical objects

For example, the movement of astronomical bodies described by the universal law of gravity (F = G * m1 * m2 / r ^ 2) and the second movement Newton law (F = ma). It is an inverse quadratic dependence on the separation of forces of matter is non-linearity, which allows for chaos. However, it does not guarantee it. It is true that the movements of bodies are among the most predictable processes in nature. Planetary movements have been comprehensively described Johannes Kepler many years ago the analysis of astronomical observations of Tycho de Brahe. The problem is, as we have already indicated in the introduction, for example, when the planet will orbit the binary stars (the three body problem). Track this planet is unpredictable. An example of one of the many trajectories is shown in the picture.

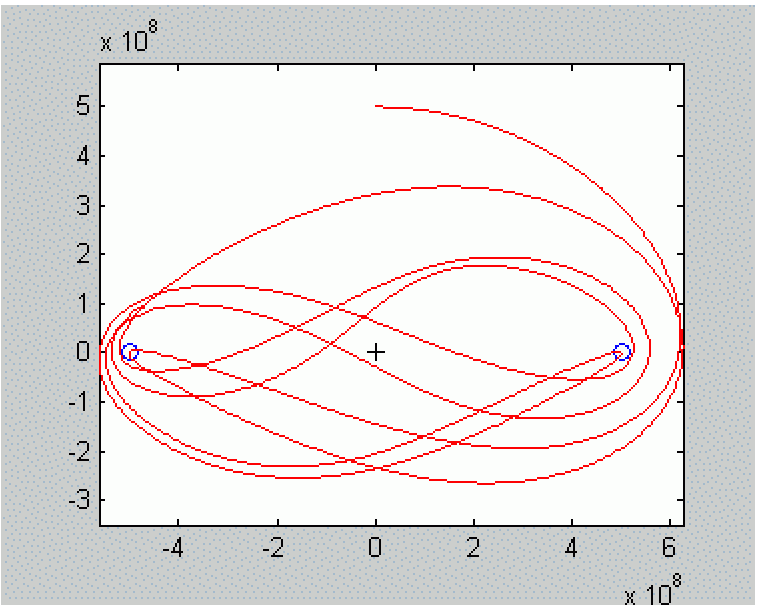


Fig. 1. Planet orbiting a binary star

2.2 Liquids

Another example is the movement of fluids (gas, or plasma). In general, the liquid consisting of a large (and infinitely large) amount of molecules to each other at each other operates. Each neighboring molecule, therefore, responds to its surroundings. It is therefore evident that such a trivial mixing of fluids is a nonlinear dynamic phenomenon. In terms of prediction of mixing is a chaotic phenomenon. It is known that divided the flow of liquids in the laminar and turbulent (chaotic). However, it is difficult to identify when and under what circumstances, turbulent flow occurs.

2.3 Non-physical systems

Rhythmic changes in blood pressure, heart and other cardiovascular ratio values indicate the importance of dynamic aspects in the understanding of cardiovascular rhythms. Several studies point to the fact that some cardiac arrhythmias are an example of chaos. This is important because it can help determine treatment. For example, cardiac arrhythmias, Atrial fibrillation, bradycardia, cardiac acceleration activity, hollow core and other arrhythmias.

Ventricular arrhythmias, both chambers fibrillation and ventricular tachycardia, the most serious of the diseases. These arrhythmias cause many deaths. Cardiac instability can be understood as a spontaneous asynchronous download of cardiac muscle fibers. Atrial chambers become spontaneous and irregular heart rhythm disorder. This can quickly proceed to the heart rhythm becomes incompatible with life.

Cardiac rhythm is one of the best indicators arrhythmic events and may lead to sudden death after myocardial infarction. Cardiac rhythm is partly controlled by an autonomous nerve system. Autonomous System (not to affect the will of man) - management of autonomous function of the nature of reflective, independent of our consciousness. The autonomic nervous system is divided into subsystems, compassionate (SNS) and parasympathetic (PNS) nervous systems. Short-term variability is mediated parasympathetic nervous system, while long-term variability of the two: sympathetic (SNS) and parasympathetic (PNS) nervous system. As you know, heart rate may vary even in the absence of physical or mental pressure.

Several studies demonstrate the relationship between cardiac arrhythmias and chaos. This is related to the deterministic characteristics of any of these arrhythmias. Clinical arrhythmia has the greatest potential for therapeutic applications of chaos theory to non-periodic tachycardia, including atrium and ventricle fibrillation. Such an approach in the evaluation could be implemented to promoters. This could be avoided, for example fibrillation chambers.

There are already some interesting comparisons between the dynamic characteristics of healthy individuals and patients with high risk of sudden cardiac fibrillation. Hearts with a high risk of sudden cardiac arrhythmia course shows chaotic signals. These methods could pose an important diagnostic tool for clinical purposes.

We can simplify the situation so that the heart is a type of oscillator. Question is, if the oscillator is valid for the nonlinear dynamic equations. The problem is, however, that these equations contain many state variables - there are many factors affecting heart rhythm. In clinical conditions of the situation but we can simplify and reduce to fewer variables. There is a chance that it will to some extent (i.e. with a short time and limited initial conditions) to predict heart rhythm. Such a device would become a very important tool in the operating theater. Surgeon should be able to prepare a situation with time.

3 SENSITIVITY TO INITIAL CONDITIONS

Sensitivity to initial conditions means that two close trajectories in phase space with increasing time of launching. In other words, a small change in initial conditions leads over time to a very different outcome. The system behaves identically, only when the same initial conditions. The combined sensitivity is the so-called Butterfly. The weather can be so sensitive that just sweep butterfly wings on one side and on the other planets (for a longer period of time) can cause tornadoes. We will test the sensitivity to initial conditions of the program Matlab. Consider Lorenzo equation is a solution for different initial conditions. Then we compare the time course of both solutions.

The equation describing Lorenz attractor: (1.)

$$\frac{dx}{dt} = \sigma(y - x)$$

$$\frac{dy}{dt} = x(r - z) - y$$

$$\frac{dz}{dt} = xy - bz$$

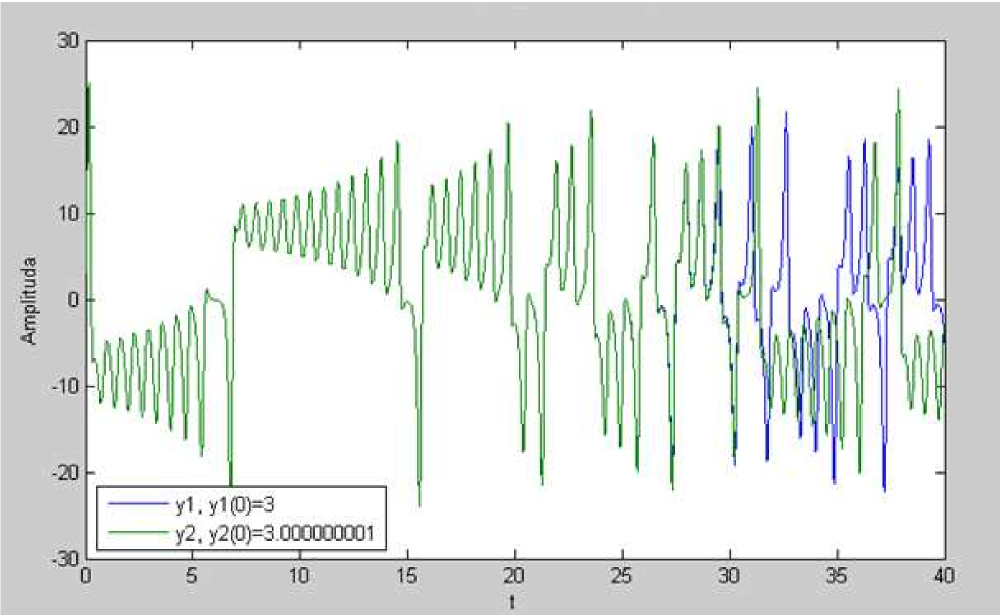


Fig. 2. Different results with different initial conditions

Coefficients Lorenz equations (2.3.1) are: σ = 10, b = 8 / 3 and r = 28. Initial conditions: x1 (0) = 8, y1 (0) = 3, z1 (0) = 4 and x2 (0) = 8, y2 (0) = 3.000000001 ,Z2(0)=4

The graph shows that even if the difference between the initial conditions is only 0.000000001, during a longer period of time leads to significant deviations.

Note, that sensitivity on the initial conditions isn't a privilege of nonlinear dynamics. It also happens in linear systems. For example, take the following time-discrete dynamical system

$$x_{n+1} = 2x_n.$$

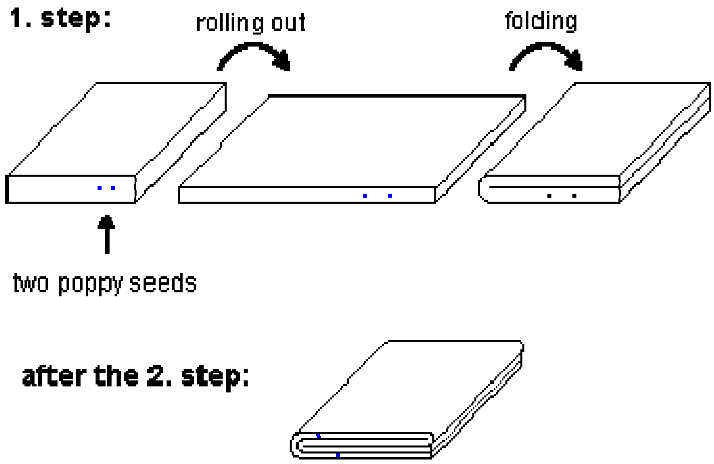


Fig. 3. Stretching and folding [9]

The distance between two different solutions increases by the factor two in each time step. But of course, this is not deterministic chaos. It is a kind of trajectory explosion. Sensitivity on the initial conditions leads to chaos only if the trajectories are bound. That is, the system cannot blow up to infinity.

With linear dynamics, you can have either sensitivity on the initial conditions or bound trajectories, but not both. With nonlinearities, you can have both. The figure shows why this is possible. Imagine the phase space is a piece of dough from which you want to make flaky pastry for croissants: You roll it out and fold it. Usually you repeat this step a few times. Suppose at the beginning two “poppy-seeds”, are located next to each other. Each time the dough is rolled out the distance between the seeds increases. Eventually, this increase is stopped when the seeds are on different sides of the folding line. At that moment the distance changes unpredictable. Thus, stretching and folding are responsible for deterministic chaos. And there is no folding without nonlinearities [9].

4 STRANGE ATTRACTORS

More often than not, orbit trajectory of chaotic system, will be painted in a small area of the state space. With the development time, the trajectory will vary deterministically, but you can not accurately predict. Movement trajectories of the object fractal called strange attractor. Strange attractors are often a manifestation of chaos. They are called "strange" because of its fractal structure. It is also possible to use the term "chaotic attractor. These terms are often interchangeable. Depends on whether it is the geometric and dynamic properties. However, some authors these names differ from each other.

It is important to discuss their properties and investigate their samples, because the most important thing is to recognize the chaotic behavior when it occurs in the experimental data. If we had a large number of objects it is possible to address the statistical issue of how common it is chaos. What are the most common value of Lyapunov exponents, size dimensions and other properties.

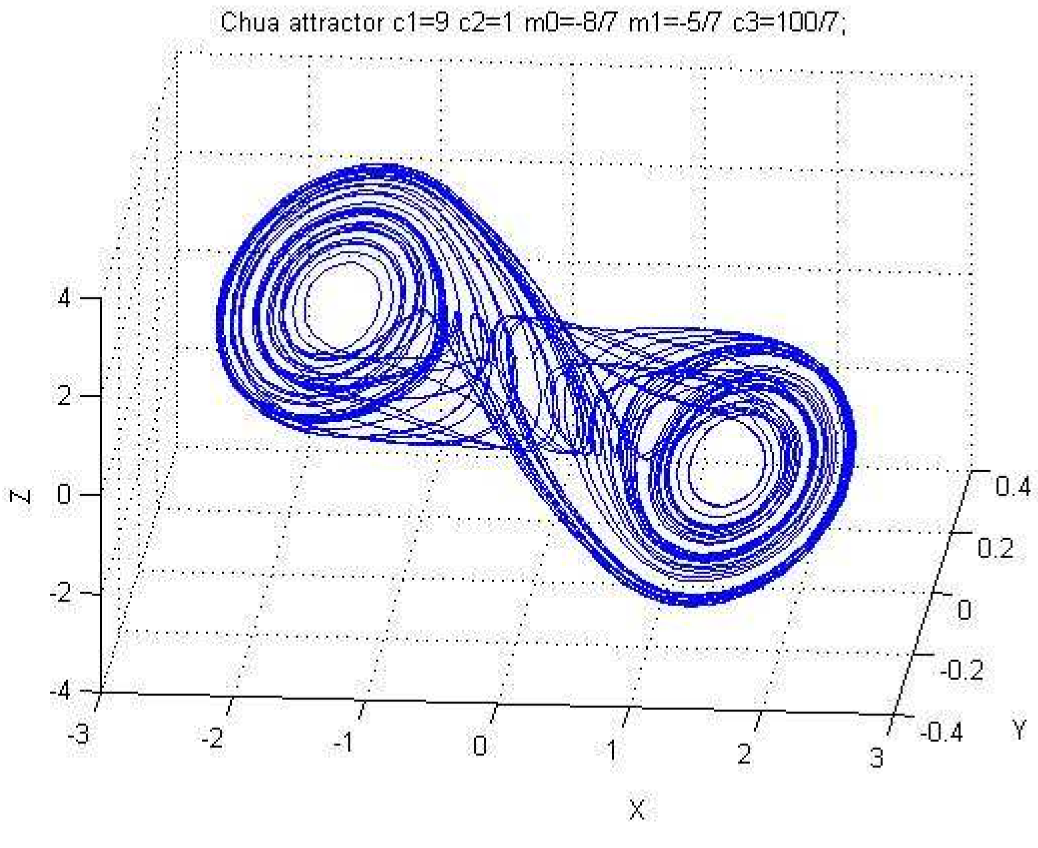


Fig. 4. An example of Chua's strange attractor

5 SPECTRUM OF CHAOTIC SIGNALS

As we know, we can express the signal using the harmonic signals, i.e. the functions sin () and cos () function is generally complex exponentials. With such simple considerations, we can convert the signal from time domain into the frequency domain. For a general transfer of signals from the time domain there is a Fourier's transformation. Fourier transform S (ω) of function s (t) is defined by an integral relationship:

$$S(\omega) = \int_{-\infty}^{\infty} s(t)e^{-i\omega t}dt. \tag{3.}$$

The definition clearly shows that the spectrum is a complex quantity. There are therefore amplitude spectrum and phase spectrum. We see that the spectrum is a continuous function defined from minus infinity to infinity. This, however, for many signals cannot be calculated analytically. It can be for example digital signals. Using computational tools, such as a PC, always leads to discretization of the signal. So if we have discrete samples, it is possible to replace the function of integral by function sum ().

$$S(\Omega) = \sum_{k=-\infty}^{\infty} s(k)e^{-i\Omega k}. \tag{4.}$$

Then for each Ω there is a sum of products weighted by exponent. Theoretically, the chaotic signal, we know has no period. This means that it is in each time different. If would signal be an infinitely long (in time sense), it would mean that the range of its spectrum is also endless.

Also we know that the attractors are attracted to the equilibrium. So if you re-introduce the state space and the length of signal is infinite, there is a demarcation, which has resulted in a reduction spectrum width. This means, that we can claim, that the spectrum of such signal is very wide-banded.

In the real world, we have limited bandwidth, so we tried to simulate a reconstruction of signal after filtration and how it affects results. It is important to investigate, because we know the sensitivity to initial conditions.

