Control of Chaotic Dynamical Systems using OGY

G. Witvoet

DCT 2005.36

Traineeship report

Coach(es): M. Steinbuch

Supervisor: L.N. Virgin (Duke University)
H. Nijmeijer

Technische Universiteit Eindhoven
Department Mechanical Engineering
Dynamics and Control Technology Group

Eindhoven, 31st March 2005

Summary

Chaos is the general name for non-linear dynamical systems which behave noise-like. Chaos is indecomposable, is highly dependent on the initial condition and consists of a large number of periodic points and orbits. Because of this, the solution of a chaotic system is difficult to predict, which calls for a way to control it. The control algorithm of Ott, Grebogi and Yorke (OGY, [13]) manages to do this.

The basic observation behind OGY is that a chaotic attractor has a large number of unstable periodic solutions embedded within itself. Furthermore, by slightly perturbing an accessible parameter of the system, it is possible to push the system towards one of these orbits. Since chaos behaves ergodically, at some point in time the solution will come into the vicinity of a certain point of the orbit where a linearization is valid. With this linearization a simple pole placement method can be used to calculate a control effort to direct the system towards this point and thus the orbit.

OGY is a discrete control algorithm, perturbing the system at discrete moments in time. OGY can therefore easily be used on discrete systems like the Hénon map. By first discretizing a system using the Poincaré map it can also be used on continuous systems like the Duffing oscillator. Periodic orbits will become a simple sequence of points on the Poincaré map (m points for a period m orbit), around which the OGY algorithm finds a linearization.

The accuracy of this linearization is very important in the implementation of OGY. In cases where no model is present the values of the periodic points should be estimated using the recurrence method and the matrices A and B in

$$x_{i+1} - x^*_{i+1} = A_i \left(x_i - x^*_i\right) + B_i \left(p_i - \bar{p}\right)$$

should be estimated using linear least squares and data points in a small vicinity around x^*_j. Better results are obtained when this estimation is performed iteratively; the fixed point estimate $\bar{x}$ is adjusted and the matrix A recalculated until the offset C in

$$x_{i+1} - \bar{x}_{i+1} = A_i \left(x_i - \bar{x}_i\right) + C$$

is close to zero. The matrix B is calculated with these new values of $\bar{x}$ and A.

The matrices A and B are then used to find the control matrix K^T with which the control effort (or parameter perturbation) during simulation can be calculated:

$$p_i - \bar{p} = -K_i^T (x_i - x^*_i).$$

When this perturbation is bigger than a limit δ, it is set to zero, so that a vicinity around $\bar{x}$ is defined. Applying this on Hénon and Duffing indeed results in controlled states, for different periodic orbits. Observations during these simulations include: a steady state control effort

when using estimations, time to achieve control differs per orbit and per simulation, maximum perturbation δ influences time to achieve control and number of false control attempts. OGY is able to overcome some amount of noise and is therefore quite robust. When noise is present during estimation though, it could cause poor estimates and therefore no control.

When only one variable of the system can be measured, delay coordinates can be used to reconstruct the chaotic attractor (and its Poincaré map). Since previous control efforts could still have a direct effect on the current state of the system, the OGY algorithm should be adjusted for this. Implementation of this new algorithm caused problems though. Therefore an adjustment is made so that the parameter perturbation is applied in less time, removing the necessity of adjusting the OGY algorithm itself. Simulation indeed proved the increased performance for both period 1 and 2 orbits.

As soon as all the estimates are made, OGY can also be implemented in Simulink®, which makes it easier to play around with the system. Therefore two models, for both period 1 and period 2 orbits, were made, which show results comparable to the previous results.

Summarizing, OGY seems to work on any system, discrete and continuous, with or without delay coordinates, as long as the error during estimation is not too big. The influence of the maximum parameter perturbation was shown, e.g. on the time to achieve control.
Not needing a model makes OGY attractive to use. However, there are some major drawbacks of OGY. For example, control is restricted to one of the embedded orbits, time to achieve control is unpredictable and could be very large, and the eigenvectors are changed after control implementation.

Recommendations considering OGY include taking different estimation methods for the periodic points and the matrices A and B. One can think of Partial Least Squares in case there is correlation present, or using a non-linear least squares method (e.g. quadratic) in order to shorten the time to achieve control by increasing the control vicinity. Furthermore, it should be investigated whether OGY can be written in continuous form, so that control can be applied around any point of the orbit instead of just one, thereby decreasing the time to achieve control even further. Targeting techniques could also be used for this purpose.

Acknowledgements

This report is the result of an international internship I've done in the year 2004. I had the privilege of spending three months in Durham NC, at Duke University, at the Department of Mechanical Engineering. Besides the research on the control of chaos I've done there, I was able to meet lots of people and see a lot of things in the USA. This gives me reason to be thankful.

First of all my thanks goes to prof Lawrence Virgin, supervisor of my internship at Duke. I owe the opportunity of this internship to him, for he accepted my request and invited me to come to Duke for this assignment on OGY. I'm grateful for the confidence he had in me and for all the help he provided regarding my assignment.
Thank also to prof Maarten Steinbuch and prof Henk Nijmeijer, my supervisors at the Technische Universiteit Eindhoven (TU/e) in the Netherlands, who were able to give directions by replying to the many questions I've sent by e-mail.
Regarding the statistical part of this report, I've received some help from prof Alessandro Di Bucchianico and Rens Kodde from the TU/e and even from my roommate in the USA, Philippe Lűdi. Thanks to you also.
Thanks also to Marten Voogt, friend and inmate, for letting me 'use' his fast computer for my time consuming calculations, since my own computer was too slow.

Besides these people I'm also very thankful to the people who were just there for me and supported me. I'd like to thank the Navigators for their friendship, BBQ's and quality time. And of course for the awesome beachtrip.
I'd also like to thank my labmates from "The Überoffice", Ryan Greer, Michael Hunter and Sophia Santillan, for their friendship and for helping me out every once in a while.

Finally, my thanks goes to my wife Sietske, who was there with me, supported me and never doubted me. And of course my thanks to my God and Savior Jesus Christ, for His protection, comfort and wisdom.

Contents

Chapter 1

Introduction

Ever since the beginning of mankind, people are fascinated by movements, like the movement of the sun along the sky or the falling of objects towards the ground. Through the years mankind has learned that these movements are generated and influenced by forces. This knowledge has resulted into the research field called *dynamics*: the study of bodies in motion and the forces acting on them. Increased insight in dynamics during the past centuries finally has paved the way for *control technology*. This latter research field manages to regulate movements studied by dynamics, and has developed very fast in the 20th century due to the invention of the computer.

In dynamics there is a general distinction between linear and non-linear dynamics. Although most dynamical systems in reality appear to be non-linear, most of the emphasis of dynamic research is put on linear systems. Linear research provides basic insight in dynamics and control, mainly because it is much easier to understand than non-linear systems. This results in the fact that dynamical analysis and control of linear systems is far more developed than non-linear systems.

This report will deal with a special kind of non-linear dynamic behavior, so called *chaos*. Main focus will not be on the analysis but on its control. Furthermore, it will only concentrate on situations where the dynamics of the system is unknown, so no model is present. It will be assumed that there is only some amount of experimental data of the system. This restriction (the absence of a model) will limit the control possibilities; e.g. control will only be possible towards certain fixed points or orbits. The goal of this report is to examine the possibility of controlling a certain experimental chaotic dynamical system towards one or multiple (unstable) orbits, by just measuring one state variable of the system. However, no actual experiment is done for this, the data used is just numerical.

In order to reach its goal, this report will use an algorithm designed by Edward Ott, Celso Grebogi and James Yorke to control chaos. These authors introduced their so called OGY algorithm in 1990, which was further improved by various authors in the following years. Chapter 2 will give an introduction to chaotic dynamical systems and some basic control theory. Chapter 3 will discuss the basics of the OGY algorithm. In the following chapters this algorithm is implemented in both discrete and continuous systems. Finally, chapter 6 will discuss the results and reflect on them.

Chapter 2

Background

Before examining and applying the algorithm for chaotic dynamical systems, it is useful to discuss some background. This chapter will therefore give some background on chaotic systems (section 2.1, see also [2], [21] and [22]) and some elementary control theory (section 2.2).

2.1 Chaotic dynamical systems

2.1.1 Definition and properties

What exactly is a chaotic dynamical system? In general, the equation of a dynamical system can be written in the form

$$\dot{x} = f(x, u) \qquad \text{or} \qquad x_{i+1} = f(x_i, u_i), \tag{2.1}$$

where x is the state vector and u the input vector of the system. If the function f is non-linear in x and/or u, the system is also non-linear. A chaotic system is a special kind of non-linear system, characterized by its chaotic behavior. However, this behavior cannot be recognized by just looking at the equation of the system, no matter how simple that equation is. One can only recognize chaotic behavior by analyzing the system response, which can be highly complicated and noise-like. Due to its complexity, there is no common definition of chaotic systems, but Devaney [2] manages to define chaos quite well using the following three properties:

1. A chaotic system has sensitive dependence on initial conditions.
2. A chaotic system is topologically transitive.
3. Periodic orbits and/or points are dense in the chaotic attractor.

In other words, a chaotic system is unpredictable, since its response will highly depend on the initial condition; a small perturbation can result in a completely different response. Topological transitivity means that the system is indecomposable; it cannot be broken down into two subsystems. Finally, a chaotic system also has an element of regularity, since it is composed by a large number of periodic orbits and points embedded within the system. A typical chaotic response contains an infinite number of periodic orbits, creating an aperiodic solution, but the response can also be a simple periodic motion. Again, this depends to a large extend on the initial condition. Further explanation can be found in [2].

A chaotic system can only be analyzed by observing the responses for a large set of different initial conditions, since a single solution, aperiodic or not, cannot fully describe the chaotic system. Because of this, only a few systems are actually proven to be chaotic, like the Hénon map of chapter 4. For the other systems used in this report, chaos is very plausible.

Although a typical aperiodic response of chaos looks like noise, its structured behavior clearly shows the contrary. This structure is called *ergodic behavior*. When the system wanders through space, at a certain time t_1 it crosses point x_1. A time t_2 after t_1 it will come into some vicinity ε of the point x_1 again. The same will happen a time t_3 after t_2, etc. In general, the smaller this vicinity ε is defined, the bigger $t_2, t_3, \ldots$ will be. But although the solution will come very close to x_1, it will never become completely equal to it (with $\varepsilon \to 0$, $t_i \to \infty$). This ergodic behavior explains the so called *strange attractor* which will arise; a certain subspace in space inside which the solution of the system will wander around.

2.1.2 Chaotic examples

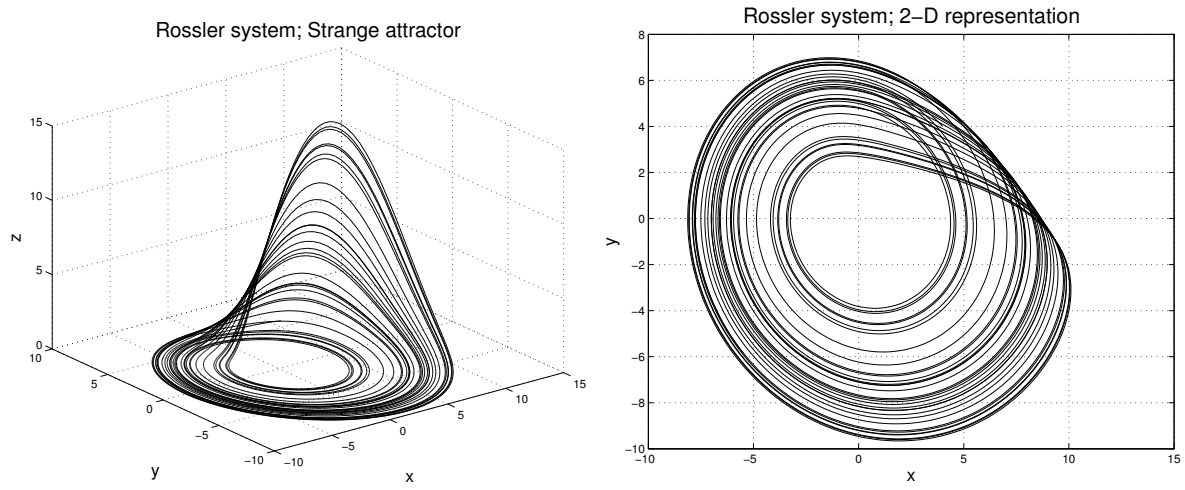
To illustrate the chaotic properties mentioned in the previous section, this section will shortly discuss two examples: the *Rössler system* and the *Duffing oscillator*.

2.1.2.1 Rössler system

In 1976 Otto E. Rössler found a very simple 3-D continuous chaotic system, including its attractor[17]. This system is defined by the following set of differential equations:

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

where a, b and c are parameters of the system. The equation contains only one simple



(a) 3-D representation

(b) 2-D representation in x and y

Figure 2.1: Response of Rössler system with initial condition (4,4,1)

non-linearity, namely xz. Still, for some initial conditions the 3-D solution (x, y, z) of the system clearly shows a strange attractor; see figure 2.1 where $a = 0.2$, $b = 0.4$ and $c = 5.7$. The solution neither converges to a stable point or a periodic orbit, nor does $x, y, z \to \infty$. Instead, the system wanders around within a certain subspace, never crossing itself, creating an aperiodic solution. This strange attractor shows the ergodic and thus chaotic behavior of the system. It should be noted though, that for some initial conditions the solution does become periodic, and for other initial conditions $x, y, z \to \infty$.

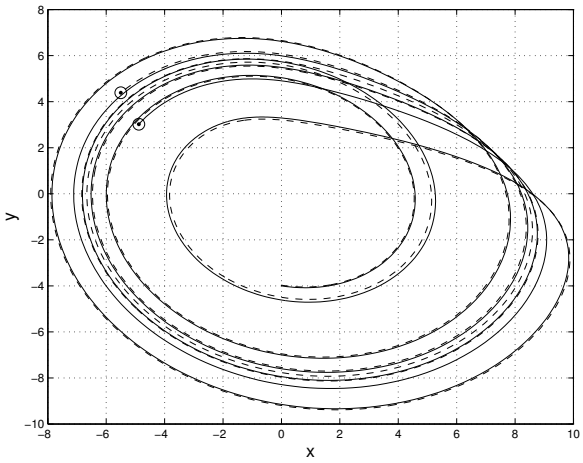


Figure 2.2: Dependence on initial conditions for (2.2); $a = 0.2$, $b = 0.4$, $c = 5.7$

The dependence on initial conditions is shown in figure 2.2. One curve has initial condition $(0, -4, 0)$ and the other has an initial condition very close to this. However, both solutions follow a different path and after $40s$ of simulation (indicated by the encircled marks) both solutions clearly differ.

2.1.2.2 Duffing oscillator

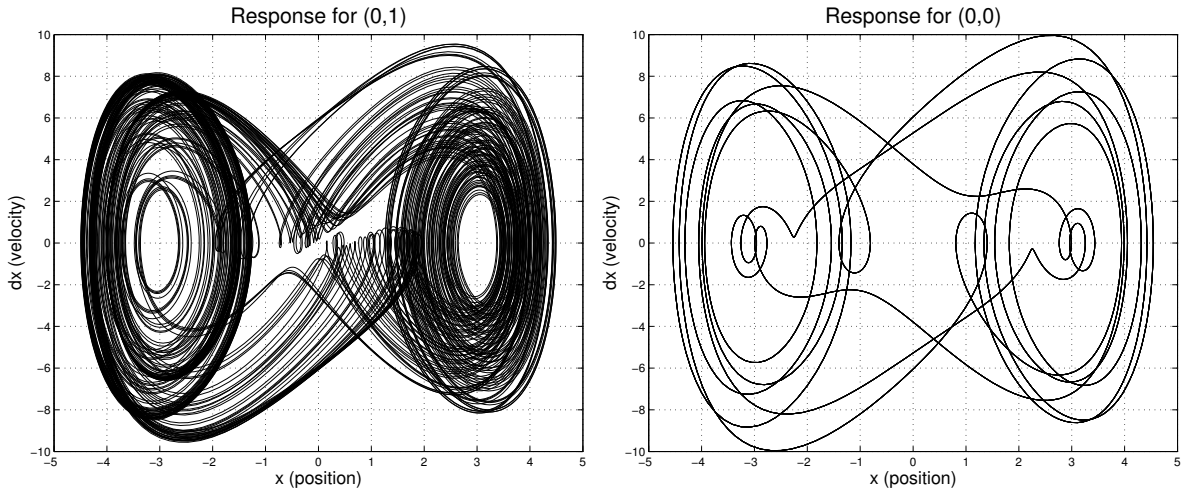
The example of Rössler is just a theoretical case meaning that the equation doesn't represent an actual system. A system which does arise in reality is the Duffing oscillator [5]:

$$\ddot{x} + \delta\dot{x} + (\beta x^3 \pm \omega_0^2 x) = \gamma \cos(\omega t + \phi). \tag{2.3}$$

Georg Duffing introduced this non-linear equation in 1918 to model the vibration modes of a beam with periodic forces acting on it. With certain parameter choices this system can behave in a chaotic way, meaning that the solution can be aperiodic. This behavior depends on the initial condition though. This is illustrated by the following Duffing oscillator:

$$\ddot{x} + d\dot{x} + x + x^3 = f \cos \omega t, \tag{2.4}$$

where $d = 0.2$, $f = 36$ and $\omega = 0.661$ (see [4]). When the initial condition is set at (0,1), the response of the system looks chaotic, as can be seen by the strange attractor in figure 2.3(a). However, figure 2.3(b) shows that the system behaves completely different when the initial condition is set on e.g. $(0, 0)$. It seems that the solution is some kind of higher periodic orbit.

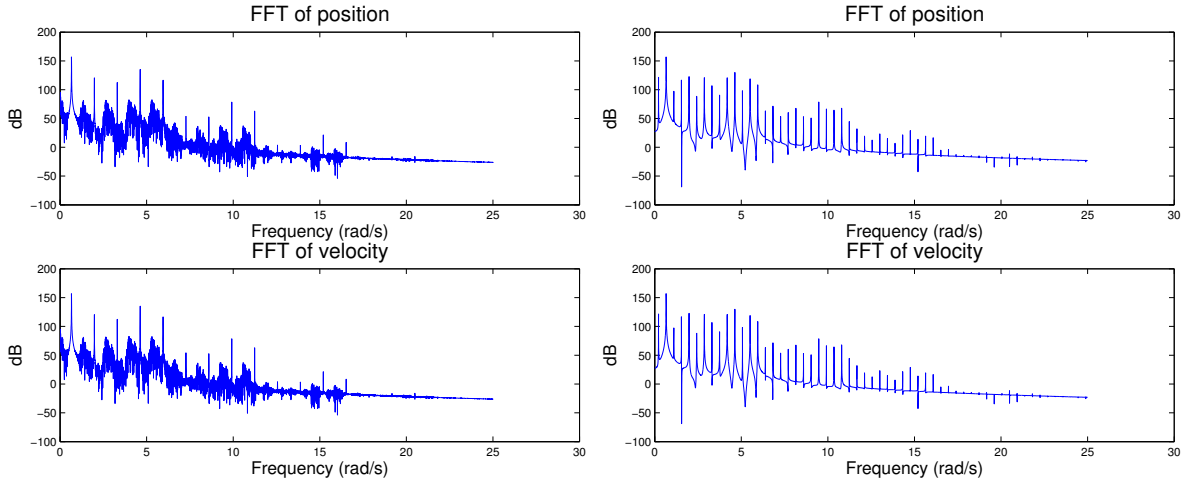


(a) Initial condition: (0,1)

(b) Initial condition: (0,0)

Figure 2.3: Response of (2.4); the first $500s$ of transient behavior is removed

One way to investigate the (a)periodicity of a solution is by making a Fourier transform (FFT), shown in figure 2.4. The difference is obvious: the first solution shows a very noisy and continuous FFT, whereas the second clearly is discrete. It has sharp peaks at $\frac{1}{3}\omega, \omega, 1\frac{2}{3}\omega, 2\frac{1}{3}\omega$, etc, in other words, $\frac{1}{3}\omega + k\frac{2}{3}\omega$, where $k \in \mathbb{Z}$. This shows that the latter attractor only contains a finite number of frequencies and is thereby a periodic attractor. The first trajectory is clearly aperiodic since its FFT shows that it contains all frequencies. It should be said though, that



(a) Initial condition: (0,1)

(b) Initial condition: (0,0)

Figure 2.4: Fourier transform of figure 2.3

the existence of an aperiodic solution doesn't prove that the system itself is chaotic (see the definition of chaos in section 2.1.1).

2.1.3 Characterizing chaos

The previous section already introduced the FFT method to investigate chaotic behavior. This section will shortly introduce some more methods to detect and characterize chaotic behavior.

2.1.3.1 Bifurcation diagram

Non-linear systems are often subject to a phenomenon called *bifurcation*: the dynamic behavior of the system changes due to changes in one of the parameters. For example, the number of fixed points or (in)stable periodic orbits can depend on the value of a single parameter. The parameter value where this change occurs is where the bifurcation arises.

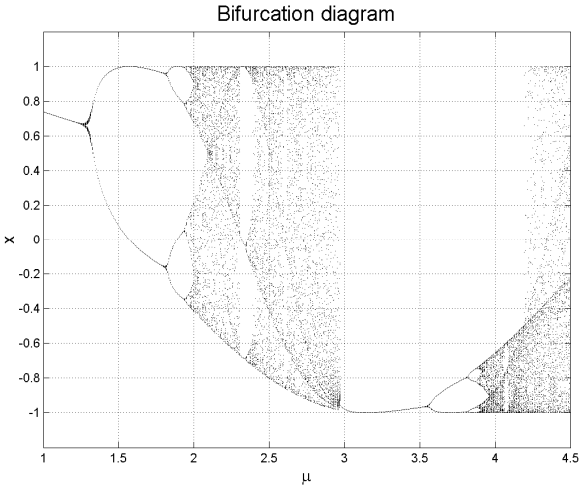


Figure 2.5: Bifurcation diagram of $x_{n+1} = \cos(\mu x_n)$; 40 points are plotted at each μ

The easiest way to illustrate bifurcations is by looking at a discrete system $x_{n+1} = f(x_n, \mu)$, where μ is the parameter, for example:

$$x_{n+1} = \cos(\mu x_n). \tag{2.5}$$

By iterating (2.5) lots of times at a certain parameter value, one can find periodic orbits: x_n will only take a finite number of values. By plotting these points for different values of μ a so called bifurcation diagram is obtained (figure 2.5). For example, at $\mu \approx 1.3$ the single equilibrium changes to a period two orbit and at $\mu \approx 1.8$ the orbit becomes period four. These points are therefore (period doubling) bifurcation points. At $\mu \approx 2.3$ and $\mu \approx 2.9$ there is an infinite amount of periodic orbits with different periods, or in other words, chaos is present.

2.1.3.2 Poincaré maps

For autonomous continuous systems, chaos can only occur in systems of third order or higher (see [21]). The response for a third order system is hard to draw though, since it needs

a 3-D plot to fully capture its behavior. Higher order systems will even need more dimensions. It is therefore useful to make plots of lower dimension which are able to show part of the dynamic behavior. A common method to do this is by making a Poincaré map (figure 2.6). It is named after Henri Poincaré, who introduced this map in 1881 (see [15]).

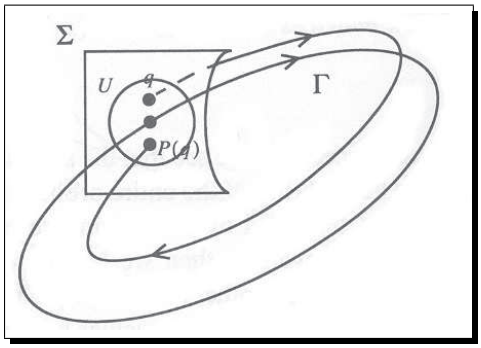


Figure 2.6: Poincaré map; from [18] page 294

Suppose an n-dimensional system has a solution which intersects a certain hypersurface of dimension n–1. The points of intersection where the solution has the same direction then form the Poincaré map. It is called a map since it *maps* each intersection point (x_j) into the next (x_{j+1}). Of course, the resulting Poincaré map will depend on the choice of the hypersurface. A periodic orbit will become a single point in the Poincaré representation, and a chaotic attractor will be characterized by a certain shape. See figure 2.7 where the Poincaré map of the Rössler system (2.2) is drawn, with $a = 0.2$, $b = 0.4$ and $c = 5.7$. The hypersurface is formed by $x = 0$. For e.g. the Duffing oscillator one can also choose a constant phase as the hypersurface, which means sampling with the same period as the acting force. It is clear that the Poincaré map not only decreases the dimension with one, it also turns a continuous time system into a discrete time system, with $x_{j+1} = f(x_j)$.

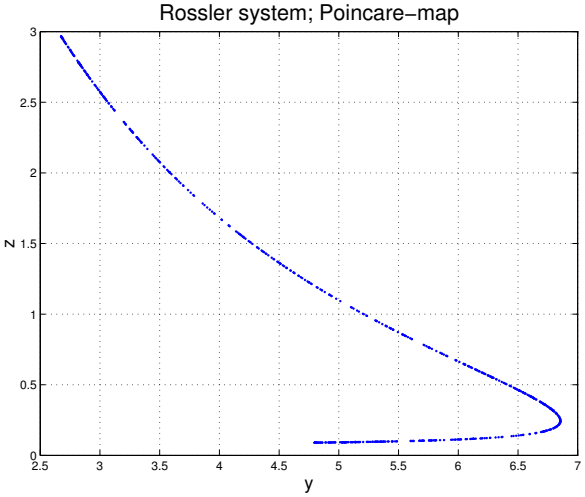


Figure 2.7: Poincaré map of figure 2.1(a); hypersurface $x = 0$, y positive

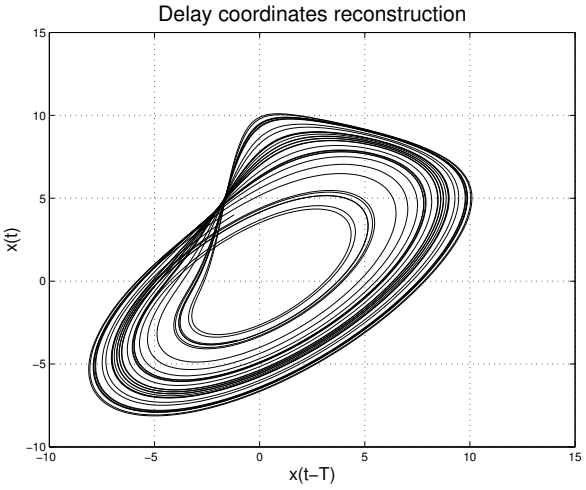


Figure 2.8: Reconstruction of figure 2.1(b), using $x(t), x(t-\tau)$; $\tau = 0.88$

2.1.3.3 Delay coordinates

During experiments it is sometimes not possible to measure all n state components, but only one or two. Still it might be useful to construct the n-dimensional chaotic attractor in order to analyze the system and its chaotic behavior. In their article [14], Packard et al. manage to do this using just one state component (say, x) by using the delay coordinates method: the attractor is reconstructed by defining n new states $x(t), x(t-\tau), x(t-2\tau), \ldots, x(t-(n-1)\tau)$,

where τ is a chosen delay time [11]. The resulting attractor can then be used for further analysis, e.g. to make a Poincaré map. Compare the delay coordinates reconstruction of the Rössler system with its original in figure 2.8.

2.2 Elementary control theory

Before chapter 3 will discuss a control algorithm for chaotic dynamical systems, this section will give a short introduction to some basic control theory. Only the so called *pole placement* method will be discussed, since this is the basic method the control algorithm of chapter 3.

2.2.1 Pole placement

The pole placement control method is probably one of the easiest control laws and therefore also quite easy to understand. It is a linear control method working well on linear systems. Since non-linear systems can normally be linearized around a equilibrium point, it is also applicable to a wide variety of non-linear systems.

2.2.1.1 Linear systems

In order to apply pole placement, a system should be expressed in state space notation:

$$\dot{x} = Ax + Bu \qquad \text{for continuous systems,} \tag{2.6a}$$

$$x_{j+1} = Ax_j + Bu_j \qquad \text{for discrete systems.} \tag{2.6b}$$

Here A is the $(n \times n)$ system matrix, B the $(n \times m)$ input matrix, x the $(n \times 1)$ state of the system and u the $(m \times 1)$ input. For continuous cases, system (2.6a) is stable when all the eigenvalues λ_i of A are negative, or $\Re(\lambda_i) < 0$. For discrete cases, system (2.6b) is stable when the magnitude of every eigenvalue λ_i of A is smaller than 1, or $|\lambda_i| < 1$.

When system (2.6) is unstable, an input u is needed to stabilize the system. The following pole placement technique will manage to do this when the system is controllable. This means that the controllability matrix P defined by

$$P = [B, AB, A^2B, \ldots, A^{n-1}B] \tag{2.7}$$

has to have full rank n. When this is the case, a so called state feedback $u = -Kx$ or $u_j = -Kx_j$ can be defined, so that (2.6) can be written as:

$$\dot{x} = Ax + Bu = Ax - BKx = (A - BK)x. \tag{2.8}$$

Now the matrix $(A - BK)$ is the new system matrix, so the stability of the controlled system is now determined by the eigenvalues of $(A - BK)$ and will depend on K. The eigenvalues, or *poles*, of the system can thus be *placed* anywhere by choosing an appropriate K. By choosing these poles according to $\Re(\lambda_i) < 0$ (for continuous systems) or $|\lambda_i| < 1$ (for discrete systems) the system will become stable.

Stabilizing a linear system is thus translated into finding K corresponding to the chosen poles. One way to do this is by using Ackermann's formula (see [1]). This algorithm is also implemented in Matlab® (command `acker`).

2.2.1.2 Non-linear systems

Non-linear systems can also be written in state space notation:

$$\dot{x} = f(x, u) \qquad \text{for continuous systems,} \tag{2.9a}$$

$$x_{j+1} = f(x_j, u_j) \qquad \text{for discrete systems.} \tag{2.9b}$$

Now suppose system (2.9) has an equilibrium point (x^*, u^*), for which holds that $\dot{x}^* = 0$ or $x^*_{j+1} = x^*_j$ resp. Then around x^* system (2.9) is linearized by defining:

$$A = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}_{(x^*,u^*)} \quad \text{and} \quad B = \begin{pmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_m} \end{pmatrix}_{(x^*,u^*)}, \tag{2.10}$$

both evaluated at the equilibrium point (x^*, u^*). Then around this equilibrium point we can approximate system (2.9) by using (2.6).

Finding an appropriate control law $u = -Kx$ is then the same as in the previous paragraph. There is one essential difference though. For linear systems the control law will control any point $x \in \mathbb{R}$, whereas for non-linear systems the control law will only work for x sufficiently close to x^*, since the linearization (2.10) is only valid in the vicinity of the equilibrium point.

Chapter 3

The OGY control theorem

3.1 Introduction

The name OGY comes from its inventors: Edward Ott, Celso Grebogi and James Yorke. In 1990 they published an article [13], showing that it was possible to control chaos, and thereby being the first to achieve this with reasonable control efforts. Ott, Grebogi and Yorke based their theory on recent articles showing that a chaotic attractor has a large number of unstable periodic orbits embedded within it [11]. The essence of the OGY theory is simply to stabilize one (or more) of these orbits by applying small perturbations. To apply these perturbations one of the parameters of the system should be accessible, meaning that this parameter can be adjusted while the system is running. This parameter thus becomes the input of the system.

Ott, Grebogi and Yorke based their theory on just experimental data of the chaotic attractor, without having a priori analytical knowledge of the system. The noise-like behavior of the data causes modelling of a chaotic system to be almost impossible. OGY can only be applied to discrete data, so continuous time systems should first be made discrete time by using e.g. the Poincaré map. In case only one state variable can be measured, the OGY theory could be combined with the delay coordinates method [14].

The method presented by Ott, Grebogi and Yorke still needed some modifications though. While their article [13] was only theoretical and used a discrete time numerical Hénon map example, Ditto et al. were the first to successfully implement the OGY theory in a real experimental system [3], namely a parametrically driven magnetoelastic ribbon. They used an article by Eckmann and Ruelle [6] to approximate the dynamic behavior in the vicinity of the desired orbit.

Dressler and Nitsche discovered a difficulty of OGY when they implemented it using delay coordinates. In their article [4] they proposed an important change to OGY, making it applicable to any experimental chaotic system.

Romeiras et al.[16] wrote an important article in further understanding of the OGY theory. They restated the algorithm and showed that OGY is nothing more then an ordinary pole placement technique (discussed in section 2.2), making the algorithm easier to implement. Furthermore the OGY algorithm was also implemented to control not only fixed points, but also periodic orbits.

The articles [20] and [7] continue the article of Romeiras et al. by applying the pole placement technique to the control of fixed points and orbits using delay coordinates, making

OGY applicable to every situation: fixed points as well as periodic orbits, discrete time maps as well as continuous time systems discretized by Poincaré and delay coordinates.

Other adjustments, extensions and discussions of the OGY theory are discussed in [12], [19], [9] and [10].

3.2 Description of the OGY algorithm

As said before, the basic idea behind the OGY algorithm is that there are a large number of unstable periodic orbits embedded within a chaotic attractor. By perturbing a certain accessible parameter, it is possible to stabilize one (or more) of these orbits. But how does it work? This section will introduce the mathematics of the OGY algorithm, mostly based on articles [16], [20] and [7].

As was mentioned in section 3.1, OGY control is only applicable to discrete time systems. In case of a continuous time dynamical system, this system should first be converted into a discrete time system by using e.g. Poincaré maps (section 2.1.3.2). This way a n-dimensional system can be written in the form:

$$x_{i+1} = f(x_i, p), \tag{3.1}$$

where $x_i \in \mathbb{R}^n$ is the n-dimensional state of the system at iteration i and p is the accessible parameter mentioned in section 3.1. For the parameter p a nominal value $\bar{p}$ is defined, for which the system has a chaotic attractor. Since the perturbation of p is assumed to be small, the value of p is restricted:

$$|p - \bar{p}\,| < \delta. \tag{3.2}$$

A fixed point of the discrete time map (3.1) with $p = \bar{p}$ satisfies $x^*_{i+1} = x^*_i$, which is a period one orbit if the real system is described in continuous time (assuming the sampling is done once per period). Similarly, a period two point for the discrete time map, or period two orbit for continuous time systems, corresponds to $x^*_{i+2} = x^*_i$. In general, a period T point or orbit satisfies $x^*_{i+T} = x^*_i$. Of course, for any $q \in \mathbb{Z}$ this also means that $x^*_{i+T+q} = x^*_{i+q}$. For understanding, the implementation of OGY for period one and higher periods is discussed separately in the following sections.

3.2.1 Fixed points

First the case $x^*_{i+1} = x^*_i = x^*$ is considered, where x^* is an unstable fixed point embedded within the chaotic attractor. Since the system is ergodic (section 2.1.1) the state x_i will come very close to this point at some point in time, while $p = \bar{p}$. When this is the case, system (3.1) can be linearized around x^*:

$$x_{i+1} - x^* = A(x_i - x^*) + B(p - \bar{p}), \tag{3.3}$$

where A is the Jacobian and B represents the influence of the 'input' p (for $j, k = 1, 2, \ldots, n$):

$$A = \frac{\partial f}{\partial x}(x^*, \bar{p}) = D_x f(x, p) \tag{3.4a}$$

$$B = \frac{\partial f}{\partial p}(x^*, \bar{p}) = D_p f(x, p), \tag{3.4b}$$

both matrices evaluated at $x = x^*$ and $p = \bar{p}$. Then a state feedback can be defined, similar to section 2.2:

$$p - \bar{p} = -K^T(x_i - x^*), \tag{3.5}$$

so that

$$\begin{aligned} x_{i+1} - x^* &= A(x_i - x^*) + B(p - \bar{p}) \\ &= A(x_i - x^*) - BK^T(x_i - x^*) \\ &= (A - BK^T)(x_i - x^*). \end{aligned} \tag{3.6}$$

Stability is now determined by the eigenvalues of $(A–BK^T)$. As said in section 2.2 the system is stable when $|\lambda| < 1$.

When the system is controllable, the poles λ can be placed anywhere by calculating the corresponding K^T (with Ackermann's method [1]). The question is, how should these poles be chosen in OGY control? Since x^* is unstable, matrix A will have n_u unstable eigenvalues and n_s stable ones (with $n_u+n_s=n$). Then choose n_s poles of $(A–BK^T)$ equal to the n_s poles of A and choose the remaining n_u poles equal to zero, which corresponds to superstability. When the OGY algorithm was first introduced, it was believed that this would cause the stable eigenvectors to remain unchanged. But although the eigenvalues n_s are still the same as before, the stable directions do change, since the eigenvectors of $A - BK^T$ are different then those of A.

With these desired poles one can find K^T, with which the parameter perturbation as defined in (3.5) can be found. One should keep (3.2) in mind though. Therefore, when

$$\left|K^T(x_i - x^*)\right| \geq \delta$$

the control is set to zero, so that $p = \bar{p}$.

3.2.2 Higher periodic orbits

The OGY algorithm for orbits (or points) with period larger than one is slightly more complicated. Suppose there is an orbit of period T embedded within the chaotic attractor, meaning that $x^*_{i+T} = x^*_i$ for any i. So for every iteration (or intersection with the Poincaré map) we can linearize the system, finding T different matrices A_i (the Jacobian) and B_i:

$$A_i = A_{i+T} = \frac{\partial f}{\partial x}(x^*_i, \bar{p}) = D_x f(x, p) \tag{3.7a}$$

$$B_i = B_{i+T} = \frac{\partial f}{\partial p}(x^*_i, \bar{p}) = D_p f(x, p), \tag{3.7b}$$

evaluated at $x = x^*_i$ and $p = \bar{p}$. At each iteration i the linearization is now given by:

$$x_{i+1} - x^*_{i+1} = A_i\,(x_i - x^*_i) + B_i\,(p_i - \bar{p})\,. \tag{3.8}$$

Furthermore, at the unstable orbit of period T (with $p=\bar{p}$) it is known that

$$x^*_{i+T} = f\left(x^*_{i-1+T}, \bar{p}\right), x^*_{i-1+T} = f\left(x^*_{i-2+T}, \bar{p}\right), \ldots, x^*_{i+1} = f\left(x^*_i, \bar{p}\right) \tag{3.9}$$

In order to determine the stability of this orbit, one can investigate the propagation of a small error ε. This can be done by using the Taylor expansion:

$$f(x_i + \varepsilon, \bar{p}) = f(x_i, \bar{p}) + D_x f(x_i, \bar{p})\varepsilon + O(\varepsilon^2) \approx f(x_i, \bar{p}) + D_x f(x_i, \bar{p})\varepsilon. \tag{3.10}$$

After T iterations, using $D_x f(x_i) = A_i$, this then leads to:

$$\begin{aligned} f(x_1 + \varepsilon, \bar{p}) \approx \quad & f(x_1, \bar{p}) + D_x f(x_1, \bar{p}) \quad && = x_2 + A_1\varepsilon && (3.11a)\\ f(x_2 + A_1\varepsilon, \bar{p}) \approx \quad & f(x_2, \bar{p}) + D_x f(x_2, \bar{p})A_1\varepsilon && = x_3 + A_2A_1\varepsilon && (3.11b)\\ & \vdots \\ f(x_T + A_{T-1}\cdots A_1\varepsilon, \bar{p}) \approx \quad & x_1 + (A_TA_{T-1}\cdots A_2A_1)\varepsilon && && (3.11c) \end{aligned}$$

The error after one period of the orbit is, by approximation, equal to $(A_T \cdots A_1)\varepsilon$. It is clear that this error is smaller than ε when the magnitude of $(A_T \cdots A_1)$ is smaller than one. In other words, the stability of the period T orbit is determined by the eigenvalues λ of $(A_T \cdots A_1)$, i.e the product of all the T matrices A_i in descending order. The starting point of the multiplication doesn't matter: the eigenvalues of $(A_TA_{T-1} \cdots A_2A_1)$ are exactly equal to those of e.g. $(A_4 \cdots A_1A_TA_{T-1} \cdots A_5)$.