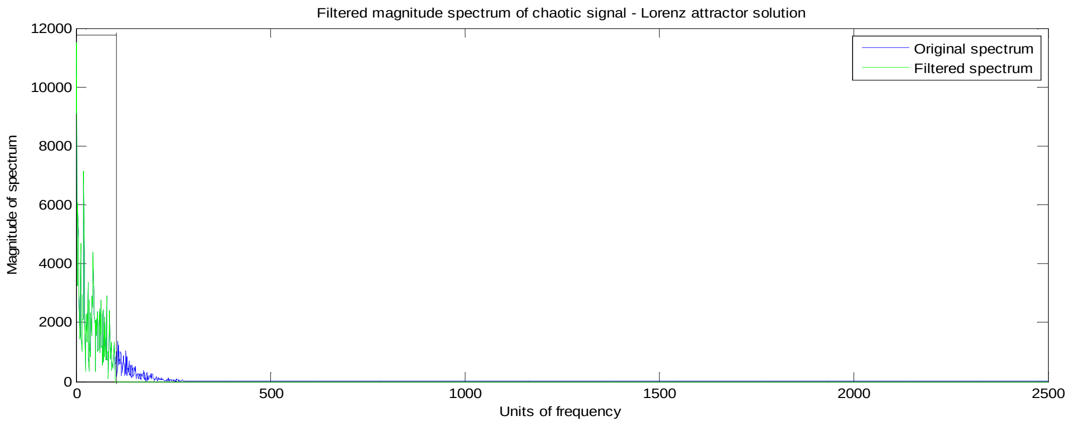


Fig. 5 Spectrum of filtered and original Lorenz attractor

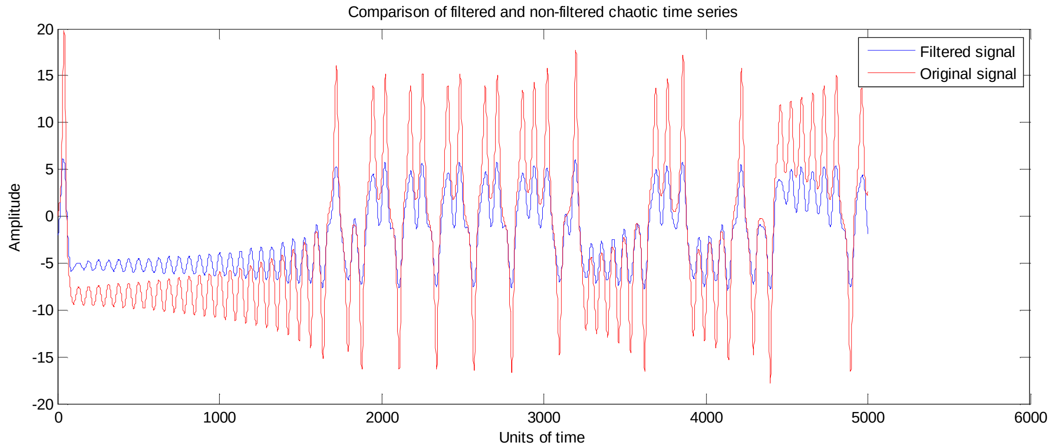


Fig. 6 Reconstructed signal by IFFT

In this example can be seen a short segment of the spectrum filtration Lorenz attractor. After the reconstruction of signal it is visible that cropping of spectrum quite reflected the results. In communications technology, this example can be solved and repaired. But if we use a very long signal (in this case signal generated by Labyrinth chaos), how it is used in next figure, we can prove what we claimed before. The spectrum of signal is wider. In this example we have crop the spectrum more.

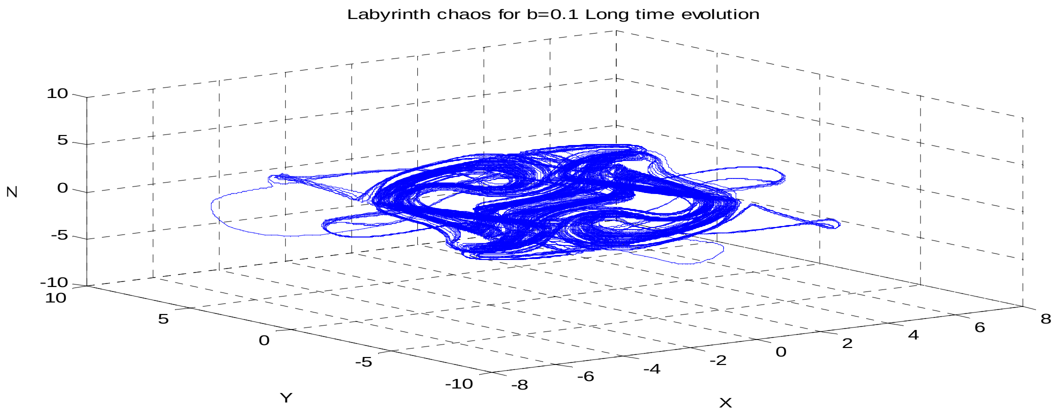


Fig. 7 Labyrinth chaos for b=0.1

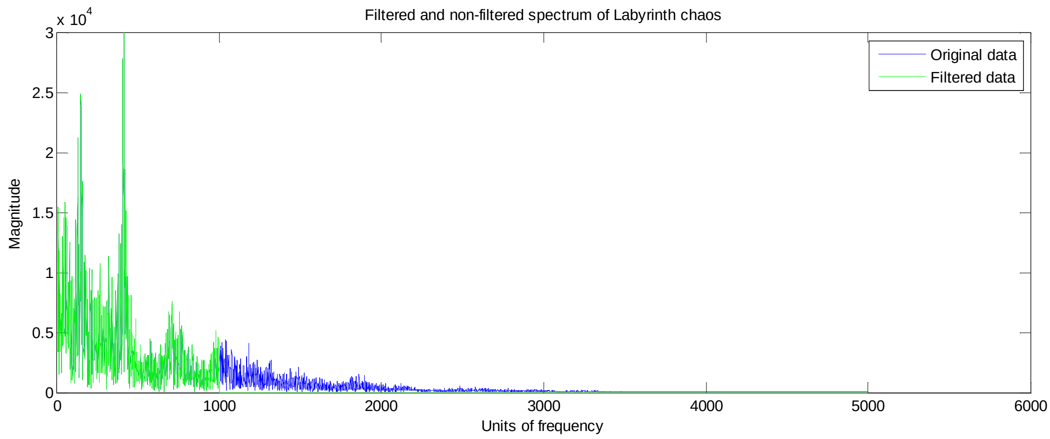


Fig. 8 Spectrum of long chaotic signal (Labyrinth chaos)

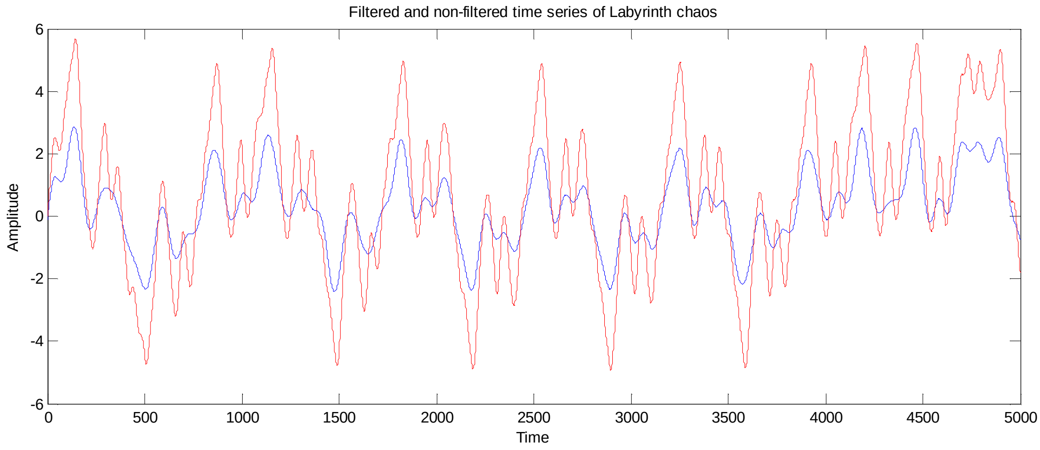


Fig. 9 Time series after reconstruction

In this case, we can see that the filtered reconstructed signal varies from the original signal. This means, that any jamming or filtering of signal can lead to different results.

6 FRACTAL DIMENSION

The value of dimension of the chaotic system is used for estimating the rate of how chaotic the system is. It is also called the fractal dimension. For systems without fractal structures, such as fixed points, limit cycles and others, fractal dimension is an integer value. In contrast, fractal dimension for chaotic attractors with a fractal structure is a real number. Dimension can be interpret as a number of real parameters needed to determine the point position on object, or as an exponent expressing the change in quantity when resizing. Quantity we consider for example volume. And size means a chosen characteristic dimension. So it is a statistical quantity that gives an indication of how completely a fractal appears to fill space, as one zooms down to finer and finer scales. There are many specific definitions of fractal dimension. The most important theoretical fractal dimensions are the Rényi dimension, the Hausdorff dimension and packing dimension. Practically, the box-counting dimension and correlation dimension are widely used, partly due to their ease of implementation.

Although for some classical fractals all these dimensions do coincide, in general they are not equivalent.

6.1 Rényi dimensions

The box-counting, information, and correlation dimensions, can be seen as special cases of a continuous spectrum of generalized or Rényi dimensions of order α, defined by

$$D_\alpha = \lim_{\varepsilon\to 0} \frac{\frac{1}{1-\alpha}\log \sum_i p_i^\alpha}{\log\frac{1}{\varepsilon}}. \qquad (5.)$$

where the numerator in the limit is the Rényi entropy of order α. The Rényi dimension with $\alpha=0$ treats all parts of the support of the attractor equally.

6.2 Hausdorff–Besicovitch dimension

The Hausdorff dimension introduced by Felix Hausdorff, gives a way to accurately measure the dimension of complicated sets such as fractals. The Hausdorff dimension agrees with the ordinary (topological) dimension on "well-behaved sets", but it is applicable to many more sets and is not always a natural number.

$$D = \frac{\log N}{\log\frac{1}{s}}. \qquad (6.)$$

Where N is the number of parts, at which an object (system) we divide. The variable s corresponds to the N-fold reduction in scale.

6.2.1 The Box Counting Dimension

Box-counting dimension is a simple way of estimating the Hausdorff dimension for fractals. We compute the box-counting dimension from a grid that is superimposed on a fractal image and counting how many boxes in the grid contain part of the fractal. Then you increase the number of boxes in the grid (but covering the same area: the boxes get smaller) and count again. If the numbers of boxes in the first and second grids are G_1 and G_2, and the counts are C_1 and C_2, then you compute a dimension by the formula:

$$D = \frac{\log\frac{C_2}{C_1}}{\log\sqrt{\frac{G_2}{G_1}}}. \qquad (7.)$$

This method is very suitable for implementation in computer algorithms. Therefore, it is also very popular. Its disadvantage is the time required for multiple state variables. Using parallel processing method would have gone faster.

6.3 Kaplan-Yorke Dimensions

The calculation using the dimensions of this method is very simple. It builds on knowledge Lyapunov exponent. But exponents obtained, should be made by orthonormalization, because we need to ensure squareness. For computer solutions to the problems we'll deal with Gram-Schmidt method.

$$D_{KY} \equiv j - \frac{\lambda_1 + \ldots + \lambda_j}{\lambda_{j-1}}, \qquad (8.)$$

where the value of j is the maximum number of exponents Lyapunov. From this value we can estimate, if the tested object can be considered as chaos. It is because chaos system doesn't usually have an integer value of dimension as we are used.

Name	**Dimension**
Sierpinski triangle	1,585
Lorenz attractor	2,060
Surface of lungs	2,970
Cantor set	0.6309

Table 1. Examples of well known systems and theirs Kaplan Yorke dimension

7 POINCARE MAP

To create the illustration for the phase portrait of systems with higher dimension, we don't have enough methods that we have been using. The only way to capture the characteristics of the phase space portrait in the lower dimension is a projection or cut multidimensional general body surface. Simply Poincare map can be seen as a slice (subset) of the phase portrait, in which one or more state variables constant value.

More precisely, one considers a periodic orbit with initial conditions on the Poincare section and observes the point at which this orbits first returns to the section, thus the name first recurrence map. The transversality of the Poincare section basically means that periodic orbits starting on the subspace flow through it and not parallel to it. A Poincare map can be interpreted as a discrete dynamical system with a state space that is one dimension smaller than the original continuous dynamical system. Because it preserves many properties of periodic and quasi-periodic orbits of the original system and has a lower dimensional state space it is often used for analyzing the original system. In practice this is not always possible as there is no general method to construct a Poincare map.

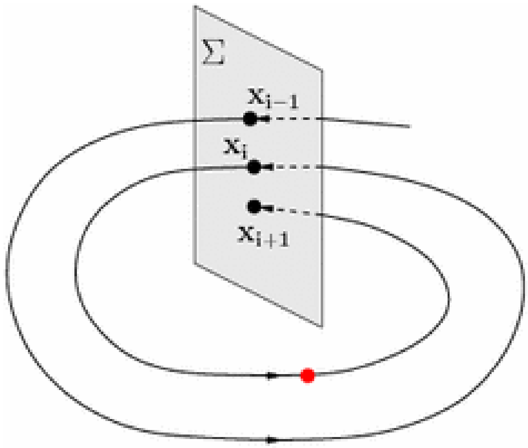


Fig. 10. Principle of Poincare map

From this definition it is apparent that a system exhibiting chaotic behavior will have a different Poincare's section. We can expect that the chaotic signal has many intersections with Poincare's plane. These intersections are irregular - chaotic. But it is a method that will help us decide whether the signal is chaotic. For example, the system will have periodic intersections, which are equidistant from each other, or they coincide.

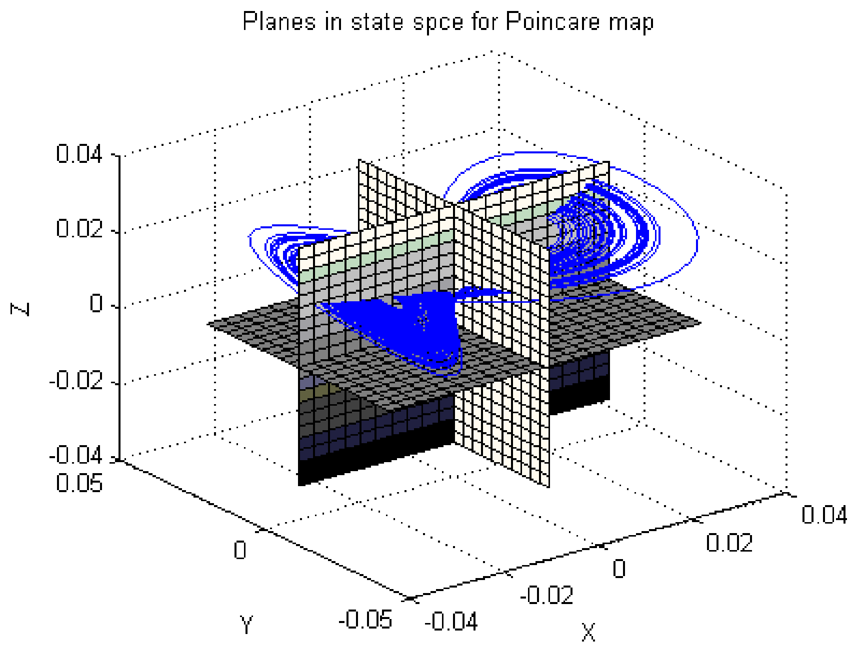


Fig. 11. Planes in state space – Lorenz attractor

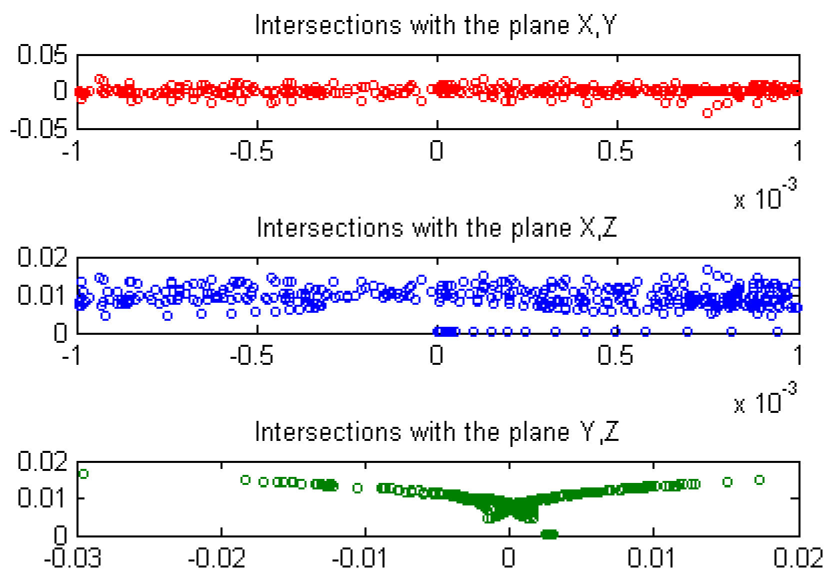


Fig. 12. Intersections with planes (for computer processing, there have to be a small range of values)

Despite its simple implementation of the algorithm, this method doesn't seem to be suitable for computer analysis. One of the problems that need to be solved in developing the algorithm is fitting the plane perpendicularly to the flow of signal. This can be solved using Gram-Schmidt orthogonalization process.