The eigen*vectors* do differ though. Where the (in)stability of an orbit is the same at each iteration, the direction in which this happens is of course different. Now assume that the orbit has s stable and u unstable eigenvalues. Then at every iteration i there are s stable eigenvectors $\{\nu_{i,1}, \ldots, \nu_{i,s}\}$ and u unstable eigenvectors $\{\nu_{i,1}, \ldots, \nu_{i,u}\}$, which are formed by the eigenvectors of $(A_iA_{i-1} \cdots A_{i-T+1})$. The idea of OGY is then, in order to control the u unstable directions, the system is perturbed u times, so that x_{i+u} lands on the linearized stable manifold of the periodic orbit through the point x^*_{i+u}. In other words, after u iterations the deviation should lie on the linearized stable subspace spanned by the stable eigenvectors $\{\nu_{i+u,1}, \ldots, \nu_{i+u,s}\}$:

$$x_{i+u} - x^*_{i+u} = \alpha_1\nu_{i+u,1} + \alpha_2\nu_{i+u,2} + \ldots + \alpha_s\nu_{i+u,s} \tag{3.12}$$

Using linearization (3.8), $x_{i+u} - x^*_{i+u}$ is approximated by

$$\begin{aligned} x_{i+2} - x^*_{i+2} &= A_{i+1}\left(x_{i+1} - x^*_{i+1}\right) + B_{i+1}\left(p_{i+1} - \bar{p}\right) \\ &= A_{i+1}A_i\left(x_i - x^*_i\right) + A_{i+1}B_i\left(p_i - \bar{p}\right) + B_{i+1}(p_{i+1} - \bar{p}) && (3.13a)\\ x_{i+u} - x^*_{i+u} &= A_{i+(u-1)}\cdots A_i\left(x_i - x^*_i\right) + A_{i+(u-1)}\cdots A_{i+1}B_i\left(p_i - \bar{p}\right) \\ &\quad + A_{i+(u-1)}\cdots A_{i+2}B_{i+1}(p_{i+1} - \bar{p}) + \ldots \\ &\quad + A_{i+(u-1)}B_{i+(u-2)}\left(p_{i+(u-2)} - \bar{p}\right) + B_{i+(u-1)}\left(p_{i+(u-1)} - \bar{p}\right) && (3.13b)\\ &= \Phi_{i,0}\left(x_i - x^*_i\right) + \Phi_{i,1}B_i\left(p_i - \bar{p}\right) + \Phi_{i,2}B_{i+1}\left(p_{i+1} - \bar{p}\right) + \ldots \\ &\quad + \Phi_{i,u-1}B_{i+(u-2)}\left(p_{i+(u-2)} - \bar{p}\right) + B_{i+(u-1)}\left(p_{i+(u-1)} - \bar{p}\right), && (3.13c) \end{aligned}$$

Where $\Phi_{i,j}$ is defined as

$$\Phi_{i,j} = A_{i+u-1}A_{i+u-2}\cdots A_{i+j+1}A_{i+j} \tag{3.14}$$

Furthermore, a matrix C_i at each iteration i is defined as

$$C_i = \left[\Phi_{i,1}B_i \,\vdots\, \Phi_{i,2}B_{i+1} \,\vdots\, \cdots \,\vdots\, \Phi_{i,u-1}B_{i+u-2} \,\vdots\, B_{i+u-1} \,\vdots\, \nu_{i+u,1} \,\vdots\, \nu_{i+u,2} \,\vdots\, \cdots \,\vdots\, \nu_{i+u,s}\right] \tag{3.15}$$

This matrix C_i can be compared to the controllability matrix of section 2.2, so the system is controllable if C_i is nonsingular. When the control effort is defined similar to (3.5), i.e.

$$p_i - \bar{p} = -K_i^T(x_i - x^*_i), \tag{3.16}$$

then Romeiras [16] and So and Ott [20] show that Ackermann's method for setting the unstable poles to zero, as described in section 3.2.1, is equivalent to defining K_i^T as

$$K_i^T = \kappa C_i^{-1} \Phi_{i,0}, \tag{3.17}$$

where κ is an n-dimensional row vector whose first entry is one and remaining entries zero. Keep in mind that the restriction of (3.2) is still valid. If $|p_i - \bar{p}| \geq \delta$, then p_i is set to zero.

3.2.3 OGY and delay coordinates

Most of the time it is impossible to measure all state variables of a system. To overcome this problem, section 2.1.3.3 gave a solution by introducing delay coordinates, for which an adjustment to the OGY algorithm is needed, as Dressler and Nitsche show in their article [4]. For a system with discrete time t_i (e.g. by taking the Poincaré map) recall the definition of the delay coordinate:

$$X(t_i) = [x(t_i), x(t_i - \tau), x(t_i - 2\tau), \ldots, x(t_i - (n-1)\tau)]\,. \tag{3.18}$$

Assume that the time between two successive intersections of the solution with the Poincaré map is t_F. It is clear to see that when $(n-1)\tau > t_F$, the delay coordinate X at time t_i contains information of the previous Poincaré intersection at time $t_i - t_F$. So if there was a parameter perturbation at time $t_i - t_F$, it still has influence at time t_i. In fact, all parameter values $\{p_i, \ldots, p_{i-r}\}$ have influence on the variable $X(t_i)$ (to be called X_i), where r is the smallest integer such that $(n-1)\tau < rt_F$. This observation leads to an alternative description of the system:

$$X_{i+1} = F\left(X_i, p_i, p_{i-1}, \ldots, p_{i-r}\right), \tag{3.19}$$

which leads to an alternative linearization:

$$X_{i+1} - X^*_{i+1} = A_i\left(X_i - X_i^*\right) + B_i^1\left(p_i - \bar{p}\right) + B_i^2\left(p_{i-1} - \bar{p}\right) + \ldots + B_i^{r+1}\left(p_{i-r} - \bar{p}\right). \tag{3.20}$$

Here B_i^j is equal to

$$B_i^j = B_{i+T}^j = D_{p_{i-(j-1)}} F\left(X, p_i, p_{i-1}, \ldots, p_{i-r}\right). \tag{3.21}$$

In (3.20) p_i is the only unknown on the right hand side. In order to solve for p_i, new variables are introduced:

$$Y_i = \begin{pmatrix} X_i \\ p_{i-1} \\ p_{i-2} \\ \vdots \\ p_{i-r} \end{pmatrix} \qquad \text{and} \qquad Y_i^* = Y_{i+T}^* = \begin{pmatrix} X_i^* \\ \bar{p} \\ \bar{p} \\ \vdots \\ \bar{p} \end{pmatrix}, \tag{3.22}$$

and the matrices A and B are redefined as

$$\tilde{A}_i = \begin{pmatrix} A_i & B_i^2 & B_i^3 & \cdots & B_i^r & B_i^{r+1} \\ \mathbf{0} & 0 & 0 & \cdots & 0 & 0 \\ \mathbf{0} & 1 & 0 & \cdots & 0 & 0 \\ \mathbf{0} & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \qquad \text{and} \qquad \tilde{B}_i = \begin{pmatrix} B_i^1 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{3.23}$$

With these new variables and matrices, the linearization (3.20) can be written in the form

$$Y_{i+1} - Y^*_{i+1} = \tilde{A}_i \left(Y_i - Y^*_i\right) + \tilde{B}_i \left(p_i - \bar{p}\right). \tag{3.24}$$

The new matrices $\tilde{A}_i$ and $\tilde{B}_i$ can then be used in the OGY algorithm described in sections 3.2.1 and 3.2.2.

3.3 Implementation of the OGY algorithm

The OGY control algorithm described in the previous section can only be applied once the fixed points or periodic orbits and the matrices A and B are known. Sometimes these have to be estimated though. Therefore, in order to control a n-dimensional continuous time system where not all variables can be measured, the following general strategy can be adopted:

1. Define the delay coordinate $X(t) = [x(t), x(t-\tau), \ldots, x(t-(n-1)\tau)]$;
2. Convert the system into a discrete time system, by constructing the Poincaré map, creating $X(t_i) = [x(t_i), x(t_i-\tau), \ldots, x(t_i-(n-1)\tau)]$;
3. Find fixed points $X(t_{i+1}) = X(t_i)$ and period T points $X(t_{i+T}) = X(t_i)$ of the Poincaré map;
4. Linearize the Poincaré map in the vicinity of these points, finding A;
5. Investigate the influence of a parameter perturbation and linearize again around the points of interest, thus finding B;
6. Implement the OGY algorithm described in section 3.2.

Especially steps 3 and 4 may need some extra attention, since the methods used at these steps require a decent amount of statistics. In order to estimate fixed points and orbits, there is a method available called *recurrence method*, and for the estimation of A and B a *Least Squares method* will be used. The theory behind these methods is presented in the following sections; further adjustments will be made in chapter 5.

3.3.1 Recurrence method

In order to apply OGY control, (an estimation of) the fixed points or orbits of the discrete time map of the system are needed, i.e. the values for x_i for which $x_{i+1} = x_i$ or $x_{i+T} = x_i$ holds, need to be found. When an exact description of the discrete time map is present, these points can be found by substituting these conditions into (3.1), keeping $p = \bar{p}$. For higher periodic orbits the calculations can become very complicated though. Sometimes an accurate description of the discrete time map isn't available, thus a different approach is needed.

Lathrop and Kostelich discuss such an approach in their article [11]. Assume a chaotic attractor of a continuous time system (or its delay coordinates reconstruction), and period one, two and three orbits embedded within it (figure 3.1). When a point x is close to an orbit, it will stay in the vicinity of this orbit for a certain time, moving with a frequency approximately equal to that of the orbit. Thus the motion of x is an approximation of the unstable orbit. The Poincaré points x_i and x_{i+T} are thus close to each other and give an approximation of x^*_i.

Now collect a large amount of data points, i.e. successive Poincaré intersections, and define a vicinity $\varepsilon > 0$. For each point x_i follow its images $x_{i+1}, x_{i+2}, \ldots$ until the smallest $k > i$ is found such that $\|x_k - x_i\| < \varepsilon$, and define $m = k - i$. Then if k exists, x_i is an (m, ε) recurrent point.

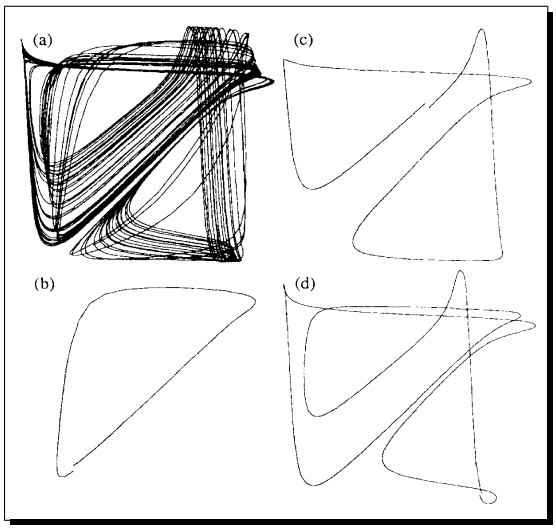


Figure 3.1: The recurrence method; from [11] page 4028

Such a point only indicates a possible period m orbit, but can not guarantee its existence. If recurrent points form a sequence though, so if all points $x_i, x_{i+1}, \ldots, x_k$ are (m, ε) recurrent points, its presence becomes more likely. One can also make a histogram in which the number of occurrences of each value of m are plotted, in order to investigate this likelihood. A smaller ε and more data points will increase the accuracy of the recurrence method. Results for very high periodic orbits are not reliable, since high periodic recurrences are normally due to the ergodic behavior and not because of the stability of the orbit. Furthermore, low periodic points could occur as high periodic ones and vice versa, depending on the size of ε.

In order to find all the m points of a period m orbit, one should collect all likely (m, ε) recurrent points, i.e. only those who are in a sequence (as described before). All points which are in each others vicinity (meaning they represent the same Poincaré point) should then be grouped and averaged. The thus achieved points $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m$ provide a reasonable estimation of the actual orbit $x_1^*, x_2^*, \ldots, x_m^*$.

3.3.2 Least Squares method

As can be concluded from section 3.2.2, the matrices A_i and B_i have to be computed for every point $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m$ found by the recurrence method. Since it is assumed that there is no model present, these matrices have to be estimated with the same data as was used for the recurrence method. The estimation is then done by an ordinary Least Squares method, described in e.g. [6].

The least squares method is normally used to find a best-fit linear curve for $y = f(x)$ using an amount of data points. For $(x_1, y_1), (x_2, y_2), \ldots, (x_p, y_p)$, the basic form of least squares finds a and c in a function $y = ax + c$ for which the sum of squares of the errors $e_j = y_j - (ax_j + c)$ is as small as possible:

$$\min_{a,c} : \quad \sum_{j=1}^{p} e_j^2 = \sum_{j=1}^{p} [y_j - (ax_j + c)]^2 \tag{3.25}$$

For n-dimensional problems this becomes:

$$\min_{a_1 \cdots a_k, c} : \quad \sum_{j=1}^{p} e_j^2 = \sum_{j=1}^{p} [y_j - (a_1 x_{1,j} + a_2 x_{2,j} + \ldots + a_k x_{k,j} + c)]^2 \tag{3.26}$$

Here, the deviation of an n-dimensional point from the orbit is considered, expressed as

$$(x_i - \bar{x}_i, x_{i+1} - \bar{x}_{i+1}) = (\delta x_i, \delta x_{i+1}) \,, \tag{3.27}$$

where δx_i has n components $\delta x_{i,j}$ (with $j = 1, 2, \ldots, n$). Thus a linear curve for $\delta x_{i+1} = f(\delta x_i)$ must be sought. The points used for this are found by defining a small vicinity $\xi > \varepsilon$. For every $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m$ combinations (x_i, x_{i+1}) for which $\|x_i - \bar{x}_i\| < \xi$ and its image $\|x_{i+1} - \bar{x}_{i+1}\| < \xi$ should be found. If ξ is small enough only points close to the orbit, where the linearization is valid, are selected.

3.3.2.1 Finding A

In order to find the matrix A, the linear fit for $\delta x_{i+1} = f(\delta x_i)$ can be written as

$$\begin{aligned} \delta x_{i+1,1} &= a_{1,1}\delta x_{i,1} + a_{1,2}\delta x_{i,2} + \ldots + a_{1,n}\delta x_{i,n} + c_1 \\ \delta x_{i+1,2} &= a_{2,1}\delta x_{i,1} + a_{2,2}\delta x_{i,2} + \ldots + a_{2,n}\delta x_{i,n} + c_2 \\ &\vdots \\ \delta x_{i+1,n} &= a_{n,1}\delta x_{i,1} + a_{n,2}\delta x_{i,2} + \ldots + a_{n,n}\delta x_{i,n} + c_n, \end{aligned} \tag{3.28}$$

creating n equations which can be solved with the least squares method. A is then formed by

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{pmatrix}.$$

The numbers c_i form a vector C which denotes an offset, which is not equal to the matrix B in the OGY algorithm. In fact, since only deviations from the orbit are considered, this offset should be equal to zero in the ideal case where $\bar{x} = x^*$. Therefore C will be neglected.

3.3.2.2 Finding B

The matrix B is found by investigating the influence of the parameter perturbation. First consider the delay coordinates case, so there are multiple matrices B^k with $k = 1, 2, \ldots, r+1$. Recall from section 3.2.3 that r is the smallest integer such that $(n-1)\tau < rt_F$. Then while the system is running, the maximum parameter perturbation $\delta = p - \bar{p}$ is turned on at each $(r+1)$th piercing of the Poincaré map and turned off the next piercing. From the resulting data series only those points are selected which are close to an orbit (similar to section 3.3.2.1). These points are then divided into $(r+1)$ groups: pairs (x_i, x_{i+1}) for which $p_i \neq \bar{p}$, pairs (x_i, x_{i+1}) for which $p_{i-1} \neq \bar{p}$, etc. (until $p_{i-r} \neq \bar{p}$). Now, for e.g. $p_i \neq \bar{p}$, (3.20) reduces to

$$x_{i+1} - \bar{x}_{i+1} = A_i (x_i - \bar{x}_i) + B_i^1 (p_i - \bar{p}).$$

Since A has just been calculated, the only unknown variables in this equation are the components of B_i^1, which of course can now be easily estimated. The matrices $B_i^2, \ldots, B_i^{r+1}$ can be estimated similarly.

When delay coordinates are not used and there is only one matrix B, the perturbation is switched on every odd piercing of the Poincaré map. Then only the pairs (x_i, x_{i+1}) in the vicinity of the orbit for which the perturbation on x_i is switched on are used to estimate B.

It should be noted though, that when there is a correlation between the separate $\delta x_{i+1,j}$ in (3.28), the least squares method will not be an accurate method to find A and B. A so called Partial Least Squares fit [23] should then give better results. However, in this case it is assumed that there is no correlation, so the ordinary least squares can be used.

Chapter 4

OGY control of the Hénon map

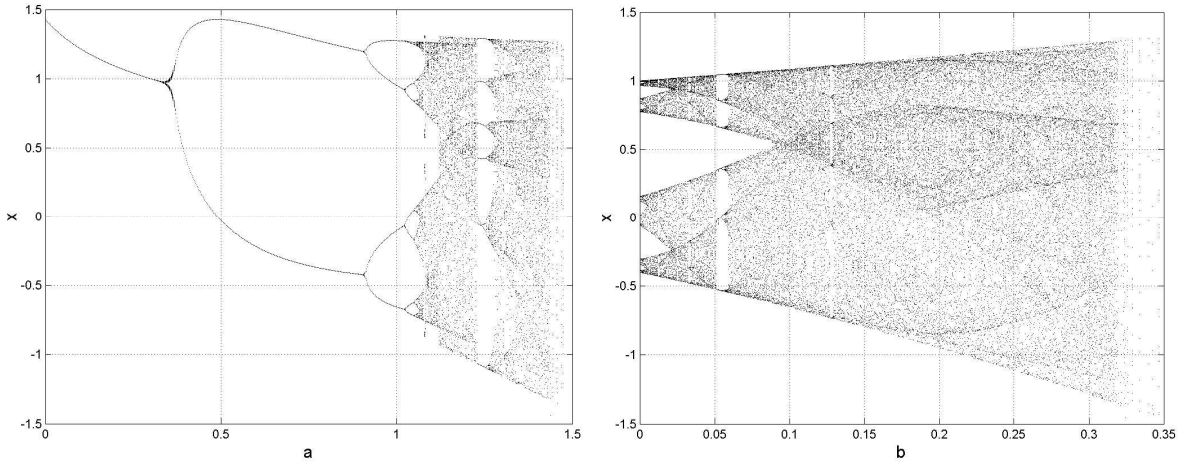
This chapter will numerically implement the OGY control algorithm as described in chapter 3. The system used is the discrete Hénon map.

4.1 Introduction to Hénon

A special kind of chaotic system is the so called Hénon map [8], named after its inventor. The Hénon map is a purely theoretical map, and doesn't represent a physical system. Still, this discrete time system or map describes the basic characteristics of chaotic behavior (see e.g. [21]) and has a very clear strange attractor. Its equation is quite simple and given by:

$$\begin{aligned} x_{i+1} &= f_1(x_i, y_i) = y_i + 1 - ax_i^2 \\ y_{i+1} &= f_2(x_i, y_i) = bx_i. \end{aligned} \tag{4.1}$$

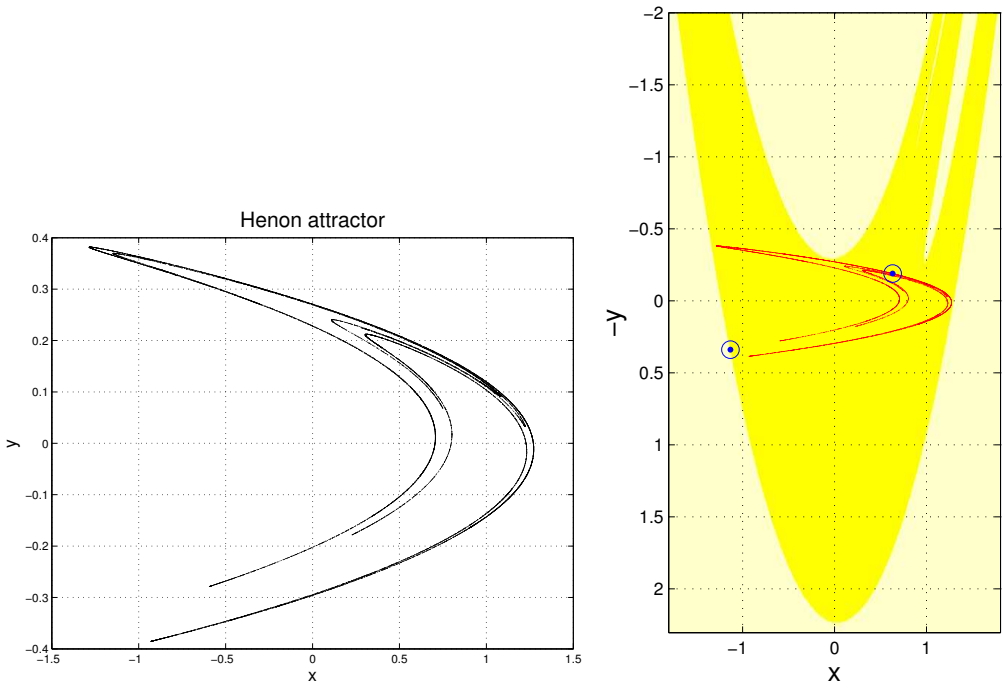
Typical parameter values for which chaos occurs are $a = 1.4$ and $b = 0.3$. This is illustrated by the bifurcation diagrams of the Hénon map, shown in figure 4.1. The strange attractor of



(a) a variable, $b = 0.3$

(b) b variable, $a = 1.4$

Figure 4.1: Bifurcation diagrams of the Hénon map



(a) Graphical representation of the Hénon map (b) Basin of attraction

Figure 4.2: The Hénon map and its basin of attraction

the Hénon map with these parameter values is shown in figure 4.2(a), where y_i is a function of x_i. After the removal of transient behavior, the solution of the map will always wander around on this attractor. Some initial conditions will never converge to the attractor though, but $y_i \to -\infty$. When the long term behavior ($t \to \infty$) of every initial condition is examined, the basin of attraction is obtained, which is shown in figure 4.2(b). All initial conditions inside the darker V-shaped part will converge to the attractor, while for the other points $y_i \to -\infty$.