8 LYAPUNOV EXPONENTS

One measure (and most used) of sensitivity to initial conditions are Lyapunov exponents λ_n. It is the average value of divergence (or convergence) of two neighboring trajectories. The mathematics is Lyapunov exponent or characteristic exponent Lyapunov dynamic system variable, which characterizes the separation of infinitely close trajectories. So to describe the behavior of trajectories in the vicinity of any trajectory Γ used Lyapunov exponents, which are generalizing their own numbers or multipliers. Lyapunov exponents (LE) are real numbers that can be usefully applied for the classification non-chaotic and chaotic attractors. If the system in an unstable state can show that the two near trajectories are moving away faster than polynomial. Distance l of two points close to the trajectories can be approximated relationship

$$l(t) = d_r e^{\lambda_t}, \tag{9.}$$

where λ is the local Lyapunov exponent.

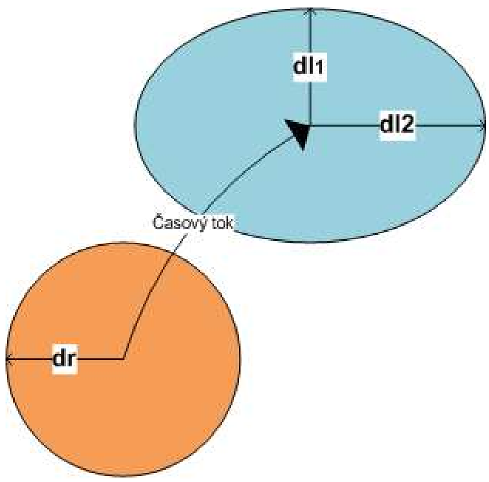


Fig. 13. Transformation circular trajectory to elliptical trajectory

Choose where sufficient space only one vector, we get one-Lyapunov exponent. In multi-phase space defined by the global spectrum Lyapunov exponents (each state variable corresponds to one). For one selected vector in the tangential space holds:

$$\lambda_i = \lim_{t\to\infty} \frac{1}{t} \ln\left(\frac{dl_i(t)}{dr}\right). \tag{10.}$$

Know if this range, then we can conclude that if all exponents are non-positive, the stable behavior of the system and if at least one exponent is positive. The system then behaves chaotically.

Lyapunov exponents are ordered to the time quantifications of the chaos. More is described in Table 2:

λ_1	λ_2	λ_3	λ_4	Attractor	Dimension
-	-	-	-	Equilibrium point	0
0	-	-	-	Limit cycle	1
0	0	-	-	2-torus	2
0	0	0	-	3-torus	3
+	0	-	-	Strange (chaotic)	>2
+	+	0	-	Strange (hyperchaotic)	>3

Table 2. Characteristic of the attractors for a four-dimensional flow [1]

We see that there are certain conditions that must be met for the emergence chaotic attractor. These characteristics are most used in the classification system, so when the subsequent optimization.

9 ESTIMATING LYAPUNOV EXPONENTS FROM DYNAMICAL EQUATIONS

There are two basic methods for calculating Lyapunov exponents. The first is precisely when we know the system of differential equations. Use them in some ways we get the Lyapunov exponents. The second way may be a set of data, or values (it is of course already on the integrated data). We will devote the first of the methods.

To get the vectors, we must first multiply the equation appropriate by Jacobi's matrix

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} \cdot \begin{pmatrix} x & y & z \end{pmatrix} \begin{pmatrix} d_1 \\ \cdot d_2 \\ d_3 \end{pmatrix} + \begin{pmatrix} b_1 \\ \cdot b_2 \\ b_3 \end{pmatrix} \cdot \begin{pmatrix} A & D & G \\ B & E & H \\ C & F & I \end{pmatrix} = \begin{pmatrix} J \\ \vdots \\ U \end{pmatrix}, \qquad (11.)$$

where the matrix with vectors A-I represents Jacobi's matrix, which is made by derivation`s of appropriate variables. For example, vector A have the following form:

$$A = a_{11} + 2xd_1c_{11} + d_1c_{12}y + d_1c_{13}z + yd_2c_{11} + zd_3c_{11}. \qquad (12.)$$

Next step is to integrate the differential equations (numerically) and the variation equation with random initial conditions. The system gives you information about the time development of small perturbation of x. It is a good idea to perform the Gramm-Schmidt orthogonalization of various y. Then

$$Ly = \frac{1}{t} \log\left(\frac{y(t)}{y(0)}\right) \qquad (13.)$$

approaches the largest Lyapunov exponent.

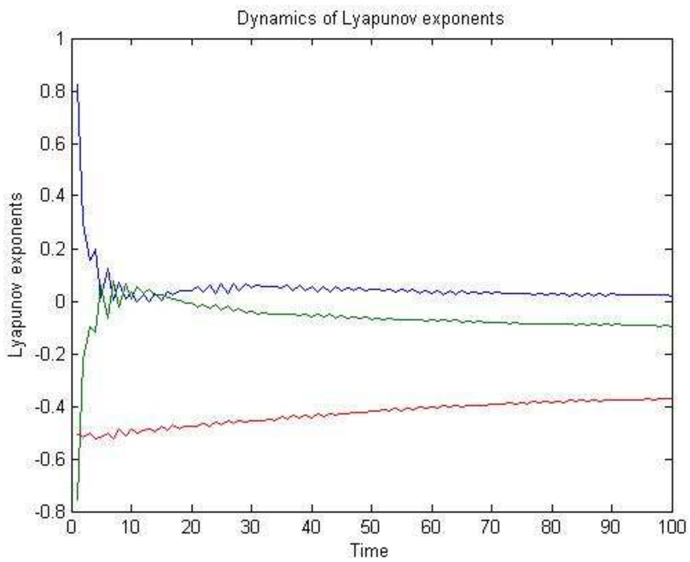


Fig. 14. Dynamics of Lyapunov Exponents (example for Lorenz`s equations)

10 TIME SERIES ANALYSIS

Few years ago, determining chaos from noise has become an important problem in many diverse fields. This is because, new numerical algorithms for quantifying chaos using experimental time series has been developed. There exist many methods. For example calculating a correlation dimension. The dimension gives us an estimate of the system complexity. The most discussed method is calculating the characteristic Lyapunov exponents. That is because the exponents give us an estimate of the level of chaos in the dynamical systems. When we know the level of chaos, we can start to predict the data. Because of this, we know, how accurate is our prediction.

10.1 Lyapunov spectrum defined

Now we define the Lyapunov spectrum in the manner most relevant to spectral calculations. In given dynamical continuous system (n-dimensional), we monitor in the long term the evolution of an infinitesimal n-sphere. When the n-sphere evolves from initial conditions, it deforms and become n-ellipsoid, following the flow. The i-th Lyapunov exponent is then defined along the principal ellipsoidal axis pi(t):

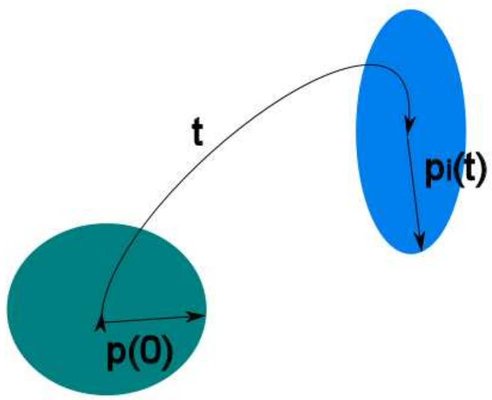


Fig. 15 Meaning of the Lyapunov exponents

$$\lambda_i = \lim_{t\to\infty} \frac{1}{t} \log_2 \frac{p_i(t)}{p_i(0)}. \qquad (14.)$$

Where the λ_i is aligned from the largest to smallest. But as the system evolves throw the time, the direction of exponent changes an many various ways. So we cannot speak about well defined direction given by exponent. exponents can also vary, because we can use a different base of logarithm. But most used are base of 2 and e. Generally we can say, that one or more positive exponents for dissipative attractor means that the system is “strange” of “chaotic”. When we have marginally stable orbit, the exponent is zero.

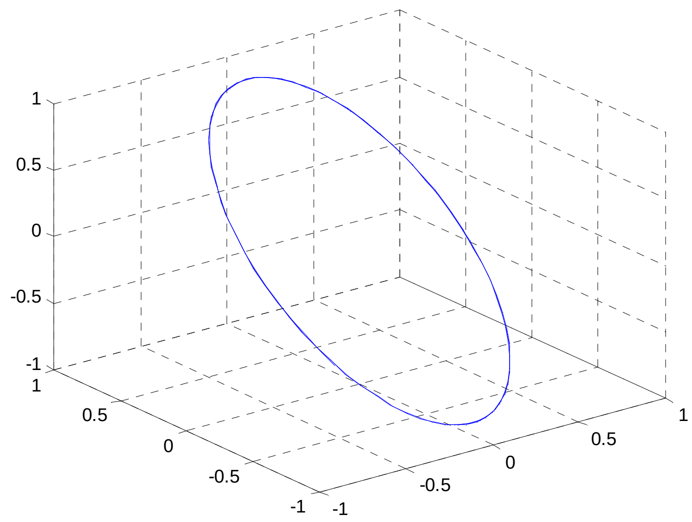


Fig. 16 Periodic data

10.2 Wolf’s method

Given the time series x (t), an m dimensional phase portrait is reconstructed with time delay method, because usually in experimental data, we don’t obtain all variables. So we locate the nearest point (in Euclidian sense) to the initial point $\{x(t_0), \ldots x(t_0+(m1)\tau\}$ and denote the distance between a point on attractor given by $\{x(t), \ldots x(t_0+(m-1)\tau\}$. The distance is designate by $L(t_0)$. The next distance will be $L'(t_1)$. The length element is propagated throw the attractor for a time short enough so that only small scale attractor structure is likely to be examined.

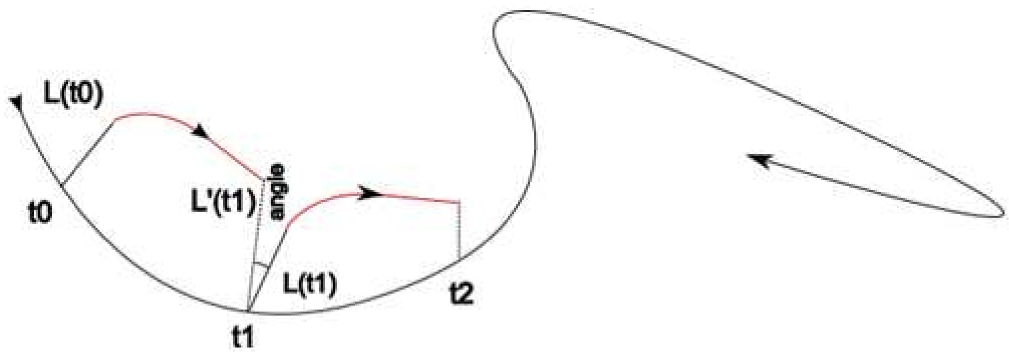


Fig. 17 Discribing the Wolf’s method

If the evolution is to large, we may see L’ shrink as it pass throw the folding area of attractor. So that could lead to underestimate Lyapunov exponents. As we know, the most important for identifying chaotic series is only the largest Lyapunov exponent. It is because, the other spectrum tells us, how the trajectory evolves. But for understanding how chaotic data are, is most important the largest exponent. Usually positive largest Lyapunov exponent leads to chaotic data. We can even say that scale of largest Lyapunov exponent is measure, how chaotic the time series is.

Now we look for a new data point that fulfils two criteria sufficiently well. The step separation of L(t1) has to be small to fiducial trajectory. And the angular separation between evolved and replaced point is small. If we cannot find an adequate replacement point we keep the points that were being used. We should repeat this procedure, until the entire fiducial trajectory is traversed. In this point we can estimate

$$\lambda_1 = \frac{1}{t_M - t_0} \sum_{k=1}^{M} \log_2 \frac{L'(t_k)}{L(t_{k-1})} \tag{15.}$$

where M is the total number of replacement steps. [4]

10.2.1 Estimating λ1+ λ2

The algorithm is basically almost similar to estimating largest λ. But it is much more complicated in implementation to the program. A trio of data points is chosen. Containing a fiducial point and two nearest points. By this we have defined an area $A(t_0)$. This area evolution is monitored. This evolution also has to satisfy previous criteria. Propagation and replacement steps are repeated until the fiducial trajectory has traversed the entire data set. In this point we can estimate

$$\lambda_1 + \lambda_2 = \frac{1}{t_M - t_0} \sum_{k=1}^{M} \log_2 \frac{A'(t_k)}{A(t_{k-1})} \tag{16.}$$

Where t_k is the time of k-th replacement step.

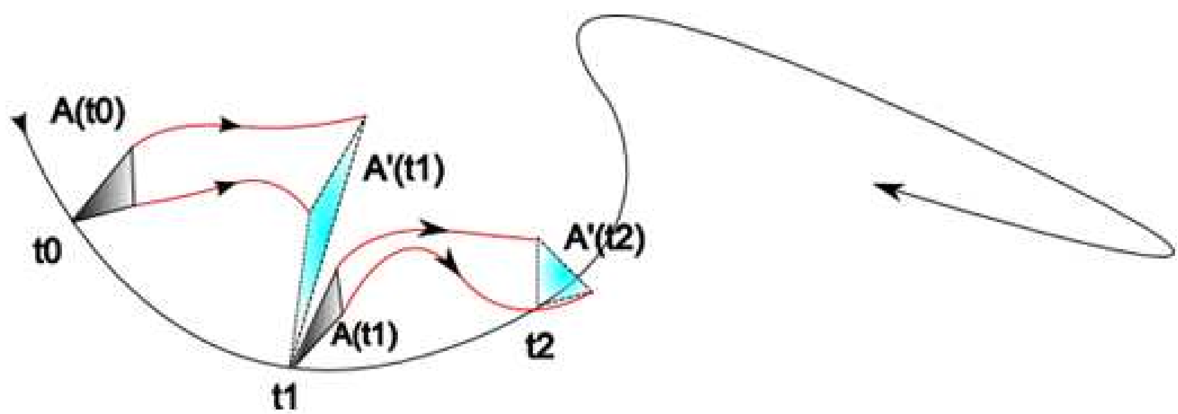


Fig. 18 Discribing the Wolf's method – the area calculating

It is possible to verify result for λ_1 throw the use of calculation $\lambda_1 + \lambda_2$. For attractors, that are very nearly two dimensional there is no need to worry about preserving orientation when we replace triples of points. These elements may rotate and deform within the plane of the attractor. But the replacement of triples always lies in the same plane. Since the λ_2 equals to zero, area evolution provides direct estimation of λ_1. We didn't impellent this algorithm because it is computational heftiness. [4]

10.3 The Rosenstein's algorithm

Rosenstein et al. [9], first step in this approach is also reconstructing the attractor dynamics from single time series. The approach is already mentioned before. They also use a delays method, where **X** is a matrix, where each row is phase-state vector. After reconstruction of dynamic the algorithm locate the nearest neighbors of each point on the trajectory. The nearest point $\mathbf{X}_{\hat{j}}$ is found by minimizing the distance between reference point $\mathbf{X}_j$. This is expressed as

$$d_j(0) = \min_{\mathbf{X}_{\hat{j}}} \left\| \mathbf{X}_{\hat{j}} - \mathbf{X}_{\mathbf{j}} \right\|, \tag{17.}$$

where $\|\ldots\|$ denotes Euclidian norm and $d_j(0)$ is the initial distance from the j^{th} point to its nearest neighbor. They define the additional limitation that the nearest neighbors have separation greater than the mean period (median frequency of the magnitude spectrum) of the time series,

$$|j - \hat{j}| > mean\ period. \tag{18.}$$

This allows us to take each pair of neighbors as almost initial conditions for initial trajectories. Then the largest Lyapunov exponent is estimated as mean rate of separation of the nearest neighbors. This method is easy to implement and fast, because it does not require large data sets and it uses a simple measure of exponential divergence that outwits the need to approximate the tangent map. It also simultaneously yields to the correlation dimension. This algorithm is also better than Wolf's, because it takes advantage of all available data. It does not focus on fiducial trajectory.

$$\lambda_1(i) = \frac{1}{i\Delta t} \cdot \frac{1}{(M-i)} \sum_{j=1}^{M-i} \ln \frac{d_j(i)}{d_j(0)}. \tag{19.}$$

Where Δt is the sampling period of time series, $d_j(i)$ is distance from the j^{th} point to its nearest neighbor after i^{th} time steps. M is a number of reconstructed points. From the previous definition of λ_1 we expect that the j^{th} pair of nearest neighbor diverge approximately by rate given by largest Lyapunov exponent.

$$d_j(i) \approx C_j e^{\lambda_1(i.\Delta t)}. \tag{20.}$$

Where C_j is the initial separation. When we logarithm both sides, we obtain

$$\ln\left(d_j(i)\right) \approx \ln\left(C_j\right) + \lambda_1(\mathrm{i}.\Delta\mathrm{t}) \tag{21.}$$

Equation represents set of parallel lines, each slope is proportional to λ_1. The largest Lyapunov exponent is then accurately calculated using least-square fit to the line defined by

$$y(i) = \frac{1}{\Delta t} \quad \langle \ln(d_j(i) \rangle. \tag{22.}$$

Where $\langle \ldots \rangle$ denotes average over all values of j. This process is key to calculation the exponent value. When we have a noisy data set, we have to especially take care about this part. [9]

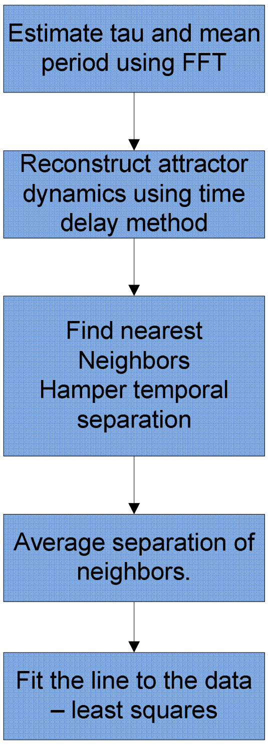


Fig. 19 Algorithm diagram

10.4 Time Delay Embedding

The simplest method to embed scalar data is the method of delays. This works by reconstructing the pseudo phase-space from a scalar time series, by using delayed copies of the original time series as components of the reconstruction matrix. It involves sliding a window of length m through the data to form a series of vectors, stacked row-wise in the matrix. Each row of this matrix is a point in the reconstructed phase-space. Letting {Xi} represent the time series, the reconstruction matrix is represented as:

$$X = \begin{pmatrix} X_0 & \cdots & X_{(m-1)\tau} \\ \vdots & \ddots & \vdots \\ X_n & \cdots & X_{n+(m-1)\tau} \end{pmatrix}. \quad (23.)$$

where *m* is the embedding dimension and $\{\tau\}$ is the embedding delay (in samples). Fixing an optimal value of *m* requires domain specific knowledge about the time series being analyzed. The method of false-nearest neighbors can be useful to some extent in this regard. Underestimating the value for delay leads to highly correlated vector elements, which would now be concentrated around the diagonal in the embedding space, and the structure perpendicular to the diagonal is not captured adequately. On the other hand, a very large estimate of the delay will result in the elements of each vector to behave as if they are randomly distributed. Quantitative tools like auto-correlation and auto-mutual information are useful guides in choosing the optimal value of $\{\tau\}$.

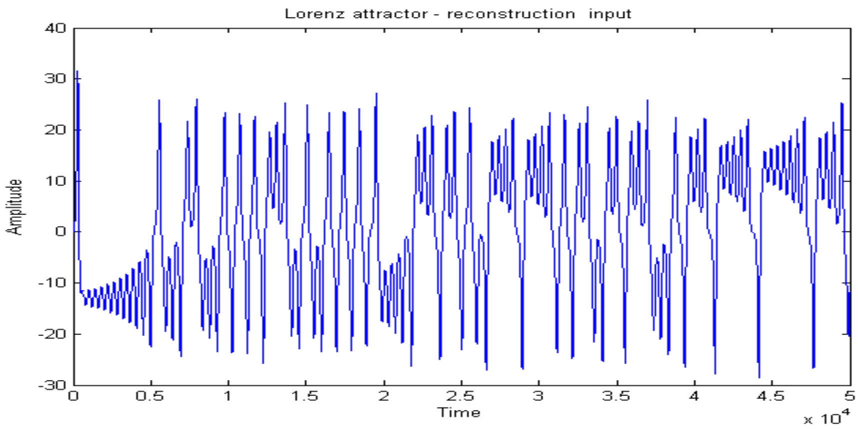


Fig. 20 Imput data

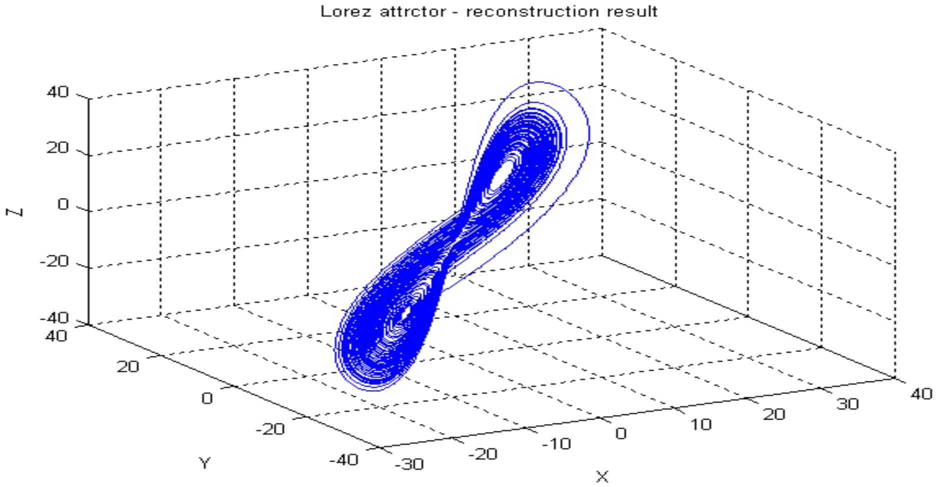


Fig. 21 Data after reconstruction

10.5 Estimating tau

10.5.1 Autocorrelation

A number of criteria for selecting τ w depend upon the autocorrelation function, Rxx (τ). Number of criteria for the selection τ w depends on the autocorrelation function Rxx (τ). Autocorrelation function provides a measure of similarity between the signal x (t), and delayed version of itself, because Rxx (τ) is maximized when the delay is zero. Autocorrelation function is not required to provide a reasonable transition from redundancy to irrelevance (depending on the delay). Usually τ_w chosen as the delay where Rxx (τ) first drops to a fraction of its original value. Similarly, it may be τ_w selected location first inflection point Rxx (τ). Related criterion based on Fourier transform Rxx (τ), i.e. the power spectrum of x (t) is the inverse group-off frequency. Autocorrelation-based methods have the advantage of short calculation time is calculated using the fast Fourier transform (FFT) algorithm. As suggested by a number of changes, however, these methods tend to be inconsistent. This means that a particular criterion may be better for a dynamic system and bad for another. This is not surprising, given the ill-defined relationship between the location of the reconstructed attractor and the temporal autocorrelation single time series.

10.5.2 Mutual information

In contrast to the linear dependence measured by autocorrelation, mutual information, I(τ), supplies a measure of general dependence. Therefore, I(τ) is expected to provide a better measure of the shift from redundancy to irrelevance with nonlinear systems. Mutual information answers the following question: Given the observation of x(τ), how accurately can one predict x(t + τ)? Thus, successive delay coordinates are interpreted as relatively independent when the mutual information is small. The minima of I(τ) has the same value as the correlation integral, Cm(r; τ). The computation is less demanding as computing the minima of I(τ).

$$C_m(r;\tau) = \frac{2}{M(M-1)}\sum_{i\neq j}^{M}\theta[r - \|X_i - X_j\|], \qquad (24.)$$

where M is the number of reconstructed points, q[..] is the Heaviside function, and denotes the Euclidean norm. It follows that an algorithm for calculating correlation dimension is easily adapted to estimate τ . The problem is, that this approach But requires enormous computational costs.

10.5.3 Symplectic geometry

Symplectic geometry is a branch of differential geometry and differential topology which studies symplectic manifolds. Symplectic geometry has its origins in the Hamiltonian formulation of classical mechanics where the phase space of certain classical systems takes on the structure of a symplectic manifold. A symplectic geometry method is proposed to determine the appropriate embedding dimension from a scalar time series. Symplectic geometry has measure preserving characteristic and can keep the essential character of the primary time series unchanged when performing symplectic similar transform.

10.6 Experimental results

It is very important to distinguish deterministic chaos from noise. When we have equations describing the system, it is more easy to decide whether the system embody marks of chaos. For testing the algorithm we have chosen a series of differential equations. We integrate to get the solution. Then we use only one signal to set as primary. Next thing, we have to reconstruct the data. It is normal, that we get only one or two signals, but the system contains much more.

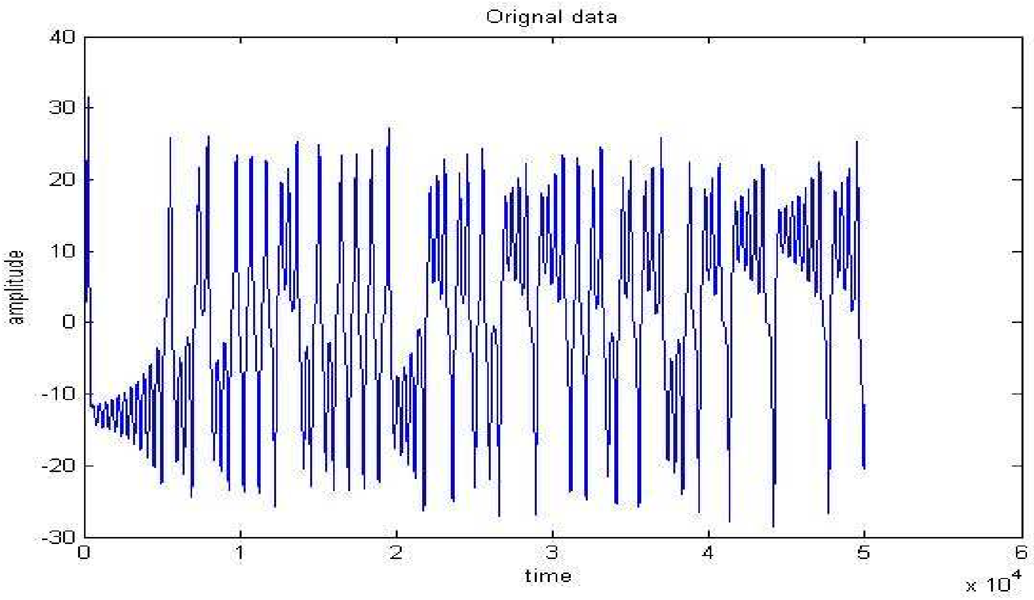


Fig. 22 Input data set

On this figure we can see the original data. We use a delay method for reconstruction with parameters of m=3 and tau=20. Of course, we can vary the parameter, but this combination was estimated by symplectic geometry method (m) and from FFT(tau).

On this figure, we can see the result of reconstruction. To estimate the largest Lyapunov exponent, we wrote an algorithm based on Rosenstein's algorithm. We were trying to make this algorithm fully automatic. That means to estimate the largest exponent without any knowledge of the system or parameters. But it is almost impossible, because in the algorithm we need to estimate the slope. That is not a problem. Problem is to choose the right area for estimating the slope. For experimental data obtained from Lorenz attractor with parameters:

$$\begin{aligned} \frac{dx}{dt} &= 16.(y-x) \\ \frac{dy}{dt} &= x(45.92-z)-y \\ \frac{dz}{dt} &= xy-4.z. \end{aligned} \tag{25.}$$

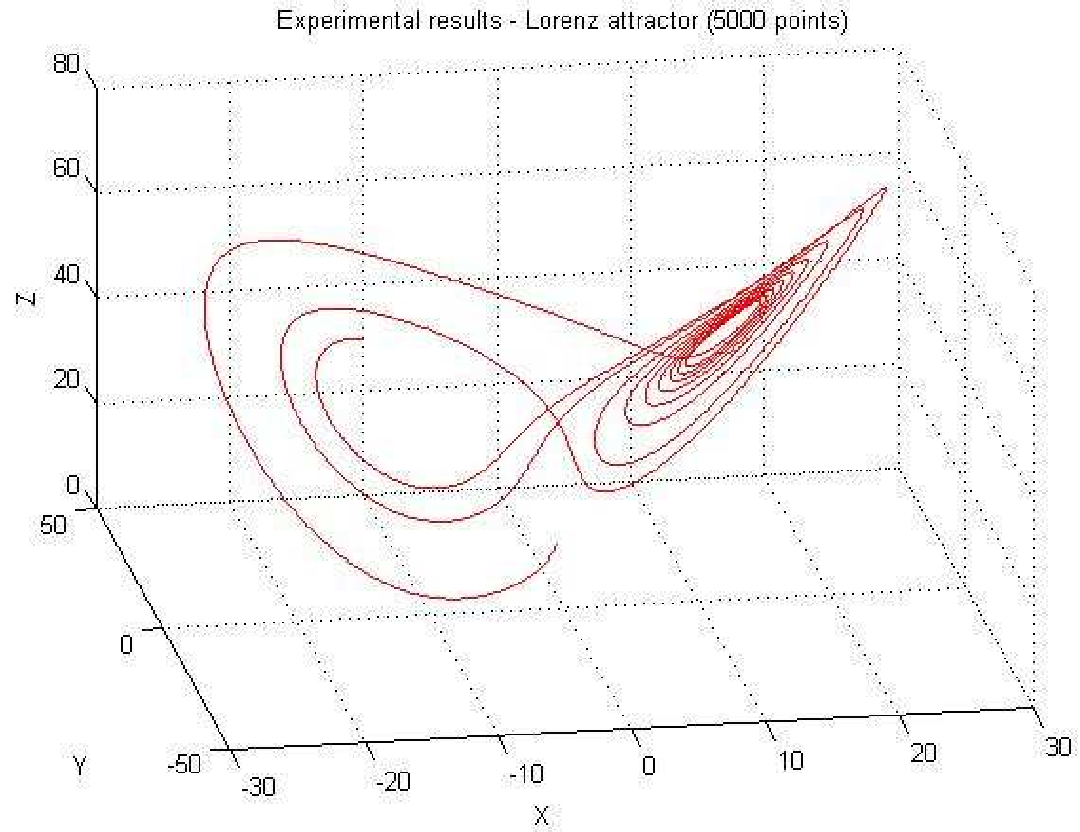


Fig. 23 Lorenz attractor with 5000 points

We used only signal x, reconstructed as mentioned before with parameters tau=17 and m=3. For this parameters we obtained λ_1=1.59 (base of e) and the expected λ_1 is 1.50. But as we said it is not easy to find the right region as shown on the next figure.

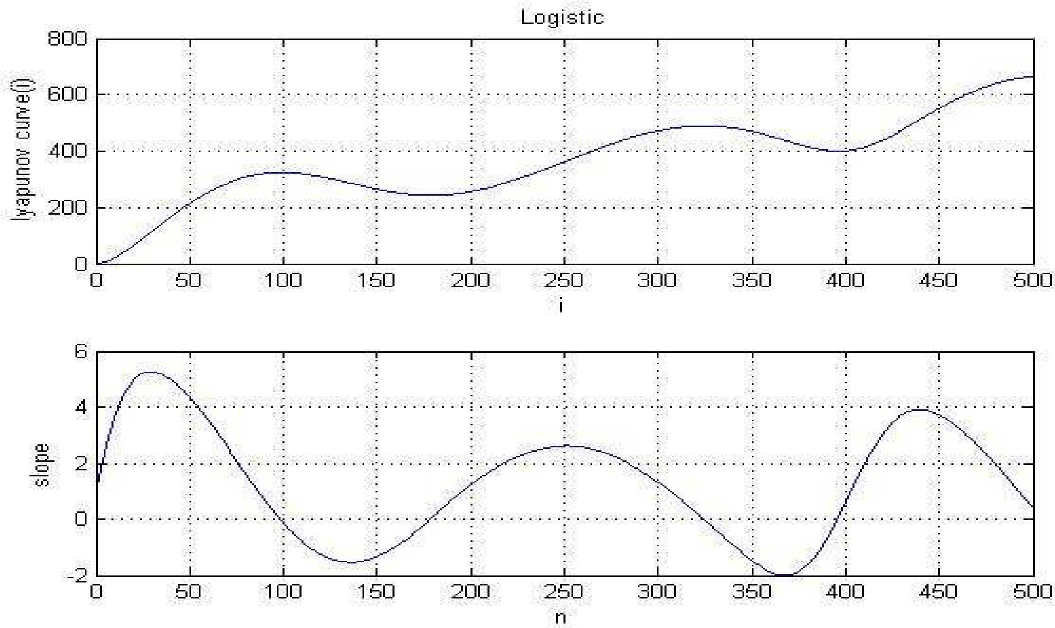


Fig. 24 Lyapunov curve and its slope

On the next figure is shown the slope sensitivity.