4.1.1 Fixed points and linearization

The fixed points of (4.1) are easily found by setting $x_{i+1} = x_i = x^*$ and $y_{i+1} = y_i = y^*$:

$$\left.\begin{array}{l} x^* = y^* + 1 - ax^{*2} \\ y^* = bx^* \end{array}\right\} \Rightarrow x^* = bx^* + 1 - ax^{*2} \Rightarrow ax^{*2} + (1-b)x^* - 1 = 0,$$

which has solutions

$$x^* = \frac{-(1-b) \pm \sqrt{(1-b)^2 + 4a}}{2a}. \tag{4.2}$$

For $a = 1.4$ and $b = 0.3$ this gives

$$\begin{pmatrix} x^* \\ y^* \end{pmatrix} = \begin{pmatrix} -1.1314 \\ -0.3394 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} x^* \\ y^* \end{pmatrix} = \begin{pmatrix} 0.6314 \\ 0.1894 \end{pmatrix}, \tag{4.3}$$

which are drawn in figure 4.2(b). In the remaining part of this chapter only the second fixed point will be used, since the first point lies on the edge of the basin of attraction, and is therefore not suitable for OGY control.

Linearizing the Hénon map using (3.4) gives

$$A = \quad D_{x,y}f(x,y,a,b) \quad = \begin{bmatrix} -2ax & 1 \\ b & 0 \end{bmatrix} \tag{4.4a}$$

$$B_a = \quad D_a f(x,y,a,b) \quad = \begin{bmatrix} -x^2 \\ 0 \end{bmatrix} \tag{4.4b}$$

$$B_b = \quad D_b f(x,y,a,b) \quad = \begin{bmatrix} 0 \\ x \end{bmatrix}. \tag{4.4c}$$

4.2 Application of OGY to Hénon

In this section the OGY control will be applied to the Hénon map described in section 4.1. The application will be done in a number of steps with increasing complexity, in order to keep the algorithm understandable for the reader.

4.2.1 Known fixed point, known A and B

Since the Hénon map knows two parameters a and b, both parameters could be used for OGY control. In the first case it is assumed that the fixed point (4.3) and the matrices A and B, (4.4), are known. Furthermore, it can be proven that the system is controllable, since (2.7) becomes

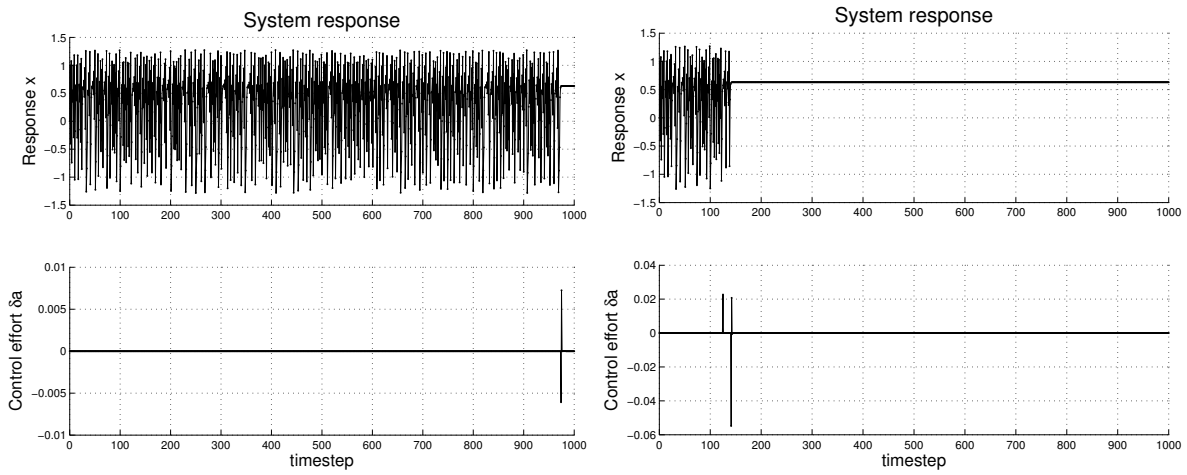
$$P_a = [B_a, AB_a] = \begin{bmatrix} -x^2 & 2ax^3 \\ 0 & -bx^2 \end{bmatrix} \qquad \text{and} \qquad P_b = [B_b, AB_b] = \begin{bmatrix} 0 & x \\ x & 0 \end{bmatrix}, \tag{4.5}$$

which both have full rank if $x \neq 0$. Now the OGY algorithm is quite straightforward: the matrix K^T can easily be calculated using Ackermann's method. The success of OGY only depends on the choice of the maximum parameter perturbation (3.2), which will be called δa_{max} or δb_{max} here. The corresponding Matlab® code is given in appendix C.4.1.

Figures 4.3 and 4.4 show the response of the Hénon map for different choices of the accessible parameter. Note the difference for different values of δa_{max} and δb_{max}. When the maximum parameter perturbation is very small, the OGY algorithm waits until the state of the system is very close to the fixed point. The linearization is quite accurate there, so in general the first attempt will immediately result in a steady state. It should be noted though that a small δa or δb doesn't have to imply that the state of the system is close to the fixed point. Recall (3.5):

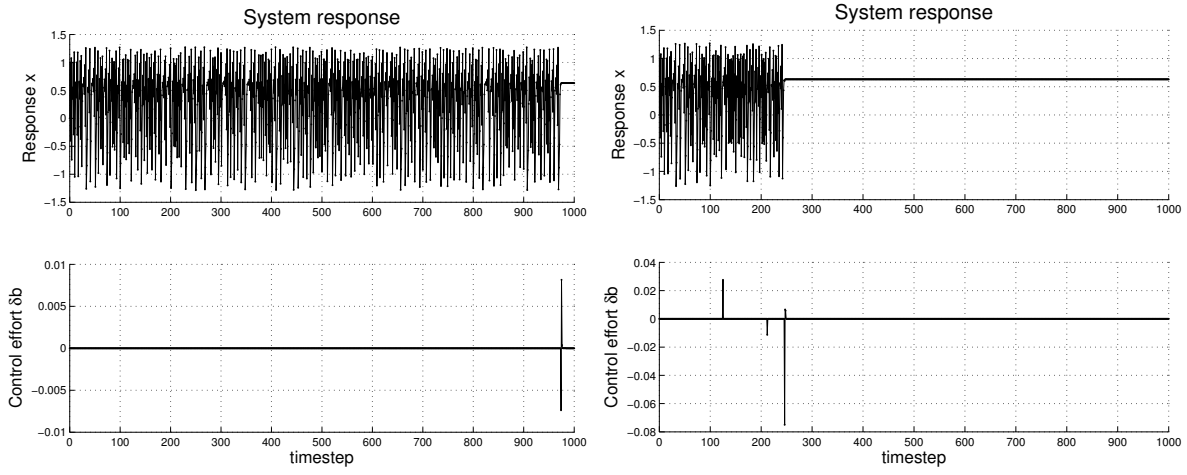
$$p - \bar{p} = -K^T(x_i - x^*).$$

Since K and $x_i - x^*$ are vectors, there is a $x_i - x^* \neq \mathbf{0}$ such that $p - \bar{p} = 0$. So (3.5) defines a *band* of points for which the control effort is nonzero, instead of an (elliptic) vicinity around x^*. This means that there can be a (false) control effort when the state is far away from the fixed point. In this case a control effort δa or δb will not result in a steady state. Furthermore, due to the behavior of the system, it is possible that the system drifts away from the fixed point at iteration $i+1$ while it was close to it at iteration i, thus resulting in a false control effort.



(a) OGY control with $\delta a_{max} = 0.008$ (b) and with $\delta a_{max} = 0.1$

Figure 4.3: Controlling Hénon with parameter a; initial condition (0,0)



(a) OGY control with $\delta b_{max} = 0.01$ (b) and with $\delta b_{max} = 0.1$

Figure 4.4: Controlling Hénon with parameter b; initial condition (0,0)

This phenomenon occurs more often when δa_{max} and δb_{max} are relatively large. The algorithm will then try to control points which are further away from the fixed point, where the linearization is not always accurate. This results in much more control attempts, including more false ones. *In general*, higher δa_{max} and δb_{max}, for as far this is practically possible, will result in a smaller time to achieve control, but this can highly depend on the ergodic behavior. Too large perturbations can make the system instable, since it can push the system outside the basin of attraction (figure 4.2(b)).

Finally, note that once a steady state is established, the control effort decreases to zero.

This is because the fixed point and A and B are known, so that the system is controlled exactly towards the actual fixed point. All remaining sections will show different results.

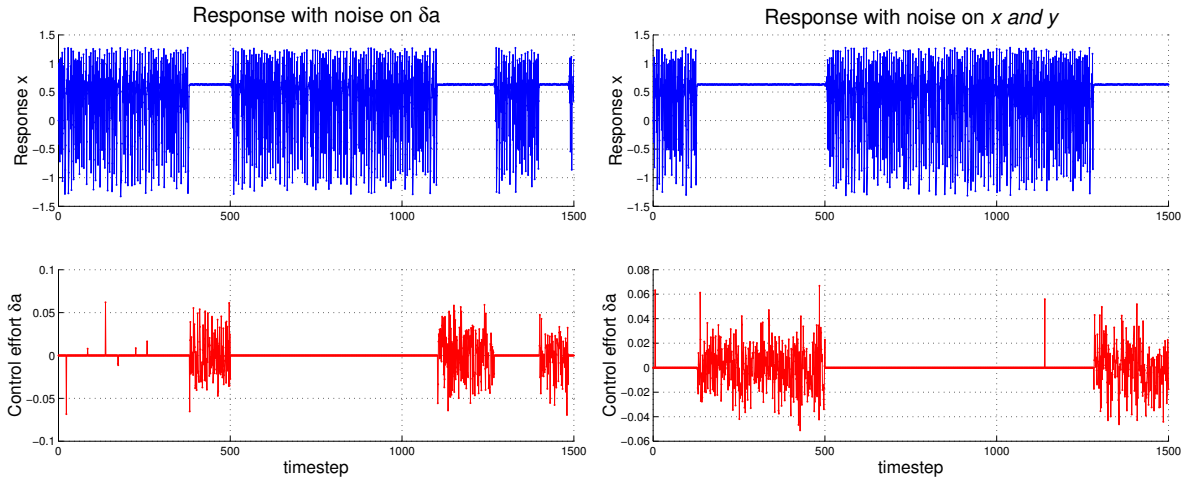
4.2.1.1 Influence of noise

In the remainder of this chapter a will be used as the control parameter. This section will shortly discuss the influence of noise on the OGY algorithm. Two cases are considered: noise on the state (x, y) and on δa. In both cases the maximum parameter perturbation δa_{max} is 5% of $\bar{a}$, and the control is switched on from iteration 0 till 500 and 1000 till 1500.

Noise on a: Results in figure 4.5(a)

In this case δa is calculated as in section 4.2.1, but before this perturbation is applied, a random number from a normal distribution is added. The variance is chosen relatively high: 20% of δa_{max}. When $\delta a = 0$ no noise is added. Two things catch the eye:

- In the steady state the control effort $\delta a \neq 0$. Due to the noise, the state of the system will never be exactly equal to the fixed point, thereby still needing control.
- Around iteration 1300 the noise forces the system outside the controlled region, into chaos again. When the noise is large enough, the control effort needed to overcome the noise could exceed δa_{max}, so that δa is set to zero, causing the system to go to chaos. Some iterations later the system is again controlled.



(a) Noise variance: 20% of δa_{max}

(b) Noise variance: 5% of δa_{max}

Figure 4.5: Influence of noise on OGY; control parameter a, initial condition (0,0)

Noise on (x, y): Results in figure 4.5(b)

In this case an amount of noise is added to x and y at every iteration, so the noise has a direct effect on the state of the system. Although the results are similar to the previous case, the most important difference is the size of the noise: variances larger than approx. 5% of δa_{max} will result in instability. The added noise could easily cause the system to land outside the basin of attraction (figure 4.2(b)), so that $y_i \to -\infty$.

4.2.2 Unknown fixed points, known A and B

In the next step it is assumed that the fixed point is unknown and it is desirable to also control higher periodic orbits. The matrices A and B are still known and given in (4.4).

To estimate fixed and periodic points, the recurrence method of section 3.3.1 will be used. The written Matlab® code can be found in appendix C.3.1. The accuracy of the recurrence method is determined by size of the vicinity ε and the length of the data series. Here the vicinity is scaled in x- and y-direction, giving it the shape of an ellipse:

$$\frac{x^2}{range_x^2} + \frac{y^2}{range_y^2} = \varepsilon^2, \tag{4.6}$$

where $range_x$ and $range_y$ denote the difference between the maximum and minimum values of x and y. For 100,000 data points a choice of $\varepsilon = 0.002$ gives reasonable results (i.e. between 20 and 50 recurrence points for lower periodic orbits), as shown in figure 4.6. Higher periodic orbits than 15 are not considered because most of the higher periodic recurrences are coincidental and therefore not reliable.

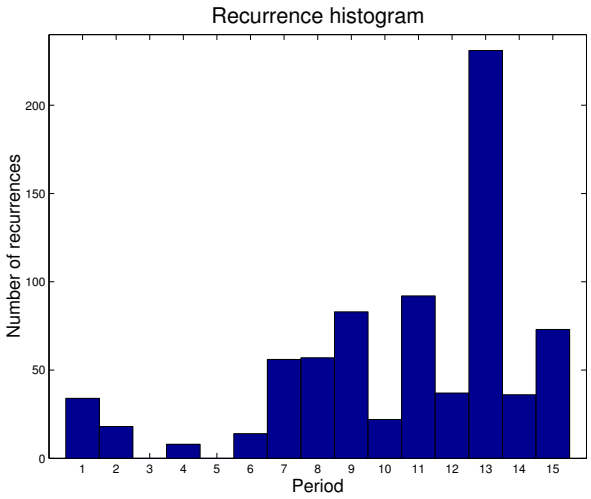
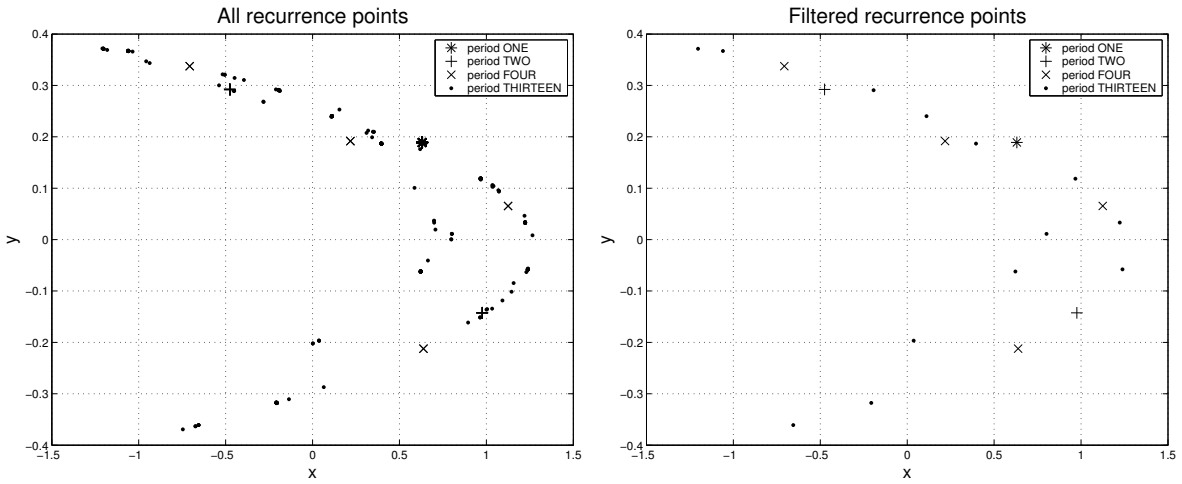


Figure 4.6: Recurrence histogram of 100,000 Hénon points, $\varepsilon = 0.002$

Two things should be noted regarding figure 4.6. First, there seems to be no recurrences of period 3 and 5, and second, period 13 seems to be highly dominant, i.e. it has by far the highest number of recurrences. Therefore it is chosen to apply OGY control to the lower periodic orbits 1, 2 and 4 and the higher periodic orbit 13.

As was discussed in section 3.3.1, not all recurrence points have to be part of the periodic orbit. A filter is needed to select only the relevant points. This filtering is illustrated in figure 4.7. In figure 4.7(a) all points selected by the recurrence method are plotted. Then a filter is applied for every orbit in a number of steps:

1. Search for the most likely orbit. This means searching the longest sequence of (m, ε)-recurrent points. If $(x_i, y_i), (x_{i+1}, y_{i+1}), \ldots, (x_{i+p}, y_{i+p})$ are all recurrent points and p is as large as possible, it is very likely that these points are close to an actual m-periodic orbit.



(a) All points found by the recurrence method

(b) Estimated fixed and periodic points

Figure 4.7: Illustration of the filtering of recurrence points

2. If $p < m$, finish the sequence to length m with original data points. Recall that if (x_i, y_i) is a (m, ε)-recurrent point, it is close to (x_{i+m}, y_{i+m}), even when (x_{i+m}, y_{i+m}) is not a recurrent point itself.

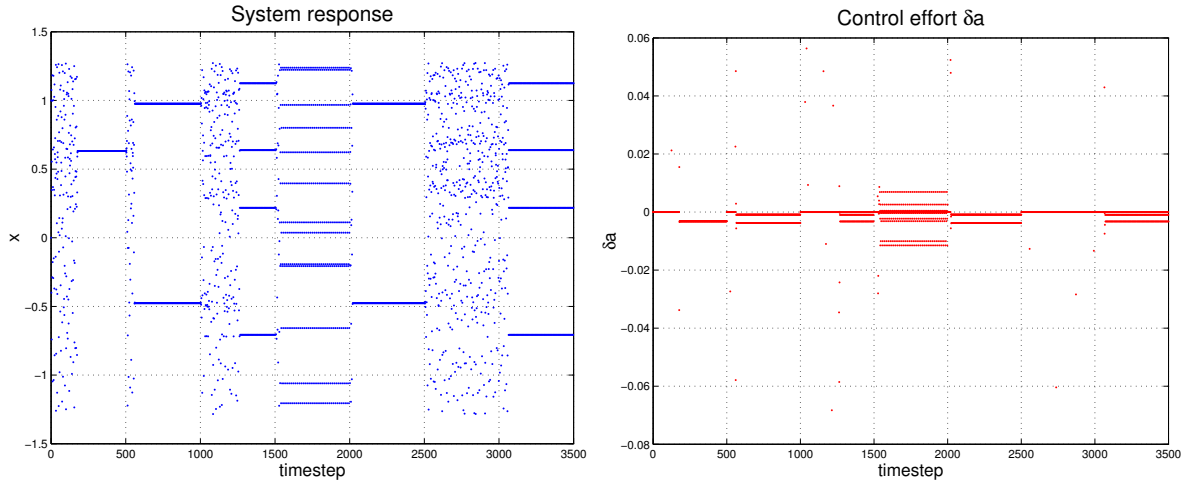
3. Gather all the other selected recurrent points to one of the points $(x_i, y_i), \ldots, (x_{i+m}, y_{i+m})$ when they are in the vicinity defined by (4.6). If they are not inside any vicinity, they're considered coincidental and thrown away.

4. Average all points in each others vicinity, obtaining m estimated values of the m periodic points: $(\bar{x}_i, \bar{y}_i), \ldots, (\bar{x}_{i+m}, \bar{y}_{i+m})$.

Figure 4.7(b) shows the selected points after the described filtering is applied. The number of points thrown away for every period in this filtering proces are 0, 4, 0 and 62. The coordinates of these points are found in appendix A.

The obtained results are then used for the simulation of OGY control, shown in figure 4.8. Every 500 iterations a different orbit is controlled, in the following order: 1, 2, 4, 13, 2, 1 and 4. The corresponding Matlab® code can be found in appendix C.4.2.