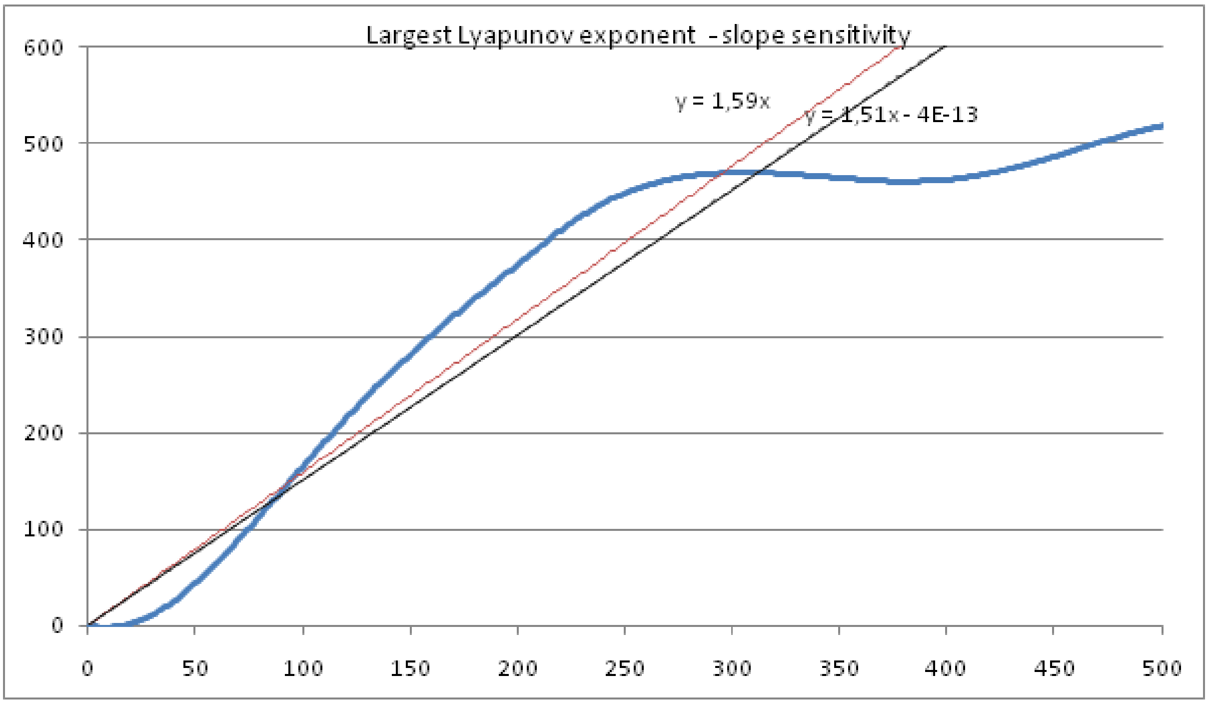


Fig. 25 Largest Lyapunov exponent – slope sensitivity

As we can see, we can very easily obtain an error in this part. But we can also very easily decide whether the data are chaotic.

We also tested the algorithm for periodic data to verify the correctness of its results. We took an equation y = cos(t) and reconstructed it with parameters m = 3, tau = 17.

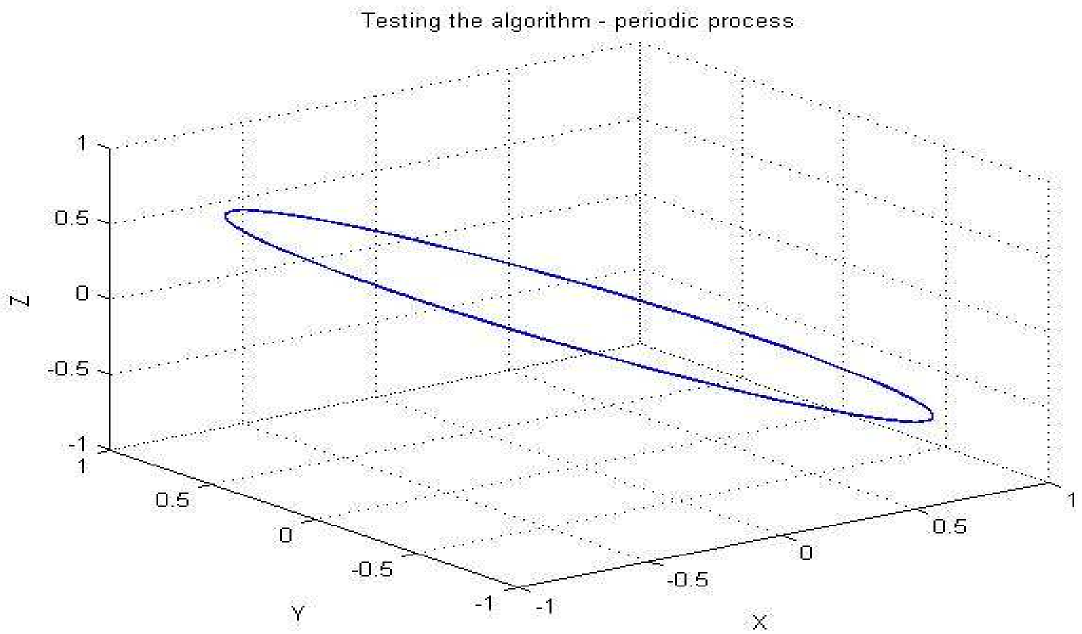


Fig. 26 Testing the algorithm

As we can see, the data doesn’t contain any noise. To obtain data without noise with measurement is almost impossible.

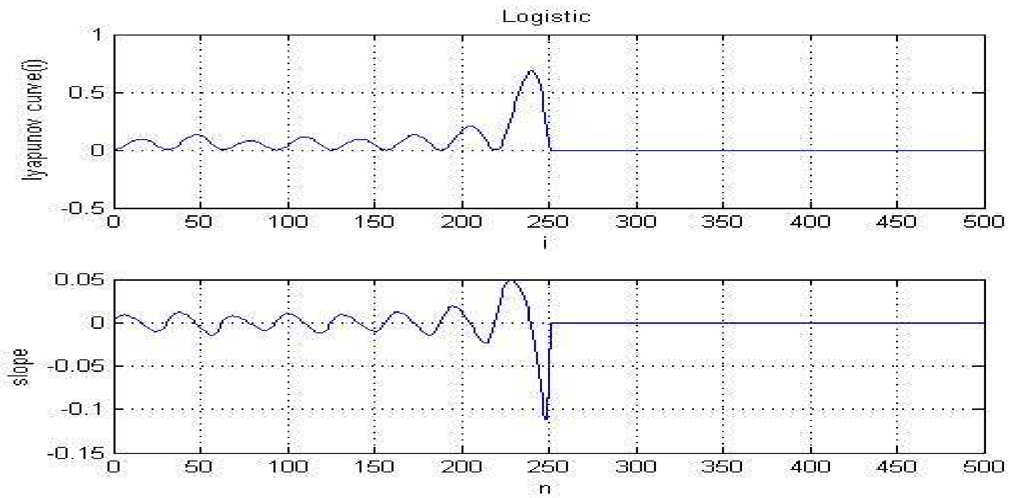


Fig. 27 Lyapunov curve and its slope for periodic data set.

Based on theory, what represents Largest Lyapunov exponent, the value have to be 0. We have obtained λ_1= 3.1522e-004.

We have also tested the algorithm with adding the white Gaussian noise to the periodic data. We have added the noise with SNR=20dB.

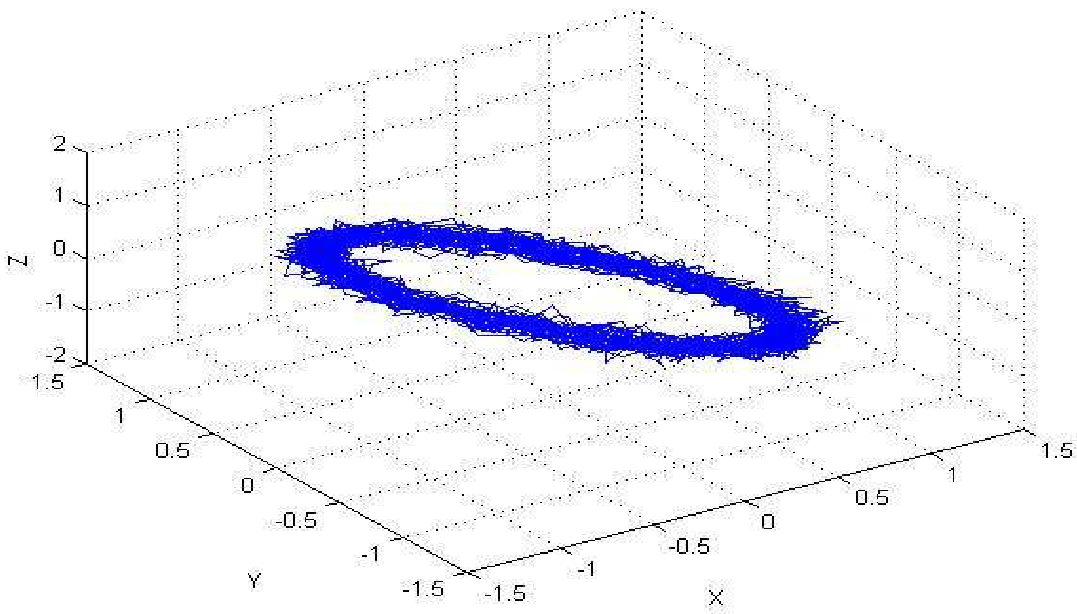


Fig. 28 Periodic data with white Gaussian noise

As we can see, the noise is added to the data set. The results are quite satisfactory. We again obtained result λ_1= 0.0011.

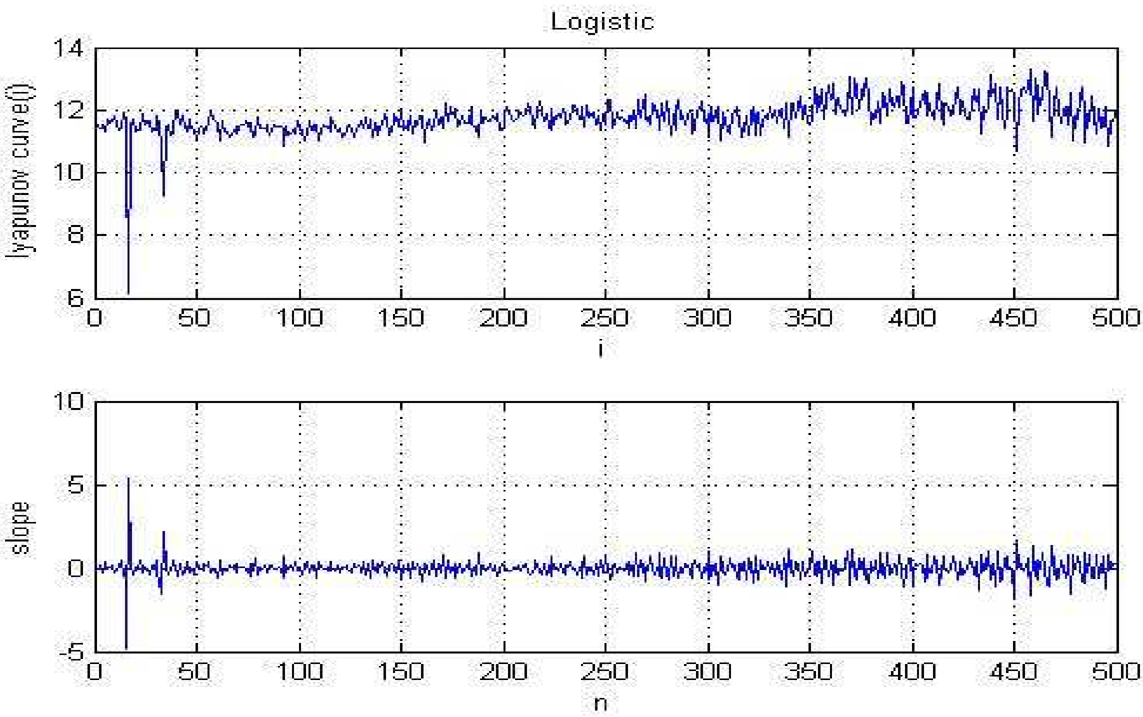


Fig. 29 The Lyapunov curve and it's slope

As we can see, the slope is oscillating about zero. That means that the algorithm is comfortable consentaneous to the theoretical expectations.

System	The number of data points	Reconstruction delay	Embedding dimension	Calculated λ_1	Expected λ_1	Error [%]
Rössler	400	8	3	0,0372	0,0900	58,7
	800			0,0722		19,8
	1200			0,1100		-22,2
	1600			0,0953		-5,9
	2000			0,0882		2,0
Rössler	2000	8	1	-	0,0900	-
			3	0,0877		2,6
			5	0,0866		3,8
			7	0,0850		5,6
			9	0,0832		7,6
Lorenz	1000	11	3	1,7550	1,5000	-17,0
	2000			1,3450		10,3
	3000			1,3720		8,5
	4000			1,3920		7,2
	5000			1,5230		-1,5
Lorenz	5000	11	1	-	1,5000	-
			3	1,5380		-2,5
			5	1,4770		1,5
			7	1,5820		-5,5
			9	1,5900		-6,0

Table 3 Experimental results for known systems

10.7 Testing the real data

To test the algorithm, we have set up the Chua's circuit. Chua's circuit is a simple electronic circuit that exhibits classic chaos theory behavior. It was introduced in 1983 by Leon O. Chua. An autonomous circuit made from standard components (resistors, capacitors, inductors) must satisfy three criteria before it can display chaotic behavior. It has to contain:

1. one or more nonlinear elements
2. one or more locally active resistors
3. three or more energy-storage elements.

Chua's circuit is the simplest electronic circuit meeting these criteria. As shown in the figure, the energy storage elements are two capacitors and an inductor. There is an active resistor. There is a nonlinear resistor made of two linear resistors and two diodes. At the far right is a negative impedance converter made from three linear resistors and an operational amplifier.

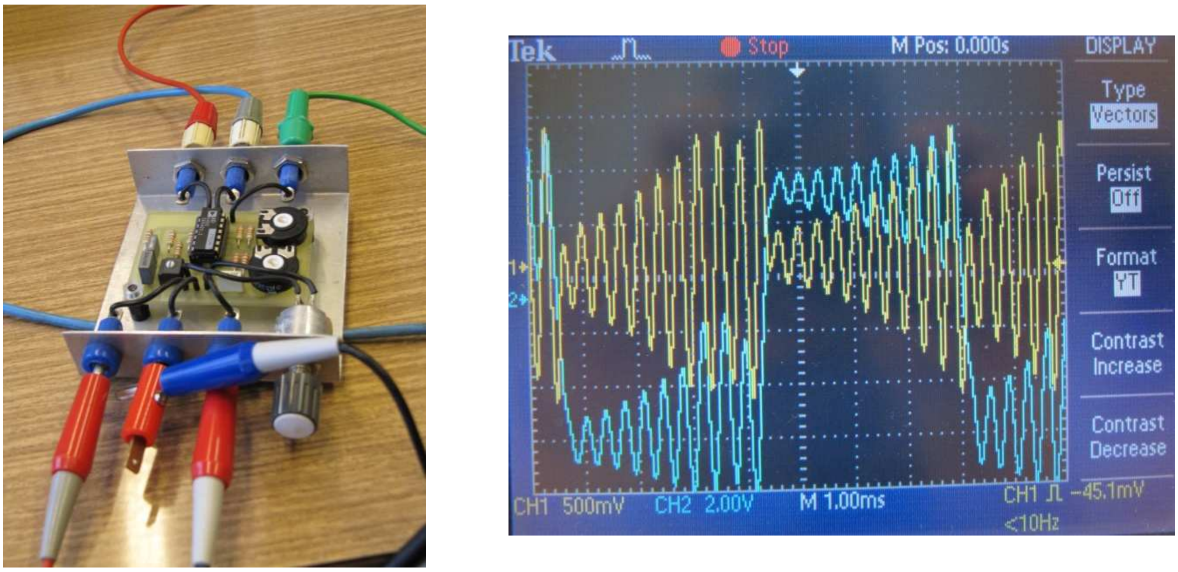


Fig. 30. Chua's testing circuit (left – circuit, right – time series)

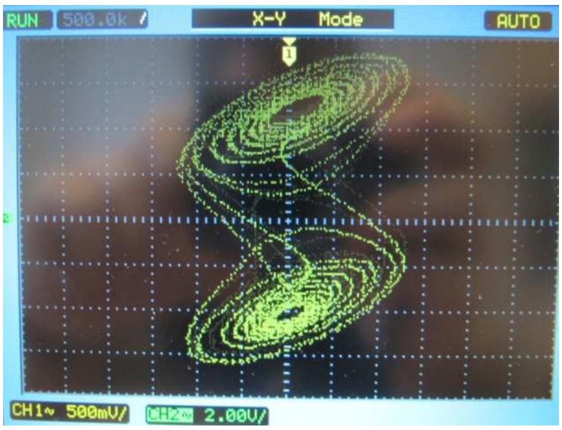
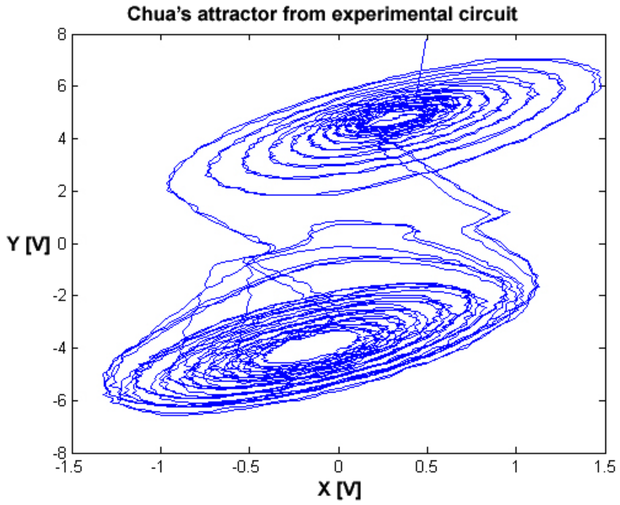


Fig. 31. 2D Notation in XY plane

So we take the input data and we analyzed them with Rosenstein's algorithm. Problem was, that the data were not enough consistent. Matlab algorithm trend to result infinity exponents or it wasn't a number. So we have to try to reduce the noise. It fortunately solved the problem. As we can see on the figure below, the attractor is smoother. The problem was in measuring the distance between neighbor points.



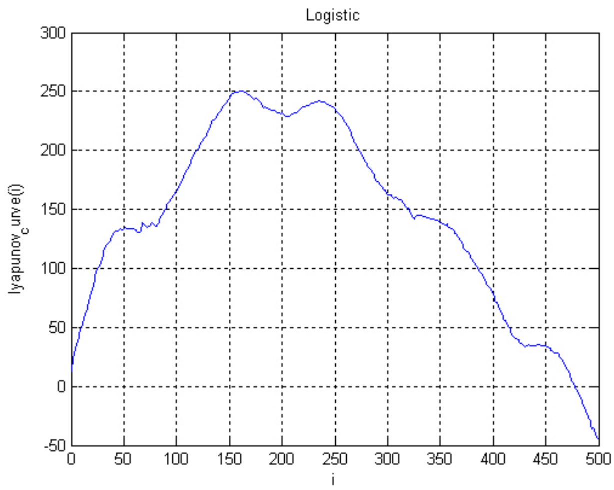


Fig. 32. Chua's attractor in Matlab and Curve for estimating the exponent

We have also decided to analyze the more common signal. We explore the ECG signal. We analyzed it as one dimensional object (although we know, it has more variables), but the chaos was also detected.

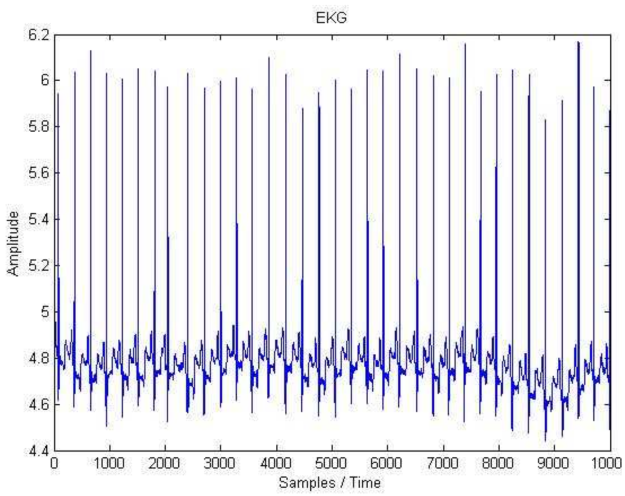


Fig. 34. ECG signal

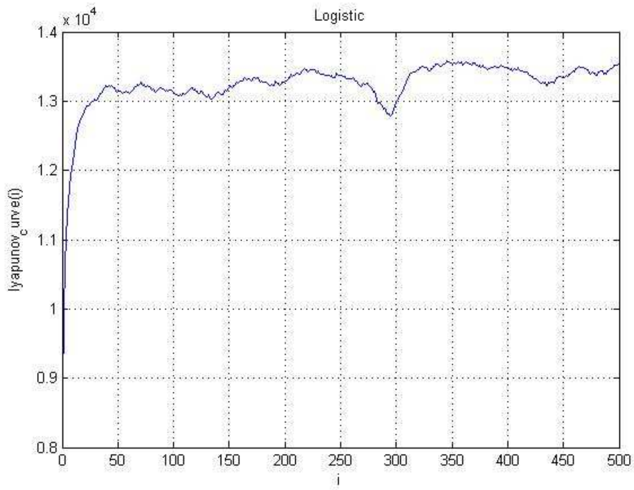


Fig. 33. Slope obtained from ECG signal

We get largest Lyapunov exponent equal to 0.74 which tell us about the level of chaos.

11 FINDING THE CHAOS

11.1 Variation of variables

In order to find a chaotic system, we can use the speed of data processing technology and to generate all combinations of variables different systems. For them we can then calculate the Lyapunov exponents, which may be regarded as quantifier chaotic system. The main problem is that the system is very sensitive to initial conditions. So we can find only approximate solution. The problem of the procedure is, however, not only in sensitivity, but also delays the calculations, because we combine all variations of variables. For example we can try to generate a system with two unknown variables.

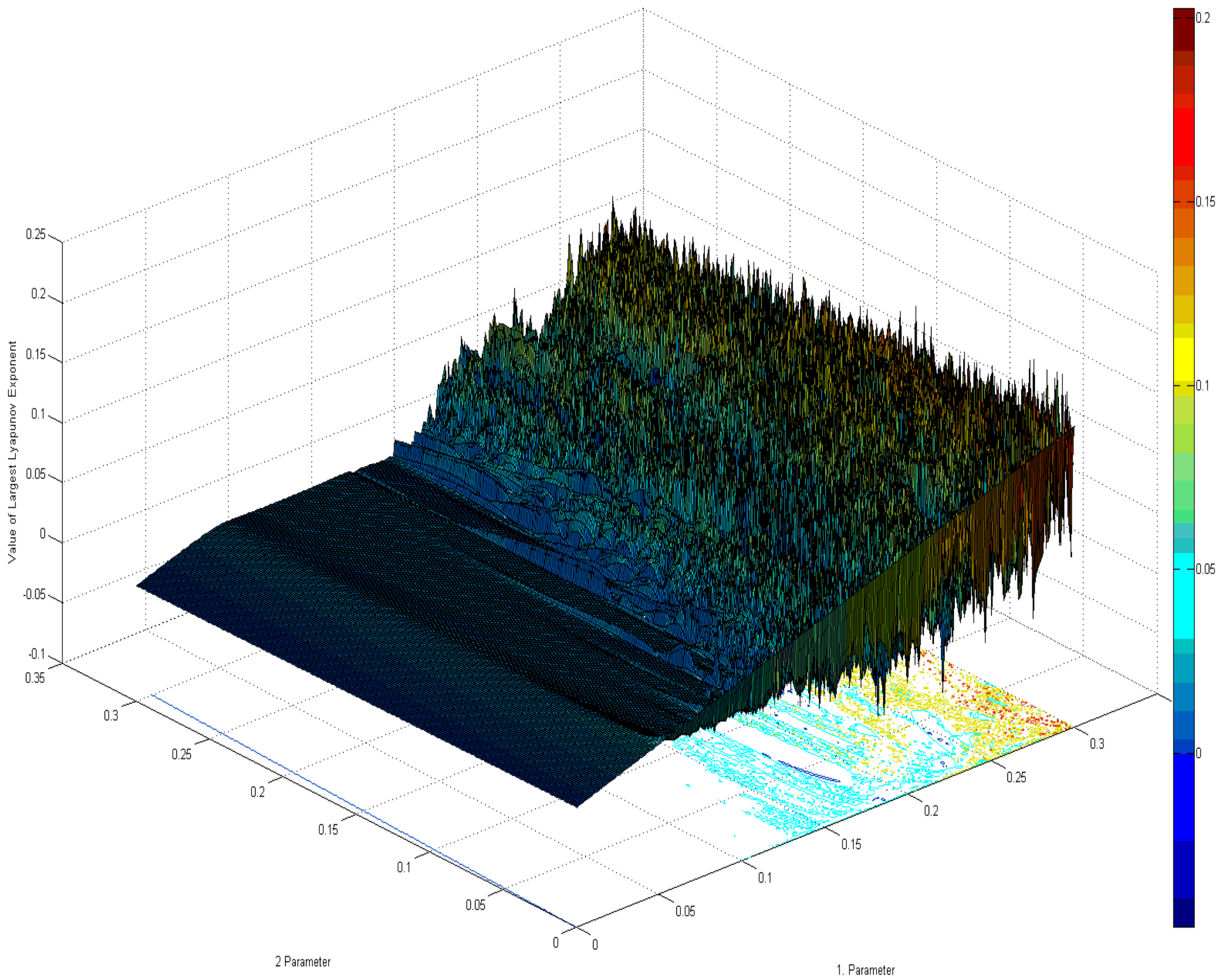


Fig. 35. Largest Lyapunov exponent generated by two variables

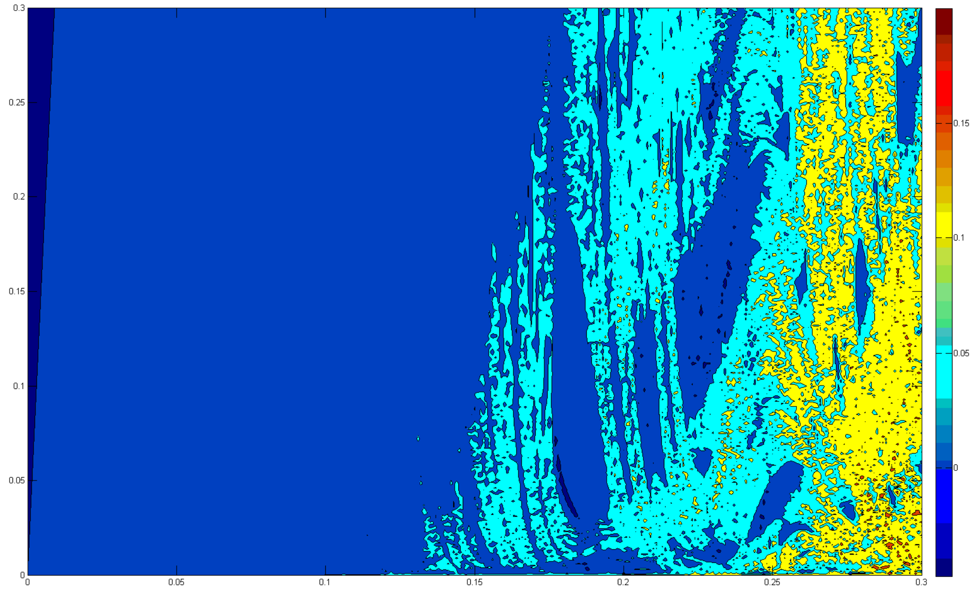


Fig. 36. Contour plot of Largest Lyapunov exponent generated by two variables

11.2 Optimization

Mathematical optimization task is an effort to find such values of variables for which the target or purpose function becomes a minimum or maximum value. Many theoretical problems and problems in the real world lead to the role of optimization solutions. Often occurs when modeling physical phenomena, where the target function f has a physical energy system, which is in the steady state system, be minimal.

11.3 Genetic algorithm

Genetic algorithm used for solving a special form of entertainment-inspired biological chromosomes and to generate new solutions to the crossing and mutation. Before a detailed description of the genetic algorithm, therefore, recall a few well-known knowledge of related areas. In each cell living organism is a set of chromosomes consisting of DNA. Chromosomes form a model of the entire body and consist of genes (units of DNA). Very simply speaking, that each gene represents a feature or characteristic, such as eye color. Possible states of a gene called allele, such as eye color may be blue, brown, etc. Each gene has a solid position in the chromosome. The entire genetic material of an organism is called genome. The specific "set" of genes in the genome is called genotype. The genotype determines the phenotype, the external characteristics of the organism (physical and mental - eye color and intelligence).

During the reproduction of organisms there is a crossover (recombination), which shall be selected genes taken from their parents and their combinations to create a descendant genotype. During or after reproduction may be a mutation - a random change of the minor genotype. The success of the organism (fitness - its rating) in the biology indicates a likelihood that the organism survives in its reproduction, or as a number of his descendants. Evolutionary theory then says that only certain organisms to survive and reproduce, the more likely to succeed them better. The entire above uses genetic algorithm. An acceptable solution to the problem, which solves the GA, is represented using the genome, i.e. a set of genes. The specific set of genes is the state and represents the genotype and phenotype, which is a concrete solution. Based on the phenotype is therefore determined by assessment solution and so are the assessment of genotype. Crossing and mutation, however, takes place only at the level of genotype. Genetic algorithm maintains a population of solutions in the form of genotypes (chromosomes), which mutates and crosses that favors genotypes with higher pay, and thus seeks to "grow" a good solution. At the beginning of your run will create a random population (first generation) and in this manner creates a new offspring (next generation) is not satisfied until a terminating condition (e.g., number of generations, time, the best fitness, etc.). Genetic algorithm can be described roughly as follows:

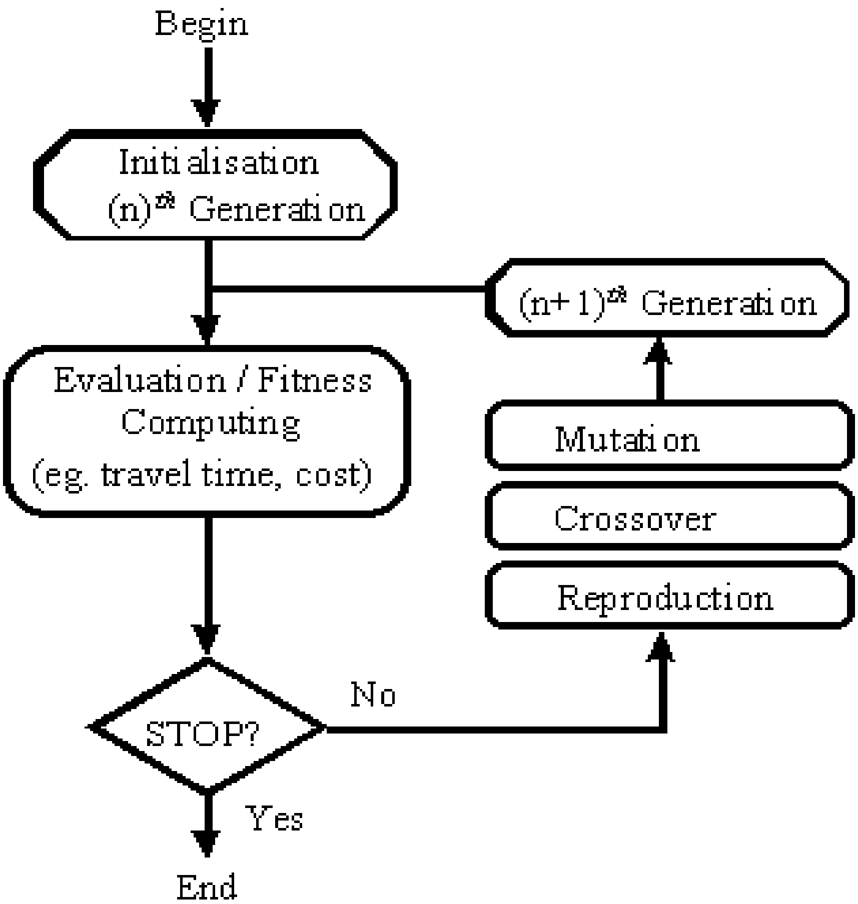


Fig. 37. Basic flow of genetic algorithm

The main strength of the genetic algorithm is its parallelism i.e. that searches multiple points of solutions at once (the whole population). As a result mainly of its resistance to deadlock in a local extreme (Imagine that, looking for the highest point in the mountains - if we look only at one place close around them, as well as much of the other optimization methods, and only by the will decide on the way, can we state that we find only a local peak (extra) and it is already in this move). Genetic algorithm is able to work without any special requirements for the scanned area and can, unlike other methods of finding good solutions even when the scanned area is "wild". Another important difference from other optimization methods is the separation of genotype and the creation of new solutions (the procedure in the solution space) from the phenotype and suitability assessment solutions.