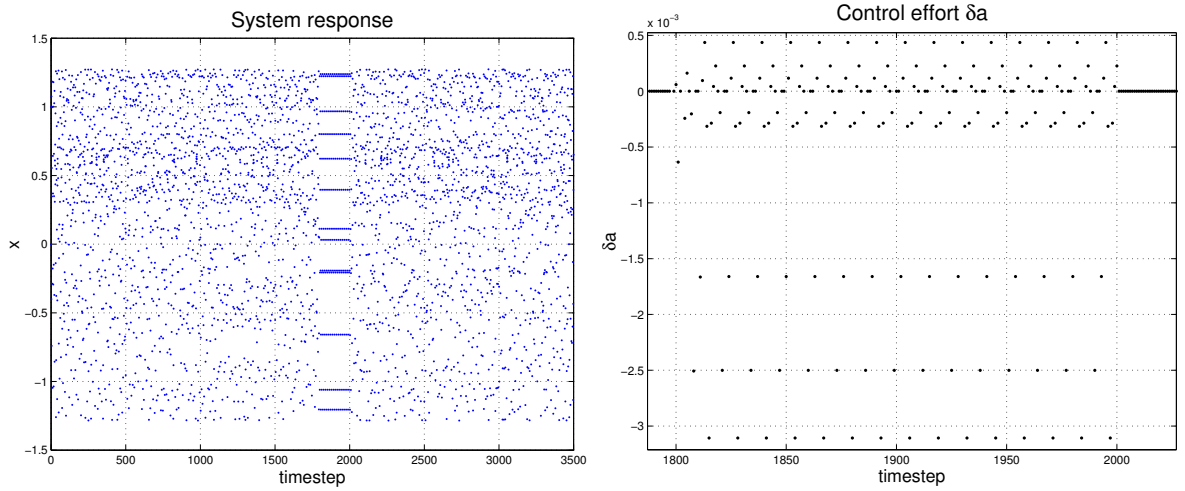
Figure 4.8 clearly shows the success of OGY for higher periodic orbits. Some remarks should be made though. First, in every controlled situation, notice that $\delta a \neq 0$, in other words, there is a continuous perturbation ($\delta a \approx -0.0032$ for the period 1 orbit). This is due to the fact that the orbit estimations are not exactly equal to the actual orbits. After each iteration the system drifts away from the estimated orbit due to its instability, needing a control effort to compensate this. In fact, the system is pushed away from the actual orbit (which needs $p = \bar{p}$), towards the slightly different estimated orbit (which needs $p \neq \bar{p}$).
Second, higher periodic orbits are in general controlled faster than lower ones. This is caused by the simple fact that there are more points to control around. For the period 13 orbit this means that as soon as the system is close to one of the 13 points, control can be applied,



(a) State of the system x

(b) Control effort

Figure 4.8: OGY control for higher periodic orbits; $\delta a_{max} = 0.05a$



(a) State of the system x

(b) Control efforts for period 13 orbit

Figure 4.9: OGY control for higher periodic orbits; $\delta a_{max} = 0.0025a$

instead of e.g. just one point. As a result of this, higher periodic orbits also need smaller control efforts, see figure 4.9, where the maximum perturbation is 20 times smaller, and the period 13 orbit is the only controlled orbit. Notice in figure 4.9(b) that three points which initially need a large δa to be controlled, are now *not* controlled, i.e. $\delta a = 0$. The system is not perturbed at these points, but it can still be controlled due to the fact that the system does not drift away very fast. The next iteration point can still correct the deviation from the orbit with a δa which is smaller than δa_{max}.

Third, notice that between iterations 2500 and 3000 the algorithm fails to control the fixed point. This also results from the previous remark; the ergodic behavior of the system causes the system not to come close to the *single* fixed point within the relatively short time of 500 iterations.
Finally, the steady state values of x and y are not equal to the estimates, nor to the real periodic points. E.g. figure 4.9 shows that the steady state value for the fixed point is:

$$\begin{pmatrix} x \\ y \end{pmatrix} \approx \begin{pmatrix} 0.6319 \\ 0.1896 \end{pmatrix}, \tag{4.7}$$

whereas the real fixed point (4.3) and its estimation are

$$\begin{pmatrix} x^* \\ y^* \end{pmatrix} \approx \begin{pmatrix} 0.6314 \\ 0.1894 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} \approx \begin{pmatrix} 0.6309 \\ 0.1891 \end{pmatrix}.$$

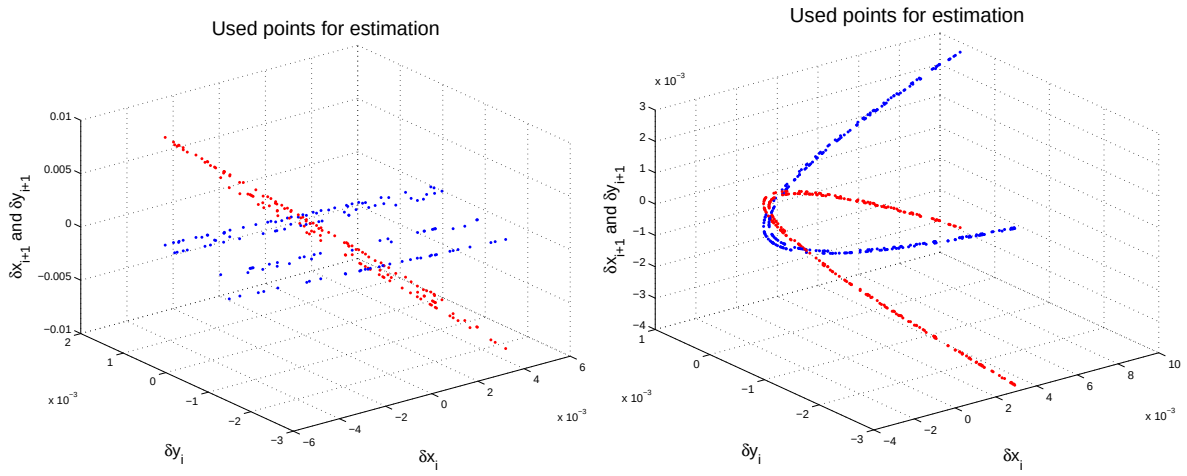
Apparently, since $\delta a \neq 0$ in steady state, there is a steady state error in x and y. In fact, (4.7) is the real fixed point of the Hénon map with $a = \bar{a} + \delta a = 1.4 - 0.0032$. So for every orbit, whenever there is a steady state control effort, there is a steady state setpoint error.

4.2.3 Unknown fixed points, unknown A and B

In this final step the matrices A and B are assumed to be unknown and will be estimated, using the least squares method of section 3.3.2. For the Hénon map, (3.28) reduces to

$$\begin{aligned} \delta x_{i+1} &= a_{1,1}\delta x_i + a_{1,2}\delta y_i + c_1 \\ \delta y_{i+1} &= a_{2,1}\delta x_i + a_{2,2}\delta y_i + c_2 \end{aligned} \tag{4.8}$$

where $\delta x_i = x_i - \bar{x}$ and $\delta y_i = y_i - \bar{y}$, and $\bar{x}$ and $\bar{y}$ are the estimations from the previous section. This linearization will use the same data series as was used for the recurrence method. As was



(a) Points used for fixed point

(b) Points used for point 11 of period 13 orbit

Figure 4.10: Representation of points used for the least squares method

described in section 3.3.2, only points in this series close to the orbit, i.e. within a vicinity ξ, can be used: so $\|x_i - \bar{x}_i\| < \xi$ *and* $\|x_{i+1} - \bar{x}_{i+1}\| < \xi$. In order to collect enough points ξ can be different at every point of an orbit, in general varying between 2ε and 6ε. Lathrop and Kostelich [11] state that ξ should be chosen such that there are approx. 50 points selected, but a graphical representation could provide more information, see figure 4.10. In each of these figures there are two data series, each representing a line in (4.8). Keep in mind that (4.8) is nothing more than the estimation of a 2-D plane in 3-D space. So ξ should be chosen such that the selected points form a well defined flat 2-D plane. Note that although the points in figure 4.10(b) form a *parabola*, they still define a 2-D flat *plane* very well. Therefore these points can all be used to estimate A. For the best results, similar plots should be made for every point in order to find a correct values of ξ.

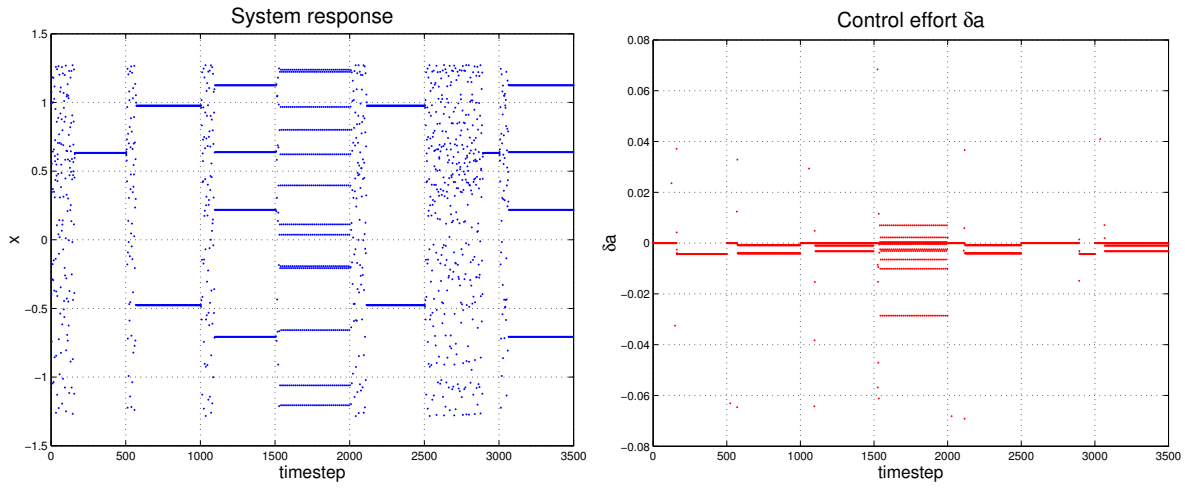
Using this method quite accurate results for A are obtained (see appendix A). Besides A also C is calculated, which is in all cases close to zero and is therefore neglected.

As was discussed in section 3.3.2.2, for the calculation of B a data series is needed where the parameter a is perturbed. This data series is generated by perturbing the system every two iterations with $\delta a = \delta a_{max}$. For Hénon this can't be chosen larger than 0.02 though, for else the system can be perturbed outside the basin of attraction, so that $y_i \to -\infty$.

The points which will be used for estimating B are formed by the pairs $(x_i, y_i), (x_{i+1}, y_{i+1})$ in the vicinity ξ of the orbit, where the perturbation is switched on at i. The value ξ can be different for each point, but should be chosen equal to the ξ with which A was found. B can then be calculated with

$$\begin{bmatrix} \delta x_{i+1} \\ \delta y_{i+1} \end{bmatrix} = A_i \begin{bmatrix} \delta x_i \\ \delta y_i \end{bmatrix} + B_i \delta a \quad \Rightarrow \quad B_i = \left(\begin{bmatrix} \delta x_{i+1} \\ \delta y_{i+1} \end{bmatrix} - A_i \begin{bmatrix} \delta x_i \\ \delta y_i \end{bmatrix} \right) \frac{1}{\delta a}. \tag{4.9}$$

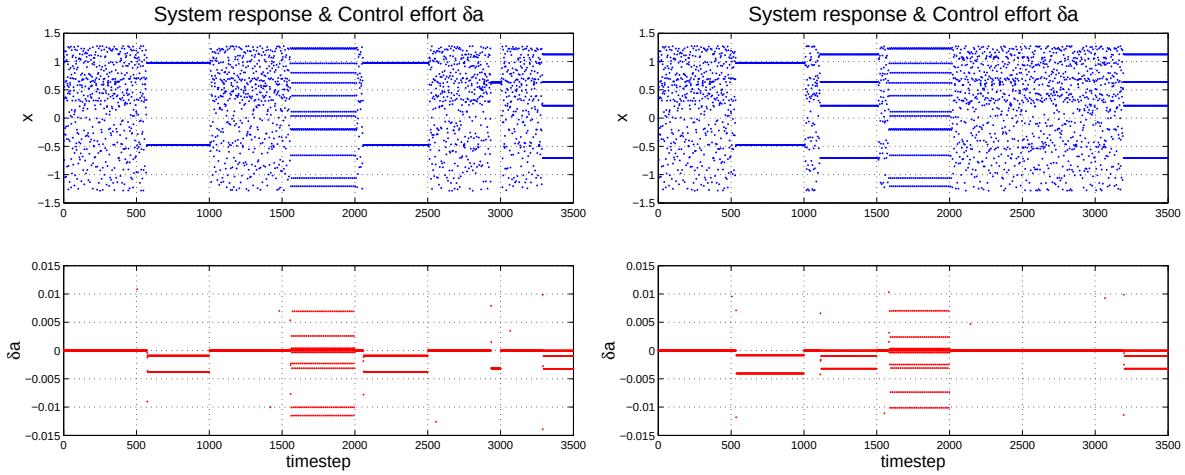
Some values for B are given in appendix A. Notice that at some points the estimates are relatively bad; the error can easily be more than 10%.



(a) State of the system x

(b) Control effort

Figure 4.11: OGY control with estimated orbits, A and B



(a) A and B are known

(b) A and B are estimated

Figure 4.12: Results of OGY control for $\delta a_{max} = 0.01a$

The values of A and B are now used to apply OGY control again. The same orbits are controlled at the same iterations as in figure 4.8, and again $\delta a_{max} = 0.05a$. The results are shown in figure 4.11 and look very similar to the previous case. In principle every orbit can be controlled, approx. with the same time to achieve control as before. The constant control efforts are also nearly equal. Apparently the OGY control is robust enough to overcome some bad estimates of A and B. Even when the maximum perturbation is lowered towards $\delta a_{max} = 0.01a$ there is hardly any difference noticeable, see figure 4.12. The only remark is that the estimated case seems to have some problems controlling lower periodic orbits. Now every orbit has some bad estimates of A and B, but especially the higher periodic orbits 4 and 13 also have some very accurate estimates. Similar to the explanation at the end of section 4.2.2, these accurate estimates are apparently able to force the system towards a steady state.

4.3 Concluding remarks

In this chapter is has been shown how the OGY algorithm can be successfully applied to the Hénon map, even when the periodic orbits and the matrices A and B have to be estimated. Estimation of A and B seemed to have little effect, in contrast to estimating the periodic orbits and fixed points. The latter case introduced constant parameter perturbations and steady state setpoint errors which in general cause the time to achieve control to increase slightly. This time can be decreased considerably using targeting techniques, see [19] and [9].

The time to achieve control is further influenced by the maximum allowed parameter perturbation, here δa_{max}. A higher δa_{max} will in general result, besides more false control attempts, into a smaller time to achieve control. Note though, that δa_{max} cannot be chosen arbitrarily large; there may be system restrictions on the value of the parameter and large perturbations may lead to instability. A small δa_{max} can cause the OGY control to fail. When

for period 1 orbits the maximum perturbation is chosen smaller than the steady state control effort, a controlled state will of course never be reached. But even when δa_{max} is slightly larger than the steady state effort, control is not guaranteed: the solution must then come very close to the periodic orbit, which can take forever. This is somewhat different for higher periodic orbits: e.g. the period 13 orbit showed that the other points on the orbit can still cause control while one or more points on the orbit are not controlled.

OGY is quite robust, which was shown first by adding noise to the system and second by using poor estimates. Noise during the estimation can introduce some major problems though. This noise can cause the estimation of fixed points and orbits to be worse than before and also A and B will generally deviate more than 10%. This error is partly due to the assumption that C can be neglected, but appendix A shows that C can be relatively large. The value of C is then attributed to B, which makes the estimate worse. This can cause OGY to fail controlling the system. One way to cope with this is to adjust the estimating algorithm, which will be done in chapter 5.

Chapter 5

OGY control on the Duffing oscillator

In this chapter the OGY control algorithm will be applied to a Duffing oscillator (2.3). This case will be treated as an extension of chapter 4 and the final results of that chapter will be the starting point of this one, extending the application of OGY to continuous time systems using Poincaré maps and delay coordinates. Furthermore an adjustment will be made to the estimations of fixed points and the matrices A and B. Finally, an adjustment to the delay coordinate algorithm will be made.

5.1 System description

In this chapter the following specific oscillator will be used (see [22]):

$$\ddot{x} + 2\xi\dot{x} + \frac{1}{2}x = F\Omega^2 \sin \Omega t, \tag{5.1}$$

where the right hand side is the acting periodic force. The nominal values of the parameters are $\xi = 0.04$, $\Omega = 0.84$ and $F = 0.188$. The chaotic attractor of this system is given in figure 5.1(a). Any initial condition within the area marked by this attractor will converge towards it, as lots of initial conditions outside this area will do. Some initial conditions outside the area will converge to a periodic orbit given in figure 5.1(b). During the application of OGY control the state of the system will stay within the area, so only the chaotic attractor is of interest.

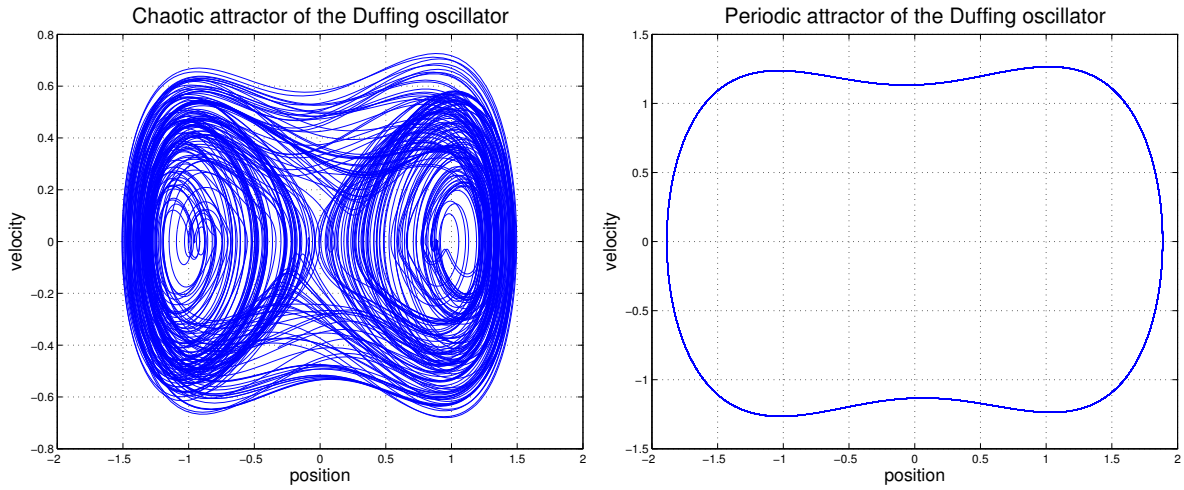
The system can be discretized by making the Poincaré map. Since (5.1) is not autonomous but forced with a certain frequency, this can be done by taking a Poincaré point every period of the acting force. In other words, the system is sampled every $t_F = \frac{2\pi}{\Omega}$ seconds, so that $\sin \Omega t$ remains constant. When position and velocity at these times are plotted, the Poincaré map of figure 5.2(a) is obtained. In some cases it is not possible to measure the velocity $\dot{x}$, but only the position x. In that case delay coordinates can be introduced, using a delay time τ. The delay coordinate is here defined as

$$Z(t) = [x(t), x(t-\tau), x(t-2\tau)]\,.$$

Taking the Poincaré section by sampling every t_F seconds, the discrete 2-D vector

$$z(t_i) = [x(t_i), x(t_i - \tau)] \tag{5.2}$$

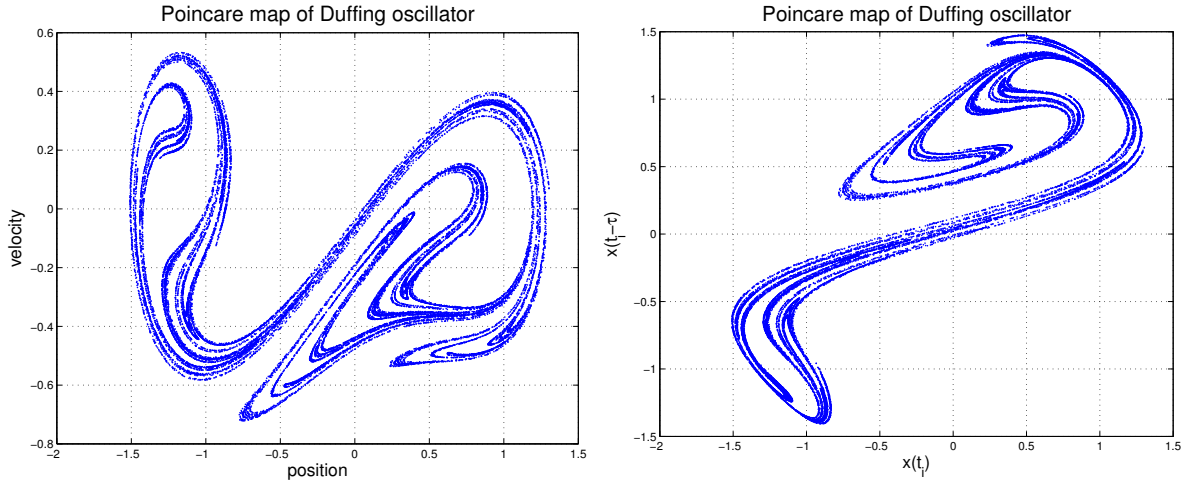
remains. The delay τ can be any positive value, but here it is defined as a quarter of the sampling time, so $\tau = \frac{t_F}{4} = \frac{2\pi}{4\Omega}$. With this the Poincaré map of figure 5.2(b) can be obtained.



(a) Chaotic attractor, initial condition (0,0)

(b) Periodic attractor, initial condition (0,-2)

Figure 5.1: Two different attractors of the Duffing oscillator



(a) Without delay coordinates

(b) With delay coordinates

Figure 5.2: Poincaré maps of the Duffing oscillator

5.2 Application of OGY to Duffing

In this section the OGY algorithm will be implemented in the Duffing oscillator (5.1). In order to do this, the amplitude of the forcing F will be chosen as the accessible parameter. The implementation will be done in a number of steps, each step with increasing complexity. All the fixed points, orbits and matrices are assumed to be unknown.

5.2.1 Adjustment of the estimates

The previous chapter concluded with the remark that a non-zero value of the offset C can cause problems when OGY is implemented. These non-zero values for C are a result of poor estimates of fixed points, which have a large impact on the success of OGY. Especially in the current case, where the discrete time Poincaré map is less known than the Hénon map, accurate estimates are important. This gives reason to redefine the estimation algorithm used in chapter 4.

Suppose that $x_i^* = \bar{x}_i$ and $x_{i+1}^* = \bar{x}_{i+1}$ in (3.28). When the system follows a periodic orbit, then $\delta x_i = \delta x_{i+1} = 0$, and (3.28) returns the zero vector for C. When $x_i^* \neq \bar{x}_i$ and the system is still on the same orbit, $\delta x_i \neq 0$ and $\delta x_{i+1} \neq 0$, so C will be non-zero. Therefore the goal of the following adjustment is to set the offset C to zero, since this implies that $x_i^* = \bar{x}_i$ and $x_{i+1}^* = \bar{x}_{i+1}$ and the estimation is correct. For a 2-D Poincaré map, define a point at a certain iteration k as $[x_k, y_k]^T$, and the actual fixed point as $[x^*, y^*]^T$. One can find a first estimation $[\bar{x}, \bar{y}]^T$ of this fixed point with the recurrence and filtering steps described in section 4.2.2. Then n points $[x_i, y_i]^T$ within a vicinity ξ of $[\bar{x}, \bar{y}]^T$ are found and $[\delta x_i, \delta y_i]^T = [x_i - \bar{x}, y_i - \bar{y}]^T$ is defined. With this the linearization around the estimated fixed point is given by

$$\begin{array}{ll} \delta x_{i+1} & = a_{11}\delta x_i + a_{12}\delta y_i + c_1 \\ \delta y_{i+1} & = a_{21}\delta x_i + a_{22}\delta y_i + c_2 \end{array}, \tag{5.3}$$

in which $a_{i,j}$ and c_j can be calculated using the least squares method, creating A and C. Now if $[\bar{x}, \bar{y}]^T = [x^*, y^*]^T$ then C is zero. Instead, $[\bar{x} + dx, \bar{y} + dy]^T = [x^*, y^*]^T$, so when $[\delta x_i, \delta y_i]^T$ is replaced by $[\delta x_i - dx, \delta y_i - dy]^T$, the vector C should be zero. The vector $[dx, dy]$ can then be estimated by rewriting (5.3) into:

$$\begin{array}{l} (\delta x_{i+1} - dx) = a_{11}\,(\delta x_i - dx) + a_{12}\,(\delta y_i - dy) \\ (\delta y_{i+1} - dy) = a_{21}\,(\delta x_i - dx) + a_{22}\,(\delta y_i - dy) \\ \qquad \Rightarrow \quad \delta x_{i+1} - a_{11}\delta x_i - a_{12}\delta y_i = (1 - a_{11})\,dx - a_{12}dy \\ \qquad\qquad \delta y_{i+1} - a_{21}\delta x_i - a_{22}\delta y_i = -a_{21}dx + (1 - a_{22})\,dy \\ \qquad \Rightarrow \begin{bmatrix} \delta x_{i+1} - a_{11}\delta x_i - a_{12}\delta y_i \\ \delta y_{i+1} - a_{21}\delta x_i - a_{22}\delta y_i \end{bmatrix} = \begin{bmatrix} (1 - a_{11}) & -a_{12} \\ -a_{21} & (1 - a_{22}) \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix}, \end{array} \tag{5.4}$$

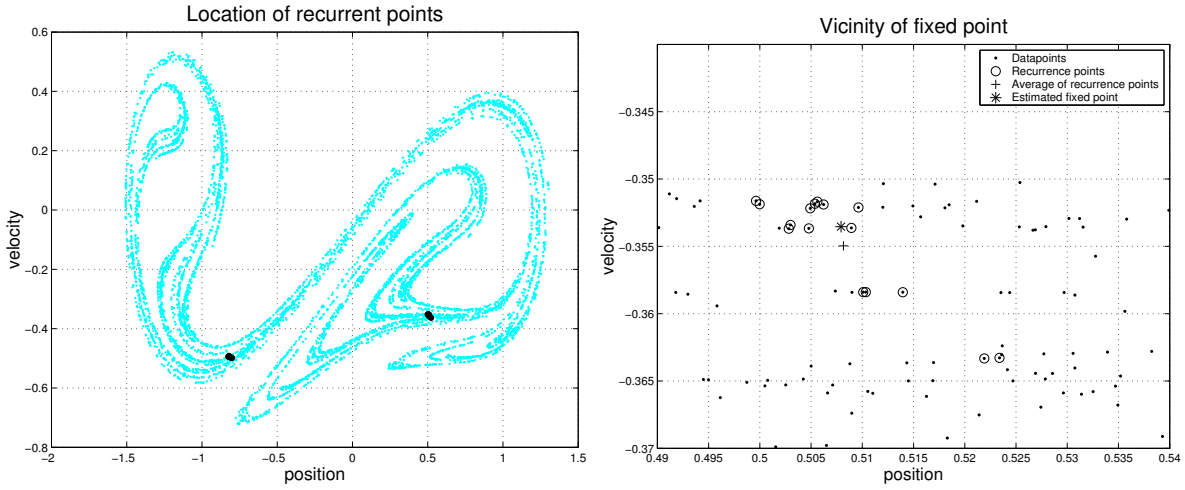
using the same n data points as for (5.3). Now (5.4) has the form $b = Ax$, which can easily be solved using the least squares method. The new estimation for the fixed point is now given by

$$\begin{bmatrix} \bar{x}_{new} \\ \bar{y}_{new} \end{bmatrix} = \begin{bmatrix} \bar{x}_{old} \\ \bar{y}_{old} \end{bmatrix} + \begin{bmatrix} dx \\ dy \end{bmatrix}.$$

This point can now be used to find new points $[x_i, y_i]^T$ within its vicinity ξ, to find new values of A and C. The algorithm can then be followed again, until C is small enough, or $[dx, dy]^T$ is below some small value (here: dx or dy smaller than 10^{-4}). The matrix B is calculated as was done before, with the new fixed point and new matrix A. The derivation for higher periodic orbits is analogous to the above. Realize that for period 2 (5.4) transforms into:

$$\begin{array}{l} \delta x_{i+1,1} - a_{11,1}\delta x_{i,1} - a_{12,1}\delta y_{i,1} = -a_{11,1}dx_1 - a_{12,1}dy_1 + dx_2 \\ \delta y_{i+1,1} - a_{21,1}\delta x_{i,1} - a_{22,1}\delta y_{i,1} = -a_{21,1}dx_1 - a_{22,1}dy_1 + dy_2 \\ \delta x_{i+1,2} - a_{11,2}\delta x_{i,2} - a_{12,2}\delta y_{i,2} = dx_1 - a_{11,2}dx_2 - a_{12,2}dy_2 \\ \delta y_{i+1,2} - a_{21,2}\delta x_{i,2} - a_{22,2}\delta y_{i,2} = dy_1 - a_{21,2}dx_2 - a_{22,2}dy_2. \end{array} \tag{5.5}$$

The Matlab® code for the described iterative algorithm is given in appendix C.3.4. Some results used in the following sections are given in appendix B.



(a) Location of recurrence points

(b) Location of estimated fixed point

Figure 5.3: Estimation of fixed point

5.2.2 Control without delay coordinates

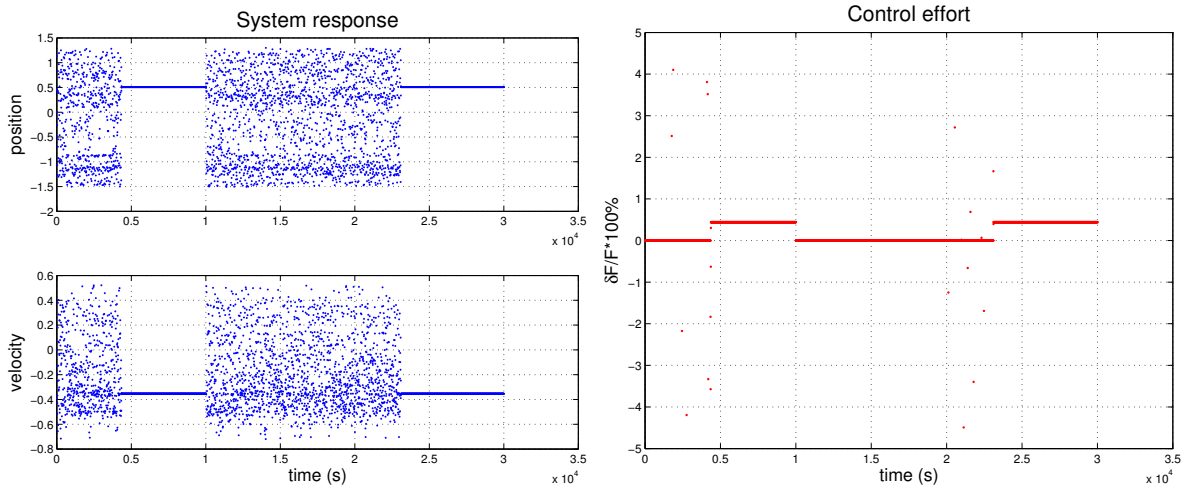
In this section it is assumed that both position and velocity can be measured, so no delay coordinates are needed. The Poincaré map of figure 5.2(a) will thus be used. In this example only one fixed point will be stabilized. The rest of this chapter will only focus on orbits of period 1 or 2, since the identification of higher periodic orbits is rather inaccurate.

First the fixed points of the Poincaré map have to be identified, by using the recurrence method. With $\varepsilon = 0.01$ defined as in (4.6), and ξ chosen twice as large as ε, this method finds 26 $(1, \varepsilon)$ recurrent points in a data series of approx. 26,600 points. These points are divided in two regions, as is shown by figure 5.3(a). For simplicity, this section will concentrate only on the right point, for which figure 5.3(b) shows a close up. This figure shows all recurrence points, together with the initial estimate of the fixed point and the fixed point after the improvement of section 5.2.1. The adjustment thus finds values for the fixed point and A, with which B can be found as in section 3.3.2.2. The perturbation used to generate the data series with which B can be found, is set to $\delta F_{max} = 0.05F = 0.0094$. The absolute values of fixed point, A and B can be found in appendix B.

These results can then be used to find the value of the control matrix K^T, so that OGY can be applied. Figure 5.4 shows the results when $\delta F_{max} = 0.05F$. Control is applied in the time intervals [0, 10000] and [20000, 30000], between those two intervals there is no control at all. Notice the extremely small constant control effort in the controlled situation (about 0.4% of F). This is probably due to a relatively accurate estimation of the fixed point. Still, there is also a steady state setpoint error, since the achieved value differs from the estimation:

$$\begin{bmatrix} x_{achieved} \\ \dot{x}_{achieved} \end{bmatrix} = \begin{bmatrix} 0.5073 \\ -0.3540 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} x_{estimated} \\ \dot{x}_{estimated} \end{bmatrix} = \begin{bmatrix} 0.5079 \\ -0.3535 \end{bmatrix} \tag{5.6}$$

Second, notice the large amount of time needed before the fixed point is reached; it takes about 4370 seconds, which is more than an hour! To understand this, remember the forcing



(a) Position and velocity as a function of time

(b) Control effort as a percentage of δF_{max}

Figure 5.4: OGY control of a fixed point of the Duffing oscillator; $\delta F_{max} = 0.05F$

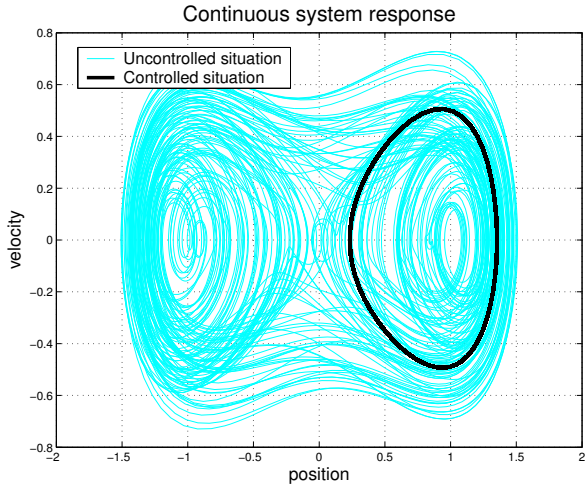


Figure 5.5: Representation of controlled fixed point in continuous time domain

period of the system: $\frac{2\pi}{\Omega} \approx 7.48s$, so there is 7,48 seconds between each Poincaré intersection. The steady state is reached after approx. 580 iterations, which is not so bad compared to the results for the Hénon map (figure 4.3).

Figure 5.5 shows which period 1 orbit corresponds with the controlled fixed point of the Poincaré map. The other fixed point of figure 5.3(a) will probably correspond with a similar loop on the left side of the attractor.

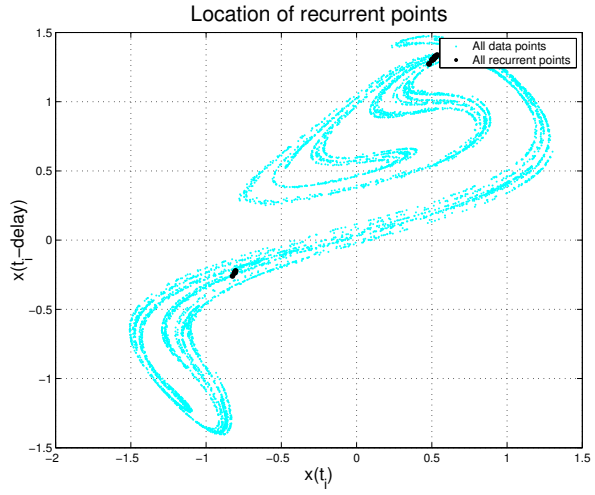


Figure 5.6: Location of $(1, \varepsilon)$ recurrent points for delay coordinate

5.2.3 Control with delay coordinates

Next, it is assumed that only the position x can be measured, so that delay coordinates have to be used. The delay coordinate defined as $z(t_i) = [x(t_i), x(t_i - \tau)]$, with $\tau = \frac{t_F}{4}$, yields the Poincaré map of figure 5.2(b). Analogously to the previous section, the recurrence method (with $\varepsilon = 0.01$) can identify the locations of the fixed points, see figure 5.6. The method described in section 5.2.1 can then be used to estimate the exact values of the two fixed points. In this case the adjustment iterations will stop when dx or dy is smaller than 10^{-5} (the convergence criterium) and again $\xi = 2\varepsilon$. This will return $\bar{z}_a = [0.5075, 1.3053]$ and $\bar{z}_b = [-0.8167, -0.2473]$. The method also returns the matrix A, which is given in appendix B.

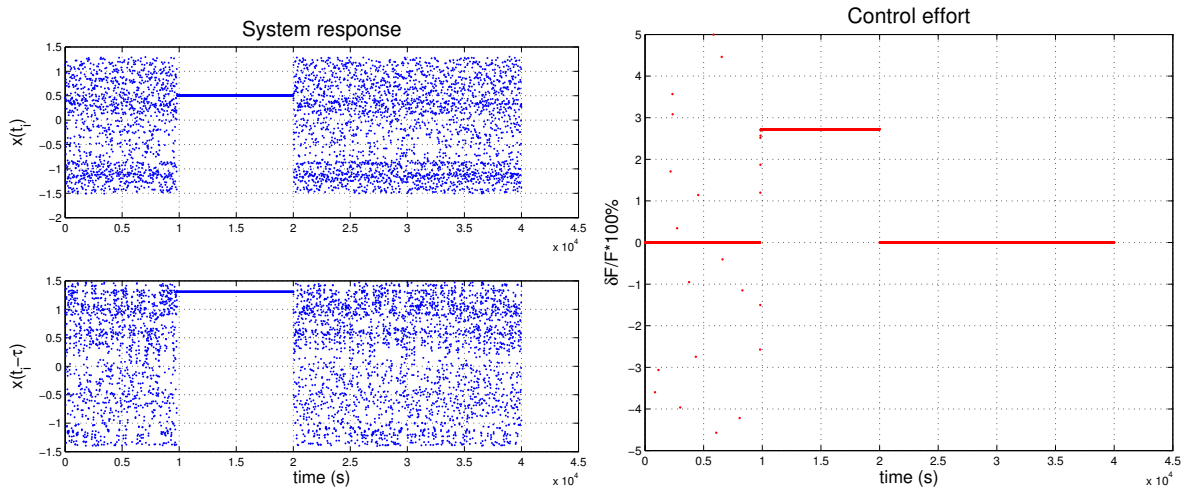
In order to find the matrix B a perturbed data series has to be generated. Recall section 3.2.3 where r was defined as the smallest integer for which $(n-1)\tau < rt_F$. Since τ was defined as $4\tau = t_F$ and $n = 3$, this reduces to $2 < 4r$, so $r = 1$. So besides F_i also F_{i-1} has influence on the state z_{i+1}. So to generate the data the system has to be perturbed for t_F seconds, after which the perturbation is switched off for t_F seconds, switched on, etc. The data series thus obtained can be used to find B^1 and B^2 (see appendix B).

To find K^T now, remember from section 3.2.3 that first $\tilde{A}$ and $\tilde{B}$ are defined as

$$\tilde{A} = \begin{pmatrix} A & B^2 \\ \mathbf{0} & 0 \end{pmatrix} \quad \text{and} \quad \tilde{B} = \begin{pmatrix} B^1 \\ 1 \end{pmatrix}. \tag{5.7}$$

and the new state is defined as $Y = [z, F_{i-1}]$. The new matrices are then used to apply the pole placement technique. The results are shown in figure 5.7. During the first 20,000 seconds the algorithm tries to control z_a, the second 20,000 seconds it tries to control z_b. Notice the large amount of false control attempts in the first 10,000 seconds. As was discussed before, this is due to the fact that $F - \bar{F} = -K^T(Y_i - Y^*)$ can be zero when $Y_i \neq Y^*$. Furthermore, notice that there is *no* effort to control the second fixed point. To understand this, compare the control matrices K for each of the two points:

$$K_a = \begin{bmatrix} 0.8167 \\ -0.8802 \\ -0.4320 \end{bmatrix} \quad \text{and} \quad K_b = \begin{bmatrix} -60.9028 \\ 43.6503 \\ 26.7399 \end{bmatrix}.$$



(a) System response as a function of time

(b) Control effort as a percentage of δF_{max}

Figure 5.7: OGY control with delay coordinates

Due to the large values of K_b for the second fixed point, the control effort defined as $\delta F = -K^T(Y_i - Y^*)$ will in general be larger than δF_{max} and thus set to zero. In other words, it takes much longer before $\delta F < \delta F_{max}$ and control is allowed.

The next section will provide a different approach for delay coordinates which will deal with this problem.

5.2.4 Alternative method for delay coordinates

When delay coordinates are used, it is assumed that the control is applied continuously between two intersections of the Poincaré map, so the control lasts t_F seconds; $\delta F_i \neq 0$ from t_i till t_{i+1}. Since δF_i is active at time t_{i+1} delay coordinates have to be used. If the control is switched off *before* the next intersection, the delay coordinates are not needed though.