For finding the results, we have chosen the implemented Genetic Algorithm and Direct Search Toolbox in Maltab. This toolbox can find minima of multiple functions using genetic algorithm. For implementing we can chose to use a single criteria function and create appropriate fitness function. We think that it is better to use multiple optimizations. Only disadvantage is that computing algorithm is not very transparent. Advantage is its simplicity of implementing it to our algorithm. Another thing is that this algorithm allows using automatic parallel computing. Of course that can speed up the process.

Serious problem we have to face, in finding the parameters, is to choose good integrating function for solving the ordinary differential equations. So if we choose the right parameters, the function can tend to diverge very quickly. This mean that the numbers can overflow in computer memory and it can lead to freezing the cycle. Using the build-in function ODE45 (or other functions) was problematic because of these deficiencies. Even though they function as a reference in the help of Matlab is noted that the integration is a Runge-Kutta algorithm, we were not able to prevent freezing. We therefore decided to write our own function for integration. For the integration it is necessary to use a function with regard to speed and accuracy, because we know that chaotic functions are sensitive even to a small change in initial conditions.

The best we seemed to use Runge-Kutta fourth order for finding approximate solutions. For more accurate calculation, we used the integrative function of the eighth order.

As already mentioned, this method has the advantage that it allows parallel processing of data for different input parameters. It was used the optimization toolbox in Matlab, which offers many options, as well as parallel processing. It should be noted that this is not a full parallel processing because of its matter. This is also because the Matlab process data only serially. Parallel option was introduced only recently, but on multiprocessor stations still uses only one processor. Even so, this option is speed up the calculation. For the calculation was created multidimensional fitness function, which was designed to maximize the greatest Lyapunov exponent and second exponents closer to zero. Throughout the program, it is added to the evaluation algorithm, which controls the values of exponents. At the time when meeting the conditions for the emergence of chaos, the value of variables, including exponents placed in a matrix. It is thus possible to re-check the calculations.

```
% Fitness function
fitnessFunction = @spousteni;
% Number of Variables
nvars = 1 ;
lb=[];%[-30 -30 -30];
ub=[];%[30 30 30];

% Start with the default options
options = gaoptimset;
% Modify options setting
options = gaoptimset(options,'Generations', 200);
options = gaoptimset(options,'PopulationSize' ,100);
options = gaoptimset(options,'PopInitRange', [-20;20]);
options = gaoptimset(options,'OutputFcns' ,{ [] });
fgoalopts = optimset('UseParallel','always');
```

```
[x,fval,exitflag,output,population,score] = ...
 gamultiobj(fitnessFunction,nvars,[],[],[],[],lb,ub,options);
```

Using the described algorithms have been found, for example, the following systems:

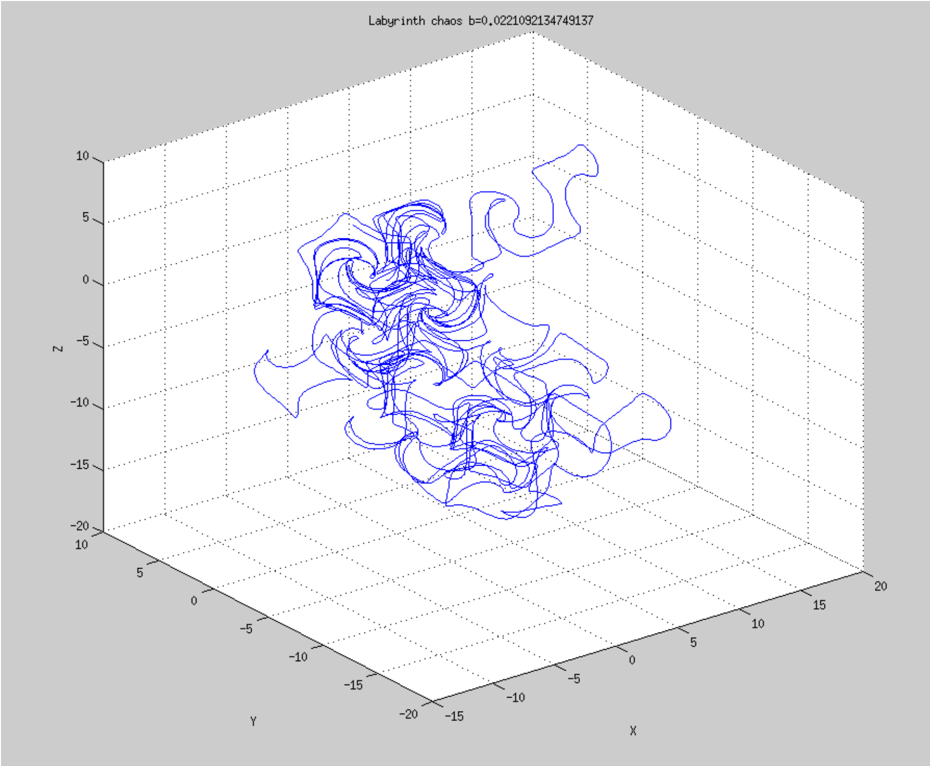


Fig. 38. Find parameter b=0.02210921 for the Labyrinth chaos

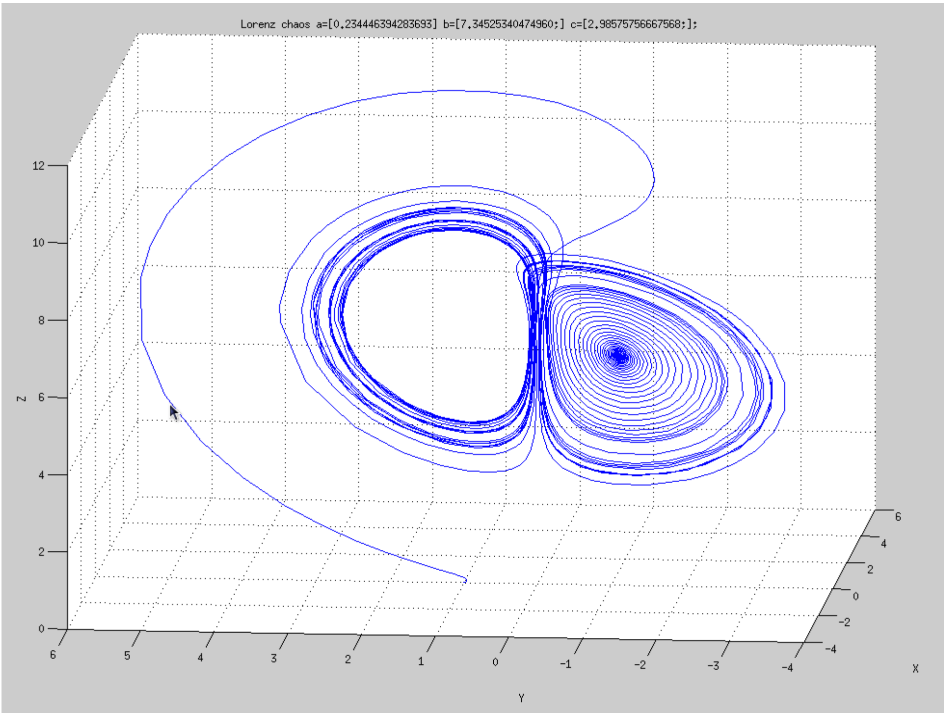


Fig. 39. Find parameters a=0.23444 b=7.345 c=2.985 for Lorenz equations

11.4 Particle swarm optimization

Particle swarm optimization (PSO) is an algorithm based on swarm intelligence. It finds a solution to an optimization problem in a search space, or model and predicts social behavior in the presence of objectives. PSO is a direct search method for finding the optimal solution for specified function in searched space. This function is called a fitness function (or an objective function). This algorithm is good, because it is easy to implement by a programmer and in its basic form, it return satisfying results. Also this method can be used by anyone without understanding the mathematical background and optimization theory.

PSO is a stochastic, population based computer algorithm, modeled on the swarm intelligence. Swarm intelligence is based on social-psychological principles and provides insights into social behavior, as well as contributing to engineering applications. The particle swarm optimization algorithm was first described in 1995. The swarm is typically modeled by particles in multidimensional space that have a position and a velocity. These particles fly through hyperspace and have two essential reasoning capabilities. Their memory of own best position and knowledge of the global or their neighborhood's best.

In a minimization optimization problem, problems are formulated so that "best" simply means the position with the smallest objective value. Members of a swarm communicate good positions to each other and adjust their own position and velocity based on these good positions.

For optimizing the problem, first we need to set a fitness function, describing the solution. We know that PSA is searching the minimum of the function. Also we know, that sign of each Lyapunov exponent is important. We have to consider the complexity of our problem. We are trying to find a solution for infinity variations of parameters. For example, if we have a nonlinear dynamic function with chaotic behavior (in special conditions). We can expect results with many local minimums.

As stated before, PSO simulates the behaviors of bees flocking. Suppose the following scenario: a group of bees are randomly searching flowers in an area. There is only one flower in the area being searched. All the bees do not know where the flower is. But they know how far the flower is in each iteration. So what's the best strategy to find the flowers? The effective one is to follow the bee which is nearest to the flower.

After finding the two best values, the partije (bee) updates its velocity and positions with following equations:

$$v = v + c_1.rand.(p_{best} - present) + c_2.rand.(g_{best} - present) \quad (26.)$$

$$present = present + v \quad (27.)$$

v[] is the particle velocity, persent[] is the current particle (solution). pbest[] and gbest[] are defined as stated before. rand () is a random number between (0,1). c1, c2 are learning factors. usually c1 = c2 = 2.

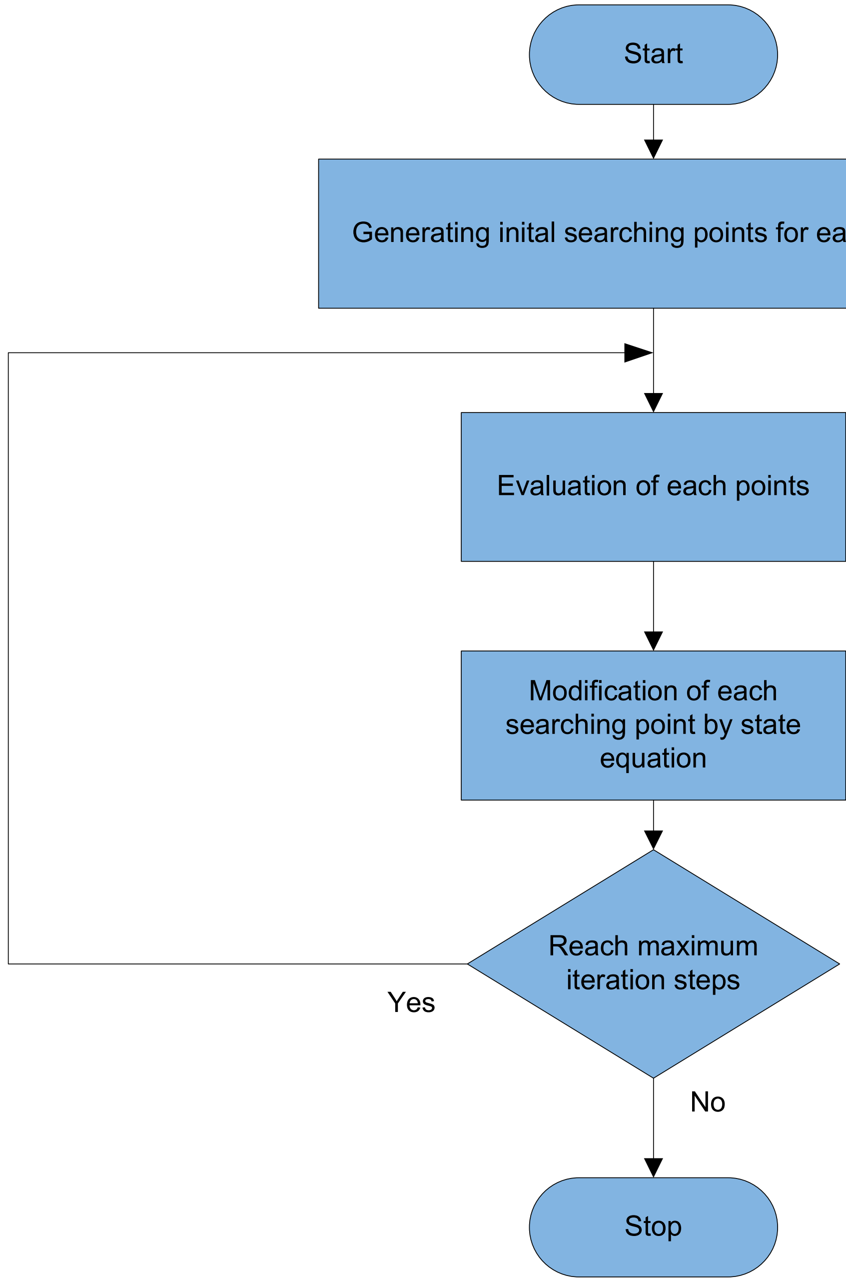


Fig. 40 Description of PSO algorithms

11.5 Results of particle swarm optimization

Finding the parameters for which the system becomes chaotic, it may seem at first glance a simple task. But we have to search the entire state space. However, we know that the system is very sensitive to initial conditions. This supposition greatly complicates the situation. Distribution of state space into regions is unnecessary, because the sensitivity of the parameters the solution may exist with many local extremes.

It is therefore necessary to define a proper fitness function. Because we know that the algorithm searches the minimum function, define the required function:

$$\Phi(x) = 10\sum_{\alpha=20}^{N}\lambda_1(\alpha) + \left|1000\sum_{\beta=20}^{N}\lambda_2(\beta)\right| + \sum_{\gamma=20}^{N}\lambda_3(\gamma) \qquad (28.)$$

The function $\Phi(x)$ looks complicated at first glance, but it is defined by sensitivity to the first expolent λ_1 and greater sensitivity to the second exponent λ_2, after which calls for close to zero. We see that the exponents α, β, γ starting up from number 20. This is because we did not have biased the results. We could say that, we wait until the trajectory of attractor is stabilized.