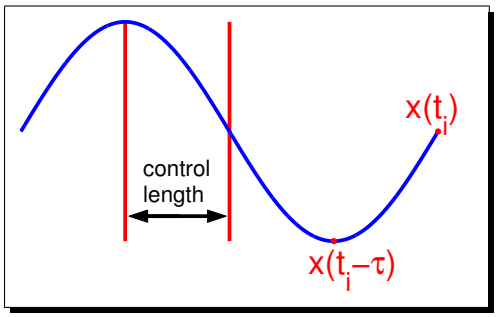


Figure 5.8: Moment of perturbation

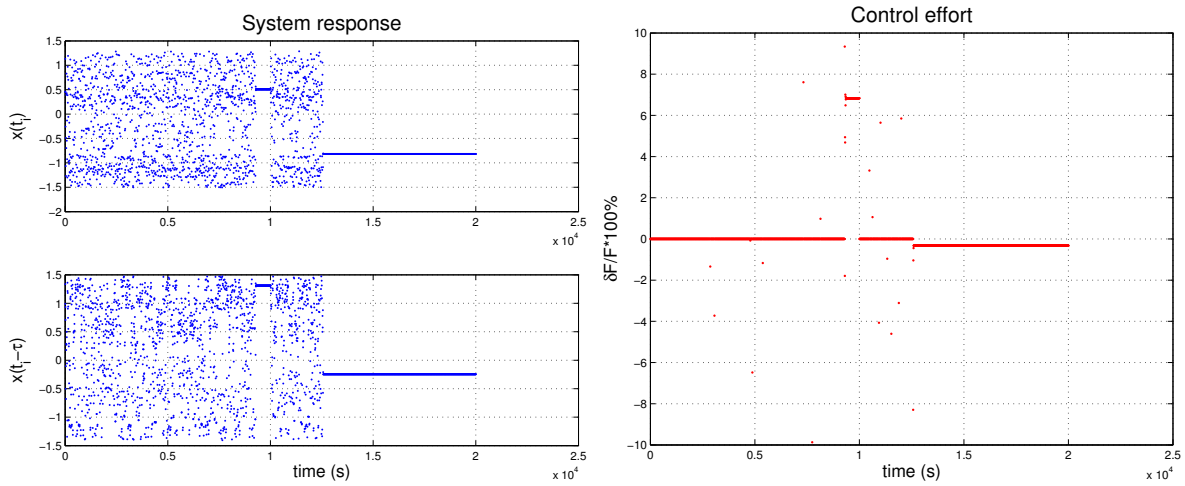
So here the parameter perturbation will be applied on just a quarter of the forcing phase, see figure 5.8. This picture shows one forcing period t_F where the perturbation is applied on the second quarter of this period. The delay coordinate is formed by the value at $\frac{3}{4}t_F$ and t_F, so the perturbation has no direct effect on the delay coordinate. When control is applied this way, the adjustment of OGY discussed in section 3.2.3 is not necessary and the strategy of section 5.2.2 can also be applied to the delay coordinate situation. The one difference is that B has to be re-estimated, since B will now describe the influence of a shorter perturbation. The data series used for B is generated by turning on the perturbation during the intervals $[kt_i + \tau, kt_i + 2\tau]$ where $k = 0, 2, 4, \ldots$. The series is then divided into two groups, as was done in section 4.2.3.

5.2.4.1 Controlling the fixed point

Using the method described above, the new matrix B can be found as given in appendix B. The control with which the used data series is made, is slightly larger than before, namely $\delta F_{max} = 0.1F$. This is done because the influence of δF is expected to be smaller than before, since the perturbation is now applied in less time. Both the estimates of the fixed points and the matrices A remain the same as in section 5.2.3. The control matrices K^T can now be calculated without using (5.7), and are equal to

$$K_a = \begin{bmatrix} 2.5301 \\ -2.7268 \end{bmatrix} \qquad \text{and} \qquad K_b = \begin{bmatrix} -0.8644 \\ 0.6195 \end{bmatrix},$$

so the problem of the previous section seems to be solved. This is also shown when these matrices are used to apply OGY control, see figure 5.9, where the first 10,000 seconds point z_a is controlled, and the next 10,000 seconds point z_b. Both points are now successfully controlled. Point z_b seems to be controlled much easier than point z_a; notice the larger control in steady state and the larger time to achieve control for point z_a. Probably the estimates of point z_b are slightly better than for point z_a. Compare e.g. x and $\bar{x}(t_i)$ in appendix B, both representing the same point (z_a): 0.5079 vs. 0.5075. Since the control of section 5.2.2 looks more successful than the current result, the former estimation is probably better.



(a) System response

(b) Control effort in relation to F

Figure 5.9: Results for adjusted delay coordinates method; $\delta F_{max} = 0.1F$

The corresponding orbits in the continuous time domain are shown in figure 5.10. Indeed, as section 5.2.2 already suggested, both fixed points of the delay coordinate correspond to a simple orbit on either side of the chaotic attractor.

5.2.4.2 Controlling the period 2 cycle

Appendix B also shows the values of the estimated Poincaré points of the period 2 orbit, together with the corresponding values for A and B. Using these values and the adjusted delay coordinates method introduced before, is should also be possible to control this orbit.

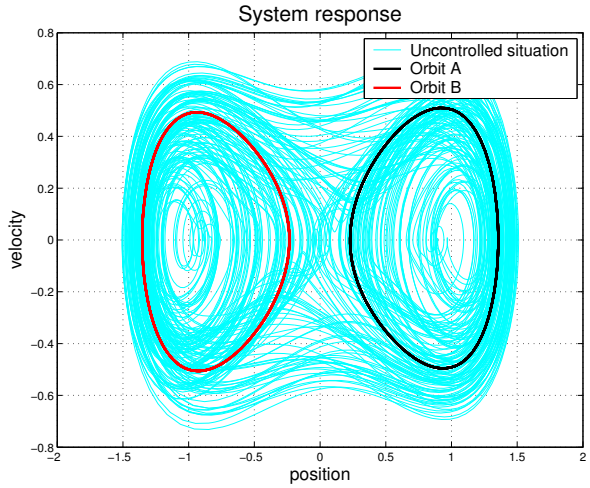


Figure 5.10: Period 1 orbits

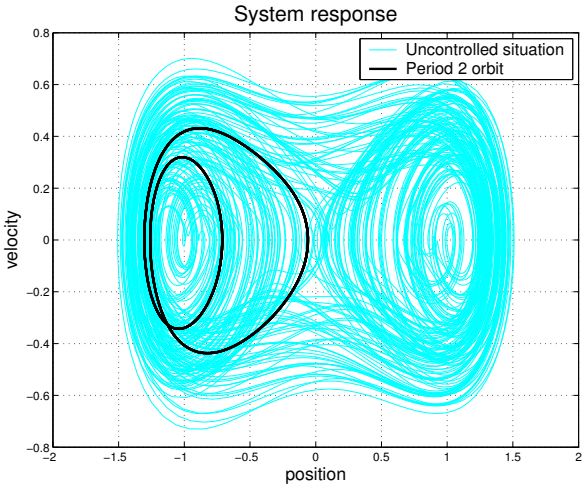
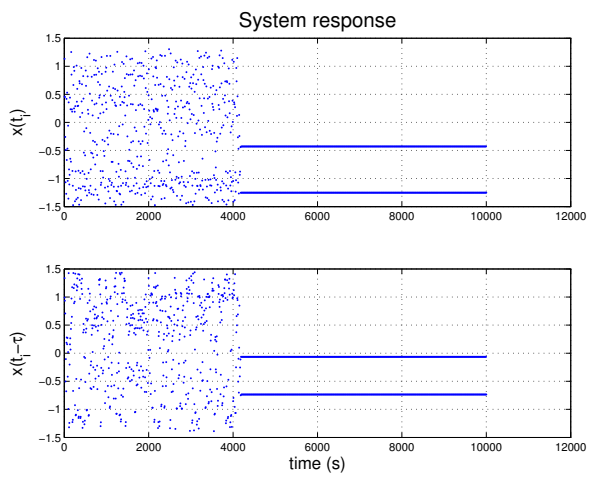
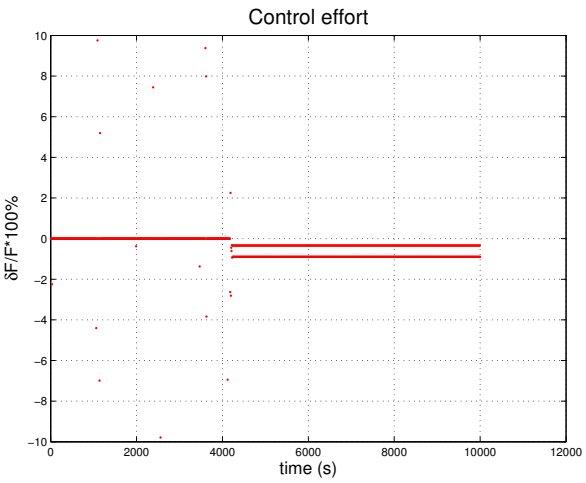


Figure 5.11: Period 2 orbit

The results are shown in figure 5.12, where it is clear to see that indeed a period 2 orbit is controlled. Notice the relatively small time to achieve control and the small steady state control efforts. This could indicate that the estimation of the Poincaré points of this orbit is quite accurate. Finally, figure 5.11 shows the representation in the continuous time domain. It seems reasonable that a similar period 2 orbit lies on the right side of the attractor. Controlling this orbit could be a problem though. It is almost impossible to predict which delay coordinate Poincaré recurrent points belong to a certain orbit. In other words, if it is desirable to control a certain continuous time orbit, one can only guess its period, guess which delay coordinate recurrent points belong to this orbit, and hope that after the application of OGY it indeed matches the desired orbit. This trial-and-error process is beyond the scope of this report.



(a) System response



(b) Control effort in relation to F

Figure 5.12: Control of period 2 orbit with adjusted delay coordinates method; $\delta F_{max} = 0.1F$

5.3 Implementation in Simulink®

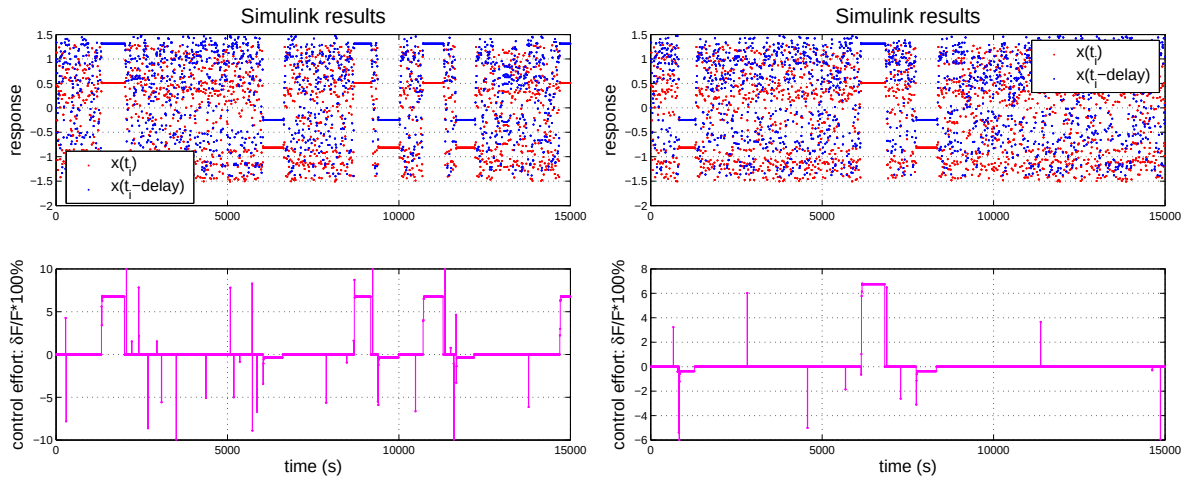
All the algorithms used before were implemented in Matlab® M-files. It is also possible to implement the OGY algorithm in Simulink®. One of the big advantages of using Simulink® is that it is easier to play around with the algorithm, for example by changing the solver, switching between fixed points or altering parameters. The estimation of fixed points, A and B and the calculation of K^T will still have to be done in Matlab®, but of course the previously obtained results can be used. The following implementation will be based on section 5.2.4. The model will thus use delay coordinates and the associated adjusted control algorithm where the perturbation is only applied for $\frac{1}{4}t_F$ seconds.

5.3.1 The models

Two different models were used for simulation: for both the period 1 and the period 2 orbits. The models are shown in appendix D. In this appendix, figure D.1 shows the main model for the period 1 case together with its specific control submodel. Notice the switch in the main model, so that it is possible to switch between fixed point at any point during simulation. Figure D.2 shows the same situation for the period 2 case. Now notice the switch inside the control submodel, so that the control law can be switched on and off. Furthermore, figures D.3(a) and D.3(b) show two submodels used by both models; the Duffing oscillator and the submodel creating the delay coordinate vector.

5.3.2 Results

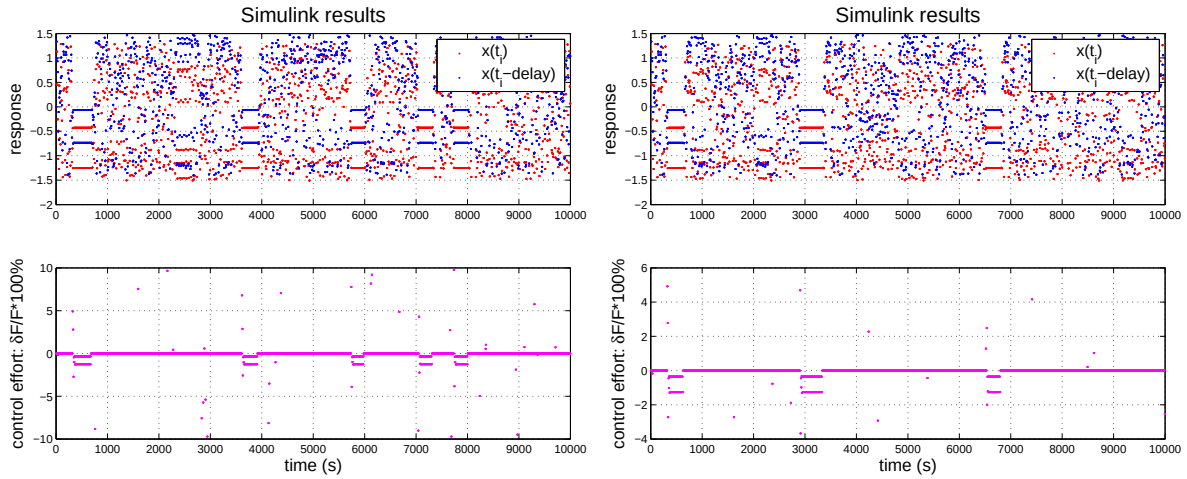
This section will show and discuss some results of the Simulink® simulations done with the models introduced in the previous section. Figure 5.13 shows some results for the period 1 case. The left figure starts with controlling fixed point z_a and switches to fixed point z_b a few seconds after the controlled state is established, switches back to z_a when z_b is successfully



(a) Results with $\delta F_{max} = 0.1F$

(b) Results with $\delta F_{max} = 0.07F$

Figure 5.13: Some Simulink® results for period 1 orbits



(a) Results with $\delta F_{max} = 0.1F$

(b) Results with $\delta F_{max} = 0.05F$

Figure 5.14: Some Simulink® results for period 2 orbit

controlled, and so forth. The right figure shows the same, but starts with fixed point z_b. Furthermore, both simulations used different values for δF_{max}, illustrating its influence. In figure 5.13(a) δF_{max} is 10% of F. In 15,000 seconds of simulation seven controlled states where established. Notice that controlling point z_a seems more difficult than point z_b, as was also concluded in section 5.2.4.1: point z_a needs a larger steady state control and needs more time to achieve control. The figure also shows that this time is to a large extent subject to the ergodic behavior of the chaotic system. Two out of three situations where point z_b is controlled need very little time, while its first attempt takes a lot of time.
Figure 5.13(b) uses a δF_{max} of 7% of F, which is slightly more than the steady state control effort needed to control point z_a. As can be expected, it takes far more time now before point z_a is controlled, resulting in only three controlled states in 15,000 seconds. Controlling point z_b is still relatively easy.

Figure 5.14 shows similar results for the period 2 orbit. In both the left and the right figure the control was switched off for a short moment as soon as a controlled state was established. The difference between the figures is again caused by different values of δF_{max}. In figure 5.14(b) this value is half the size as in figure 5.14(a), resulting in a larger time to achieve control and thus less controlled states (three instead of five) in 10,000 seconds of simulation. Furthermore, notice that in figure 5.14(a) it seems that around 2500 seconds the system is accidentally very close to a different higher periodic orbit embedded within the chaotic attractor.

5.4 Concluding remarks

This chapter showed that the OGY algorithm can successfully be used for continuous time systems. In all cases, except for section 5.2.3, a controlled state, i.e. a periodic orbit, was established. The uncontrolled problem was solved in the following section, where the delay coordinate algorithm was successfully adjusted. The same conclusions as for the Hénon applied, e.g. the presence of a steady state control effort and setpoint error. In conclusion,

OGY is a very useful tool to control any chaotic system towards a certain periodic solution, even when detailed measurement information is not available. Furthermore, this chapter showed once again the influence of the maximum allowed perturbation, δF_{max}. Its value should *at least* be larger than the steady state control effort, and increasing its value (subject to system restrictions) will decrease the time to achieve control.

There are some additional restrictions of OGY though. As was discussed in section 5.2.4.2 one cannot define a desired arbitrary orbit beforehand, since one is restricted to the orbits embedded within the chaotic attractor. It is even impossible to predict beforehand which shape these embedded orbits have. Not until the control is successfully applied, one can know what exactly the OGY was controlling. OGY only turns chaotic behavior into periodic behavior, but the exact shape and value of the periodic orbit cannot be controlled.

The influence of noise was not investigated in this chapter, but the observations of section 4.3 are also valid here. As long as the estimates of fixed points and the matrices A and B are quite accurate OGY can overcome some amount of noise. Too much noise on the estimates can make them very poor though, and the OGY control algorithm will undoubtedly fail.

The remark that "OGY works" should be taken with a grain of salt. Recall again the huge times to achieve control of this chapter: hundreds or thousands of seconds. Although section 5.2.2 gives an explanation for this, it is still not desirable to wait half an hour before a real-time system is finally controlled. Conclusions and recommendations considering this and other problems are given in the next chapter.

Chapter 6

Conclusions and recommendations

6.1 Conclusions

The basic observation behind the control of chaos is that any continuous or discrete time chaotic system $\dot{x} = f(x, p)$ or $x_{i+1} = f(x_i, p_i)$, with control parameter p, has a large number of periodic orbits embedded within its chaotic attractor. The OGY control algorithm controls a chaotic system towards one or more of these orbits by perturbing the parameter p with small amounts. Because the chaotic attractor is ergodic, the solution will at some point in time come into the vicinity of the considered orbit where a linearization and a simple pole placement control method can be applied.

This OGY control seems to work for any system, whether discrete or continuous, with or without delay coordinates. Very important in its implementation is the accuracy of the estimates for fixed points and the matrices A and B. As long as the estimates are relatively accurate, OGY will indeed manage to turn the chaotic behavior into a periodic solution using only small perturbations. When the fixed point is not estimated correctly, there will be a steady state control effort. The worse the estimates are, the larger this effort will be, in the end resulting in a failure of the OGY control algorithm.

In order to improve the estimates this report suggests a method where the fixed point and the matrix A where calculated iteratively, until the offset matrix C became small enough. When $C = 0$ the estimation of the fixed point is expected to be as close to the real fixed point as possible and therefore A and B are as good as possible.

The dependency on the estimates means that when there's noise present in the estimation process, the OGY algorithm might not manage to control chaos. As long as the estimates are accurate, OGY can cope with noise very well though. As long as the solution, subject to noise, stays within the vicinity of the orbit (determined by the maximum parameter perturbation δp_{max}), control is guaranteed. But larger noise values can push the system outside this vicinity and the solution will return to chaos.

This report also showed the influence of δp_{max}. In general, small values of δp_{max} will result in large times to achieve control. When δp_{max} is smaller than the steady state control effort, the controlled state will never be reached and OGY will fail. Besides a smaller time to achieve control, higher values of δp_{max} will result in more false control attempts, since not every point where control is applied lies within the linear vicinity of the orbit.

The orbits controlled by OGY are limited to the ones embedded within the attractor. OGY only turns chaos into a periodic solution, but which solution this is cannot be determined

beforehand and its shape cannot be modified by the control law. It is even impossible to control one of the state components (i.e. an output $y(t)$) towards a certain arbitrary trajectory $y_d(t)$. This is a drawback compared to other (non-linear) controllers, which can often control the output towards *any* desired trajectory. Furthermore, these non-linear controllers can obtain much smaller times to achieve control, not depending on the ergodic behavior of the chaotic system. Also, recall that the OGY algorithm results in a steady state setpoint error, while most non-linear controllers will have smaller or even no steady state errors. Take, for example, a computed torque controller, which guarantees that the tracking error becomes zero for $t \to \infty$, following any desired trajectory, $y(t) \to y_d(t)$.

OGY needs a very small control effort though, which is clearly an advantage, e.g. for systems where parameters can not be changed a lot due to various restrictions. Another advantage is that a model is not needed. Using delay coordinates it's sufficient to measure only one state component. Nowadays there are lots of different model estimation methods and model-based non-linear controllers available though, so the question remains how useful this really is. Chaotic systems might be difficult to model, but keeping in mind the drawbacks mentioned before, often using a continuous non-linear controller for a poor model of the system will give faster and better results than OGY does.

Finally, it should be noted that OGY works differently than Ott, Grebogi and Yorke initially thought. Instead of leaving the stable eigendirections of the orbit in tact, the OGY algorithm, which is nothing more than a pole placement method, changes both all the eigenvectors and the unstable eigenvalues of the system. This is obvious, since the eigenvalues and the eigenvectors of the closed loop $A - BK^T$ are different than those of A.