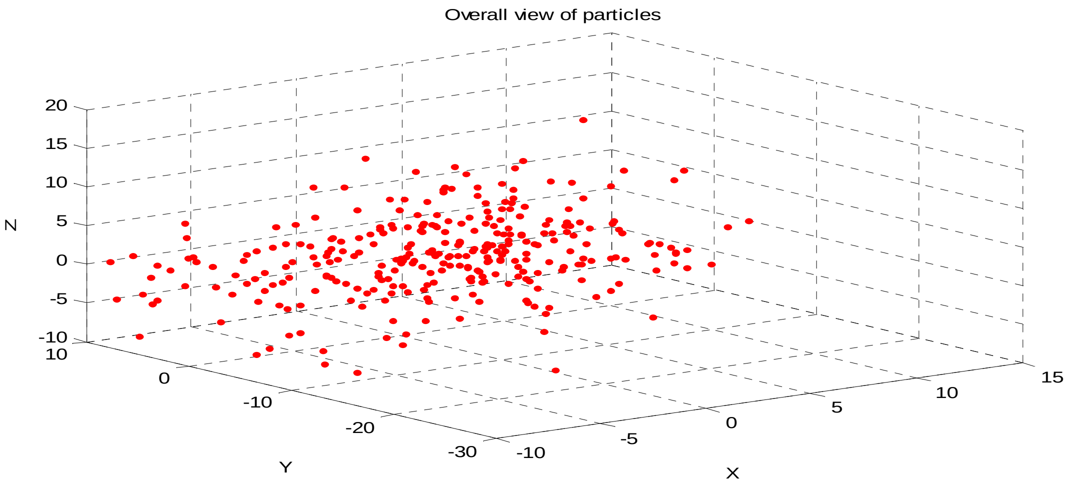


Fig. 41 Overal distribution of particles

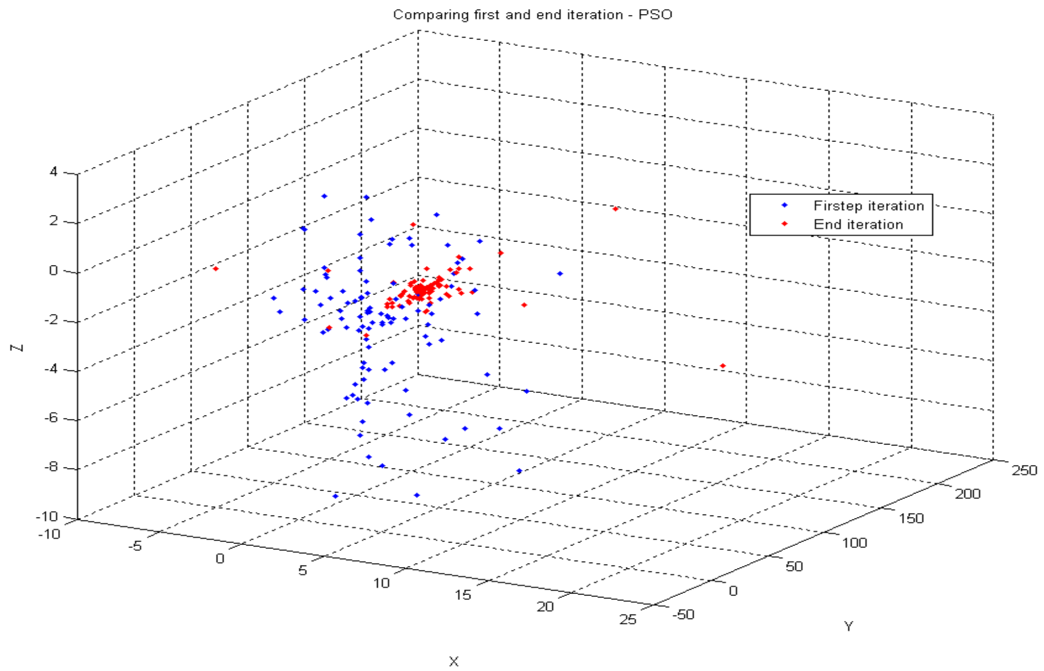


Fig. 42 Distribution of particles in first step (blue) and last step (red) of iteration

Again we had to, to create an algorithm, be careful to select the correct integration features. Speeding up the calculation could be achieved by splitting the state space of subspaces and the launch parallel computing - separated. For the control algorithm was used as generators known Lorenz system of equations (equation 10.) and a new system known as the Labyrinth chaos [24] (equation 29.).

$$\dot{x} = \sin y - bx$$
$$\dot{y} = \sin z - by \qquad (29.)$$
$$\dot{z} = \sin x - bz$$

For the first system, we found many solutions that were exhibiting chaotic behavior. Here are those that exhibit strong chaos:

p	r	b	λ1	λ 2	λ 3
55,49341	1,217994	3,67807	0,631487	0,038927	-6,57041
16,85988	1,105048	11,34285	0,39375	0,009977	-13,8537

Table 4. Solutions for Lorenz system

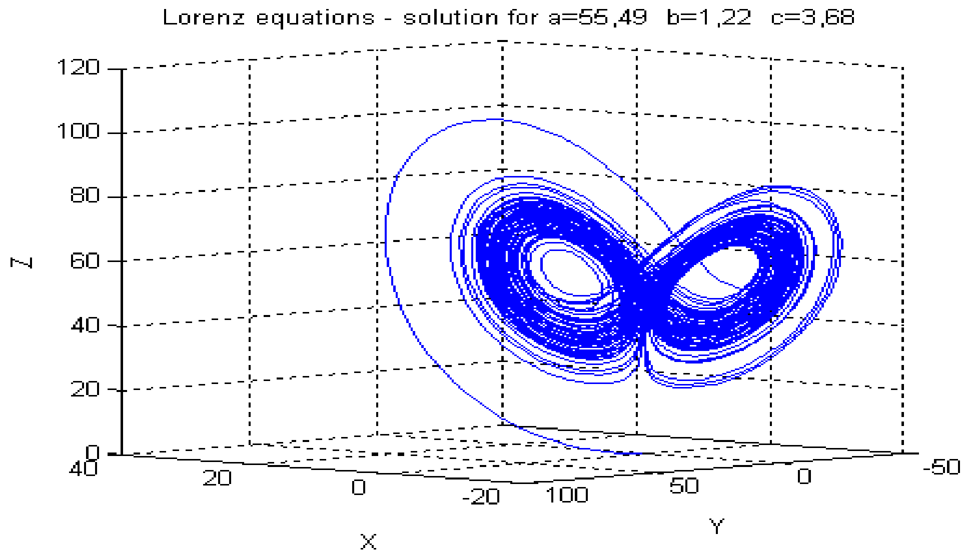


Fig. 43. Lorenz system for found parameters

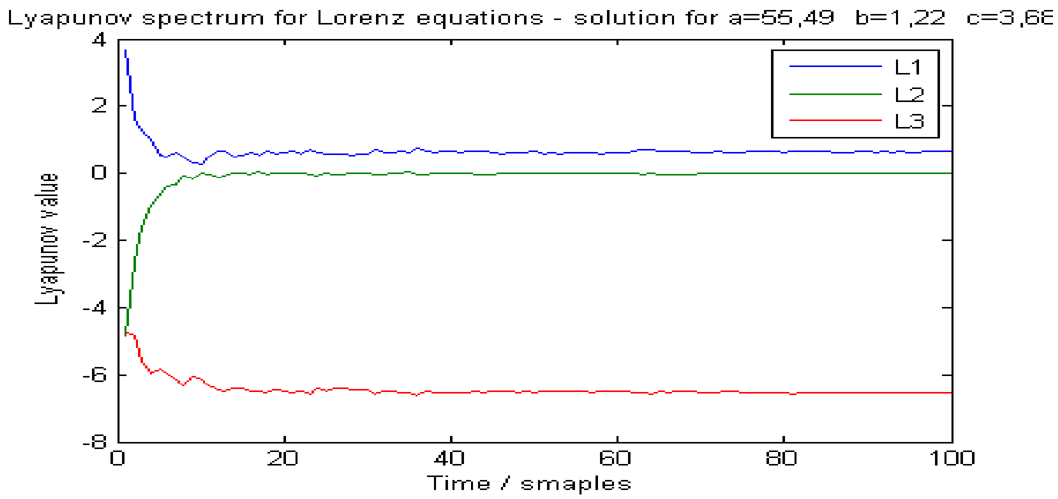


Fig. 44. Spectrum of exponents for solution

The second system exhibits weak chaos. By our algorithm we found 368 possible solutions for a quite short time. All solutions were correlation about b=0.1. What we know [24], this is a point of chaotic behavior.

11.6 Comparisons between Genetic Algorithm and PSO

From the procedure, we can learn that PSO shares many common points with GA. Both algorithms start with a group of a randomly generated population, both have fitness values to evaluate the population. Both update the population and search for the optimum with random techniques. Both systems do not guarantee success.

However, PSO does not have genetic operators like crossover and mutation. Particles update themselves with the internal velocity. They also have memory, which is important to the algorithm.

Compared with genetic algorithms (GAs), the information sharing mechanism in PSO is significantly different. In GAs, chromosomes share information with each other. So the whole population moves like a one group towards an optimal area. In PSO, only gBest gives out the information to others. It is a one -way information sharing mechanism. The evolution only looks for the best solution. Compared with GA, all the particles tend to converge to the best solution quickly even in the local version in most cases. This is the general method comparison. But we know that solution for our problem is not as easy. We know that the solution in real space **R** consists of many local sharp solutions. We can draw a solution, which each of methods suit for a different set of equations.

In future it would be interesting to combine PSO as generator for initial population for genetic algorithm. This can guarantee enough diversity for initial population. And can reduce time needed for computation.

12 SUMMARY

The aim of this work is to find the universal ways in which it is possible to investigate the dynamic system of differential equations. The methods that were used can be considered successful, because using them was found new solutions that meet the conditions. The proposed program in the Matlab language is universal. Just enter the general differential equations and their Jaccobi`s matrix and run the calculation. We have also tested other methods using a fractal dimension. This was useful, but it was very time consuming. Easiest to implement was box counting method. This method can be applied to time series without knowledge the equations. This method also was very slow. But it can be improved by using parallel computing. Another method was to use Poincare maps. Also quite simple method, but the problem is in analyzing the results. It cannot be easily decided, whether the results tend to be chaotic either not.

We have created algorithms in Matlab to calculate determine and find chaos in time series. The best showed up using Lyapunov exponents. How, in examining the issue turned out not needs the calculation of the spectrum, since the calculation of the greatest exponents is sufficient and simpler. These algorithms, we finally managed to create. We have implemented the calculation using the two best-known calculations.

Wolf's calculation, which appears to be relatively simple to mention, however, tends to be divergent. And second method is Rosenstein's. The algorithm we seemed much better and more stable. We tried to do an automated algorithm that works without prior knowledge of the system or issue. Unfortunately it was not possible to do. Algorithm seems to be very sensitive to the evaluation results. We must also choose the correct interval for the accuracy of the maximum Lyapunov exponent. The program underwent tests on models created by differential equations. We compared the results with the empirically calculated values. Indeed, we added a noise the signal, we tested the accuracy of the calculation. This proved to be very impressive.

Finally, we tested on real data. For testing we used Chua's oscillator and the real ECG signal. We were faced with the noise in the measured signal. He had to be removed before the actual implementation of the calculation. For Chua's oscillator exponent we walked around 1.3767, which already indicates a relatively high degree of chaos. And for ECG, we set its value to 0.74, which also indicates the presence of chaos.

The last object was to create an algorithm, which will find the system parameters for finding the chaos. We deal with several ways to find the parameters of dynamic differential equations, which exhibit chaotic behavior and are displayed in state space attractors. Best results in terms of delays, had multi-criteria genetic algorithm and Particle swarm optimization. When there is sufficiently large parental population and a large number of generations that should be sufficient for testing the system. However, it is recalled that it is not indifferent to the integration of numerical methods used. If the system integrate in time, it is possible that the solution is divergent. It can cause problems in the calculation. With great precision (meaning small numerical error) calculation may also be divergent and the computer will not be able to find a solution. The control in terms of accuracy and time cost is to use the integration method of Runge-Kutta 4th order. Interesting would be to use a fully parallel processing. This could greatly speed up the calculation (for example, using four-core processors up to 40%).

13 BIBLIOGRAPHY

[1] CARROL, T., PECORA, L. *Nonlinear Dynamics in Circuits*. WORLD SCIENTIFIC PUBLISHING,1995.

[2] WYK, M. A., STEEB, W. H. *Chaos in Electronics*. KLUWER ACADEMIC PUBLISHERS,1997.

[3] SPROTT, J. C. *Chaos and Time Series Analysis*. OXFORD UNIVERSITY PRESS, 2003.

[4] A. WOLF, J. B. SWIFT, H. L. SWINNEY, AND J. A. VASTANO, *Determining Lyapunov exponents from a time series,* PHYSICA D 16, 1985.

[5] ROSENSTEIN, M. T., COLLINS, J. J., LUCA C. J.*Reconstruction expansion as a geometry-based framework for choosing proper delay times. PHYSICA D 73:82--98. PHYSICA D, 1994* .

[6] SRINIVASAN S., PATIL S. A., PRASAD S., LAZAROU G., PICONE J. *Estimation of Lyapunov Spectra From a Time Series. IEEE,* 2006.

[7] FAN J., YAO Q., *Nonlinear Time Series*. SPRINGER, 2003.

[8] ZENG X., EYKHOLT R, PIELKE R.A., *Estimating the Lyapunov-Exponent Spectrum from Short Time Series of Low Precision.* PHYS. REV. LETT. 25, 1991.

[9] ROSENSTEIN, M. T., COLLINS, J. J., LUCA C. J., *A practical method for calculating largest Lyapunov exponents from small data sets*. *PHYSICA D*, 1992.

[10] KAPITANIAK T., BISHOP S.R., *The Illustrated Dictionary of Nonlinear Dynamics and Chaos.* JOHN WILEY & SONS, 1993.

[11] SMALL M., *Applied Nonlinear Time Series Analysis: Applications in Physics, Physiology and Finance.* WORLD SCIENTIFIC PUBLISHING COMPANY, 2005.

[12] KANTZ H., SCHRIEBER T., *Nonlinear Time Series Analysis.* CAMBRIDGE UNIVERSITY PRESS, 2004.

[13] THIELER J., *Efficient algorithm for estimating the correlation dimension from a set of discrete points*. PHYSICAL REVIEW A, 1987.

[14] ATAEI M., KHAKI-SEDIGH A, LOHMANN B., LUCAS C., *Estimating the Lyapunov exponent sof chaotic time series: A model based method*. BROOK HOUSE LIMITED, 2003.

[15] WILLIAMS G., *Chaos Theory Tamed*. JOSEPH HENRY PRESS. 1997

[16] KAPITANIAK T. A BISHOP S., *The Illustrated Dictionary of Nonlinear Dynamics and chaos*. JOHN WILEY & SONS LTD., 1999

[17] FALCONER K., Fractal *Geometry Mathematical Foundations & Applications*. JOHN WILEY & SONS LTD., 1990

[18] PETRZELA J. *Modelování zvláštních jevů ve vybraných nelineárních dynamických systémech,* DISERTAČNÍ PRÁCE VUT BRNO, 2006

[19] JIANQING F., QIWEI Y., *Nonlinear Time Series*, SPRINGER-VERLAG NEW YORK, INC., 2003

[20] HOME PAGE OF FRANZ-JOSEF ELMER
[Online] *http://elmer.unibas.ch/pendulum/chaos.htm*

[21] WOLFRAM MATHWORLD[Online] http*://mathworld.wolfram.com/*

[22] YALAMOVA R., QI L., WANG L., CHEN Y., MATHEWS S., GONG G., *Detecting chaos in financial time series*, US NATIONAL SCIENCE FOUNDATION, 2008

[23] ZENG X., EYKHOLT R., PIELKE R.A., *Estimating the Lyapunov-Exponent Spectrum from Short Time Series of Low Precision*. PHYSICAL REVIEW LETTERS VOLUME 66 - NUMBER 25, 1991 .

[24] SPROTT J.C., CHLOUVERAKIS K.E., *Labyrinth chaos*, WORLD SCIENTIFIC PUBLISHING COMPANY, 2007

14 ENCLOSURE LIST

CD with source codes in MATLAB:

- Poincare map
- Lyapunov exponents from ODEs
- Rosenstein’s method
- Wolf’s method
- Simple variation of parameters
- Optimizing by genetic algorithm
- Multi-objective optimizing by genetic algorithm
- Optimizing by particle swarm algorithm
- Spectrum of time series by FFT and filtering
- Reconstruction of dynamics by time delay method
- Autocorrelation for estimating tau
- Estimating Mutual information and correlation integral
- Estimating embedded dimension – symplectic geometry