6.2 Recommendations

The time to achieve control for the OGY algorithm is in general very large and unpredictable, since the ergodic behavior of the chaotic system determines when the solution is close enough to the orbit. This makes OGY not very attractive, but there are some possibilities to improve this.

1. As was mentioned in section 4.3, there are targeting techniques available, which are able to target the solution towards an orbit even when it's far outside its vicinity.

2. Notice that since OGY is a discrete time algorithm, it only tries to control at discrete moments in time. In case of the Duffing oscillator this means that only every $\frac{2\pi}{\omega} = 7.48s$ there is a possibility to control, once per forcing period. OGY would perform faster if this could be changed to much more moments, to ideally an infinite amount of points. It should thus be investigated whether OGY can be written as a continuous time algorithm.

3. Instead of using linear estimates, non-linear least squares methods can be used to obtain A and B. For example for the Hénon map a quadratic least squares fit might be more accurate in a larger vicinity around the considered (fixed) point. This increasing size of the vicinity can result in smaller times to achieve control.

In addition the the latter point, the simple linear least squares method used to find A and B, is not allowed when there's a correlation between the separate $x_{i+1,j}$ in (3.28). This wasn't the case for the systems in this report, but in some cases the possibility of using Partial Least Squares should be considered.

Although this report assumes that the models of the used systems where unknown, all results are purely based on numerical simulations. It still can not guarantee the success of OGY in real experimental systems. Therefore, it is recommended to test the algorithm described in this report on experimental chaotic systems. Various difficulties may then arise, including measurement noise. Furthermore, the construction of the Poincaré map may be much more difficult, especially when there is no periodic force present. In that case, samples should be taken when, for example, position or velocity is at a certain value, which can be very difficult or even impossible. This will also cause the time between samples to be different, which calls for reconsideration of the length of time of the applied control (section 5.2.4).

Finally, one can ask why the control of chaos is necessary. First of all, there aren't a lot of real life applications of chaos. Secondly, in some situations it's not clear whether periodic behavior of a system is preferred over chaotic behavior. This could depend on the consequences of either behavior, expressed in e.g. energy absorption or material fatigue. Or maybe the price of perturbing the parameter(s) is higher than the gain of periodic over chaotic behavior.

References

[1] J. Ackermann. Der entwurf linearer regelungssysteme im zustandsraum. *Regulungestechnik und Prozessedatenverarbeitung*, 7:297–300, 1972.

[2] Robert L. Devaney. *An Introduction to Chaotic Dynamical Systems*. The Benjamin/Cummings Publishing Co., Inc., 1986.

[3] W. L. Ditto, S. N. Rauseo, and M. L. Spano. Experimental control of chaos. *Physical Review Letters*, 65(26):3211–3214, 1990.

[4] Ute Dressler and Gregor Nitsche. Controlling chaos using time delay coordinates. *Physical Review Letters*, 68(1):1–4, 1992.

[5] G. Duffing. *Erzwungene Schwingungen bei veränderlicher Eigenfrequenz und ihre technische Bedeutung*. Vieweg, Brunswick, 1918.

[6] J. P. Eckmann and D. Ruelle. Ergodic theory of chaos and strange attractors. *Review of Modern Physics*, 57(3):617–655, 1985.

[7] Celso Grebogi and Ying-Cheng Lai. Controlling chaotic dynamical systems. *Systems & Control Letters*, 31:307–312, 1997.

[8] M. Hénon. A two dimensional mapping with a strange attractor. *Commun. Math. Phys.*, 50:69–77, 1976.

[9] Eric J. Kostelich, Celso Grebogi, Edward Ott, and James A. Yorke. Higher-dimensional targeting. *Physical Review E*, 47(1):305–310, 1993.

[10] Ying-Cheng Lai, Mingzhou Ding, and Celso Grebogi. Controlling hamiltonian chaos. *Physical Review E*, 47(1):86–92, 1993.

[11] Daniel P. Lathrop and Eric J. Kostelich. Characterization of an experimental strange attractor by periodic orbits. *Physical Review A*, 40(7):4028–4031, 1989.

[12] Henk Nijmeijer. Control of chaos and synchronization. *Systems & Control Letters*, 31:259–262, 1997.

[13] Edward Ott, Celso Grebogi, and James A. Yorke. Controlling chaos. *Physical Review Letters*, 64(11):1196–1199, 1990.

[14] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw. Geometry from a time series. *Physical Review Letters*, 45(9):712–716, 1980.

[15] H. Poincaré. Mémoire sur les courbes définies par une équation différentielle. *J. Mathématiques*, 7:375–422, 1881.

[16] Filipe J. Romeiras, Celso Grebogi, Edward Ott, and W. P. Dayawansa. Controlling chaotic dynamical systems. *Physica D*, 58:165–192, 1992.

[17] O. E. Rössler. An equation for continuous chaos. *Physics Letters A*, 57:397–398, 1976.

[18] Shankar Sastry. *Nonlinear Systems: Analysis, Stability, and Control*. Springer-Verlag, 1999.

[19] Troy Shinbrot, Edward Ott, Celso Grebogi, and James A. Yorke. Using chaos to direct trajectories to targets. *Physical Review Letters*, 65(26):3215–3218, 1990.

[20] Paul So and Edward Ott. Controlling chaos using time delay coordinates via stabilization of periodic orbits. *Physical Review E*, 51(4):2955–2962, 1995.

[21] Steven H. Strogatz. *Nonlinear Dynamics and Chaos*. Perseus Books Publishing, LLC, 1994.

[22] Lawrence N. Virgin. *Introduction to Experimental Nonlinear Dynamics*. Cambridge University Press, 2000.

[23] Herman Wold. Partial least squares. *Encyclopedia of Statistical Sciences*, 6:581–591, 1985.

Appendices

Appendix A

Estimation results for Hénon

In this appendix chapter some results are presented for the implementation of the OGY control algorithm to the Hénon map, given by

$$x_{i+1} = y_i + 1 - ax_i^2$$
$$y_{i+1} = bx_i.$$

The theoretical stable fixed point of this map is

$$\begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} 0.6314 \\ 0.1894 \end{bmatrix},$$

and its corresponding matrices A and B, assuming a as the accessible parameter, are

$$A = \begin{bmatrix} -2ax & 1 \\ b & 0 \end{bmatrix} \quad \text{and} \quad B_a = \begin{bmatrix} -x^2 \\ 0 \end{bmatrix}.$$

In section 4.2.2 the coordinates of the period 1, 2, 4 and 13 were estimated using the recurrence methods. These coordinates are, written in matrixform:

$$\text{period 1:} \quad \begin{bmatrix} 0.6309 \\ 0.1891 \end{bmatrix}$$

$$\text{period 2:} \quad \begin{bmatrix} -0.4752 & 0.9760 \\ 0.2921 & -0.1425 \end{bmatrix}$$

$$\text{period 4:} \quad \begin{bmatrix} 0.6380 & 0.2181 & 1.1251 & -0.7067 \\ -0.2122 & 0.1917 & 0.0654 & 0.3375 \end{bmatrix}$$

$$\text{period 13:} \quad \begin{bmatrix} -0.2068 & 0.6223 & 0.3957 & 0.9675 & -0.1932 & 1.2384 & \dots \\ -0.3178 & -0.0621 & 0.1867 & 0.1185 & 0.2907 & -0.0579 & \end{bmatrix}$$
$$\begin{bmatrix} \dots & -1.2032 & -0.6556 & 0.0372 & 0.8014 & 0.1111 & 1.2228 & -1.0601 \\ & 0.3713 & -0.3610 & -0.1967 & 0.0112 & 0.2401 & 0.0332 & 0.3669 \end{bmatrix}.$$

In section 4.2.3 also the matrices A and B are estimated for all of these points. Since it's unnecessary and paper filling to give all these 20 matrices A, only those for period 1 and 2 will be given here. For period 4 and 13 the eigenvalues of the orbit (i.e. the eigenvalues λ of the product of the matrices) are given.

$$A_1 = \begin{bmatrix} -1.7666 & 0.9996 \\ 0.3000 & -0.0000 \end{bmatrix}$$

$$A_{2,1} = \begin{bmatrix} 1.3285 & 0.9929 \\ 0.3000 & 0.0000 \end{bmatrix} \qquad A_{2,2} = \begin{bmatrix} -2.7297 & 0.9960 \\ 0.3000 & 0.0000 \end{bmatrix}$$

$$\begin{aligned}\lambda_4 &= \lambda(A_{4,1}\cdots A_{4,4}) &&= [-8.6464; -0.0010]\\ \lambda_{13} &= \lambda(A_{13,1}\cdots A_{13,13}) &&= [-8.0490; 0.0000]\end{aligned}$$

Indeed the eigenvalues show that the total orbit has one stable ($|\lambda| < 1$) and one unstable ($|\lambda| > 1$) direction. The offset matrix C can also be calculated, and should be zero. This is not exactly true though:

$$\begin{aligned}C_1 &= 10^{-3}\begin{bmatrix} 0.7651 \\ 0.2302 \end{bmatrix}\\ C_{2,1} &= 10^{-3}\begin{bmatrix} -0.1261 \\ -0.0650 \end{bmatrix} \quad C_{2,2} = \begin{bmatrix} -0.0011 \\ 0.0007 \end{bmatrix}\end{aligned}$$

The matrices B are only given for period 1, 2 and 4:

$$\begin{aligned}B_1 &= \begin{bmatrix} -0.3563 \\ 0.0115 \end{bmatrix}\\ B_{2,1} &= \begin{bmatrix} -0.2290 \\ -0.0032 \end{bmatrix}, B_{2,2} = \begin{bmatrix} -0.9970 \\ 0.0339 \end{bmatrix}\\ B_{4,1} &= \begin{bmatrix} -0.4111 \\ -0.0134 \end{bmatrix}, B_{4,2} = \begin{bmatrix} -0.0487 \\ 0.0000 \end{bmatrix}, B_{4,3} = \begin{bmatrix} -1.2531 \\ 0.0000 \end{bmatrix}, B_{4,4} = \begin{bmatrix} -0.4807 \\ 0.0090 \end{bmatrix}.\end{aligned}$$

To examine the accurateness of these values, look at the actual value of B_1 at the estimated fixed point:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0.6309 \\ 0.1891 \end{bmatrix} \Rightarrow B_1 = \begin{bmatrix} -x^2 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.3981 \\ 0 \end{bmatrix}$$

This means that the error in B_1 is more than 10%. For other points this error in B is approx. between 0% and 15%.

Appendix B

Estimation results for Duffing

In this appendix chapter some results are presented for the implementation of the OGY control algorithm to the Duffing oscillator, given by

$$\ddot{x} + 2\xi\dot{x} + \frac{1}{2}x = F\Omega^2 \sin \Omega t,$$

in which F is chosen as the control parameter.

When both position and velocity can be measured, there's no need to use delay coordinates. The recurrence method and the adjustment of the fixed point estimation then finds a fixed point

$$\begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 0.5079 \\ -0.3535 \end{bmatrix},$$

with the corresponding matrix A and remaining small offset C:

$$A = \begin{bmatrix} -3.8958 & -6.7048 \\ 0.0161 & -0.1131 \end{bmatrix} \qquad C = 10^{-5} \begin{bmatrix} -0.0112 \\ 0.4123 \end{bmatrix}.$$

Using a perturbation $\delta F_{max} = 0.05F$ a data series can be generated with which B can be found as

$$B = \begin{bmatrix} -5.4938 \\ -0.7330 \end{bmatrix}.$$

Delay coordinates are necessary when only position can be measured. For the delay coordinate $z(t_i) = [x(t_i), x(t_i - \tau)]$, the recurrence method and the adjustment of the fixed point estimation are able to calculate two fixed points and a period 2 orbit:

$$\begin{bmatrix} \bar{x}(t_i) \\ \bar{x}(t_i - \tau) \end{bmatrix}_{1,a} = \begin{bmatrix} 0.5075 \\ 1.3053 \end{bmatrix} \qquad \begin{bmatrix} \bar{x}(t_i) \\ \bar{x}(t_i - \tau) \end{bmatrix}_{1,b} = \begin{bmatrix} -0.8167 \\ -0.2473 \end{bmatrix},$$
$$\begin{bmatrix} \bar{x}(t_i) \\ \bar{x}(t_i - \tau) \end{bmatrix}_{2} = \begin{bmatrix} -0.4287 & -1.2510 \\ -0.0648 & -0.7362 \end{bmatrix}.$$

The adjustment method also returns A and the offset C:

$$A_{1,a} = \begin{bmatrix} -5.1180 & 5.3419 \\ -1.5385 & 1.4965 \end{bmatrix} \qquad C_{1,a} = 10^{-16} \begin{bmatrix} 0.1525 \\ -0.8388 \end{bmatrix}$$
$$A_{1,b} = \begin{bmatrix} -5.5756 & 3.8982 \\ -2.3079 & 1.5174 \end{bmatrix} \qquad C_{1,b} = 10^{-16} \begin{bmatrix} -0.1133 \\ 0.0991 \end{bmatrix}$$
$$A_{2,1} = \begin{bmatrix} 0.4824 & -0.0233 \\ -1.3269 & 0.5916 \end{bmatrix} \qquad C_{2,1} = 10^{-16} \begin{bmatrix} 0.3747 \\ -0.5538 \end{bmatrix}$$
$$A_{2,2} = \begin{bmatrix} -15.4317 & 8.1570 \\ -4.7384 & 2.4191 \end{bmatrix} \qquad C_{2,2} = 10^{-15} \begin{bmatrix} 0.9061 \\ 0.2762 \end{bmatrix}.$$

The perturbation matrices B^1 (influence of δF_i) and B^2 (influence of δF_{i-1}) for the fixed points were found using $\delta F_{max} = 0.05F$:

$$B^1_{1,a} = \begin{bmatrix} -4.8474 \\ -1.0575 \end{bmatrix} \qquad B^1_{1,b} = \begin{bmatrix} 1.5454 \\ 1.4537 \end{bmatrix}$$
$$B^2_{1,a} = \begin{bmatrix} 2.6853 \\ 0.7935 \end{bmatrix} \qquad B^2_{1,b} = \begin{bmatrix} 2.3862 \\ 0.9270 \end{bmatrix}$$

When the adjusted delay coordinate method of section 5.2.4 is used, the values of $\bar{z}(t_i)$ and A remain the same. In that case there's only one B at each point, instead of B^1 and B^2. Using $\delta F_{max} = 0.1F$, B becomes:

$$B_{1,a} = \begin{bmatrix} -1.4256 \\ -0.0539 \end{bmatrix} \qquad B_{1,b} = \begin{bmatrix} 6.2522 \\ 2.3932 \end{bmatrix}$$
$$B_{2,1} = \begin{bmatrix} -0.2289 \\ 0.2358 \end{bmatrix} \qquad B_{2,2} = \begin{bmatrix} 13.3566 \\ 3.9703 \end{bmatrix}$$

Appendix C

Matlab® code

C.1 The Hénon map

```
% Henon Parameters
a = 1.4; b = 0.3;
iter = 100000;

% Creation of data points
data = [0;0];
for i = 1:iter;
    data(1,i+1) = data(2,i)+1-a*data(1,i)^2;
    data(2,i+1) = b*data(1,i);
end
xrange = max(data(1,:))-min(data(1,:));
yrange = max(data(2,:))-min(data(2,:));
```

C.2 The Duffing oscillator

The code for delay coordinate reconstruction of Duffing:

```
% Duffing parameters
ksi = 0.04;
omega = 0.84;
F = 0.188;

% Defining delay
period = 2*pi/omega;
deel = 4;
delay = periode/deel;

% Generating data series
begin = [0 0];
endtime = 200000;
nop_end = ceil(endtime/period);
endtime = nop_end*period;
data = [];
for i = 0:(nop_end-1);
    timedc = [i*period , (i+1)*period-delay , (i+1)*period];
    [tdc,xdc] = ode45(@duffing,timedc,begin,options,ksi,omega,F);
    data = [data , [xdc(3,1) ; xdc(2,1)] ];
    begin = xdc(end,:);
end

xrange = max(data(1,:))-min(data(1,:));
yrange = max(data(2,:))-min(data(2,:));
iter = length(data)-1;
```

This code uses the function duffing.m:

```
function dx = duffing(t,x,ksi,omega,F);
dx = zeros(2,1);

dx(1) = x(2);
dx(2) = F*omega^2*sin(omega*t) - 2*ksi*x(2) - 0.5*x(1)^3 + 0.5*x(1);
```

C.3 Estimation tools

The scripts in sections C.3.1, C.3.2 and C.3.3 were used for the Hénon map. Section C.3.4 shows the script for the adjustment of the estimations for fixed points and A and B for the Duffing oscillator, described in section 5.2.1.

C.3.1 Estimating fixed points and orbits

```
% Recurrence parameters
ksi_grens = 0.002; maxrecur = 15;

% Recurrence iterations
veld = sparse(1,iter);
for i = 1:(length(data)-1);
    for j = i+1 : length(data);
        ddata = data(:,i) - data(:,j);
        ddata = [ddata(1,:)/xrange ; ddata(2,:)/yrange];
        ksi = norm(ddata,2);
        if ksi < ksi_grens;
            veld(1,i) = j-i; break;
        elseif j-i >= maxrecur ; break;
        else ;
        end
    end
end
clear i j ddata ksi;
```

For every period m of interest, perform the following:

```
periode = m;

% Recurrent points for period m
[i,j] = find(veld==periode);
for n = 1:length(j);
    rec(1,n) = j(n);
    rec(2:3,n) = data(:,j(n));
end
clear n;

% Searching longest sequence
len = 1;
lengte = 0;
for n = 1:length(rec)-1;
    if rec(1,n+1)-rec(1,n) == 1;
        len = len + 1;
    else if len > lengte;
            lengte = len;
            begin = n+1-lengte;
            eind = n;
        else ;
        end
        len = 1;
    end
end
if len > lengte;
```

```
    lengte = len;
    begin = n+2-lengte;
    eind = n+1;
else ;
end
clear len n;
serie = rec(2:3,begin:eind);
sd = ones(1,lengte);

% Creating extra points if needed
if lengte < periode;
    n = [rec(1,eind)+1 : rec(1,eind)+periode-lengte];
    serie(:,lengte+1:periode) = data(:,n);
    sd(1,lengte+1:periode) = ones(1,periode-lengte);
    lengte = periode; eind = begin+periode-1;
end
clear n;

% Gathering all points in each others vicinity
for n = [1:begin-1 , eind+1:length(rec)];
    for m = 1:lengte;
        dx = rec(2:3,n) - serie(:,m);
        dx = [dx(1,:)/xrange ; dx(2,:)/yrange];
        if norm(dx,2) < ksi_grens;
            sd(1,m) = sd(1,m) + 1;
            serie(:,m,sd(1,m)) = rec(2:3,n); break
        else ;
        end
    end
end
clear n m dx;

% Averaging all points
perpunt = serie(:,1:periode,:);
diepte = sd(1,1:periode);
for n = 1:lengte-periode;
    ratio = ceil(n/periode)-1;
    oud = diepte(1,n-ratio*periode);
    diepte(1,n-ratio*periode) = oud + sd(1,n+periode);
    perpunt(:,n-ratio*periode,1+oud:diepte(1,n-ratio*periode)) = serie(:,n+periode,1:sd(1,n+periode));
end
clear n sd serie ratio oud;
for n = 1:periode;
    pm(:,n) = sum(perpunt(:,n,1:diepte(1,n)),3)/diepte(1,n);
end
clear n;
```

Now p_m contains all m points of the period m orbit.

C.3.2 Estimating A

The following function estimates A:

```
function [A,C] = algorithm_a(data,pn,ksi_grens,grootte,fignr);

periode = size(pn,2);
dim = size(pn,1);

% Looking for points to estimate with
[orig,afb] = zoeka(data,pn,grootte,ksi_grens);
% Calculate A and the small perturbation C
for i = 1:periode;
    temp_afb = afb(:,:,i);
    while temp_afb(:,end) == [0;0];
        temp_afb = temp_afb(:,1:(end-1));
```

```
    end
    temp_orig = orig(:,1:size(temp_afb,2),i);
    n = size(temp_orig,2);
    x = [ones(1,n) ; temp_orig]';
    y = temp_afb';
    c = x\y;
    A(:,:,i) = c(2:end,:)';
    C(:,:,i) = c(1,:)';
end
```

This function uses a function to find points close to the orbit:

```
function [orig,afb] = zoeka(data,p,grootte,ksi_grens);

xrange = max(data(1,:))-min(data(1,:));
yrange = max(data(2,:))-min(data(2,:));

% 'grootte' should be as long as p
if length(grootte) == 1;
    grootte = ones(1,size(p,2))*grootte;
elseif length(grootte) == size(p,2);
else error('De vector grootte heeft niet dezelfde lengte als p');
end

% Searching points within 'ksi_grens'
punt = [ p , p(:,1) ];
k = ones(size(p,2),1);
for i = 1:(length(data)-1);
    for j = 1:size(punt,2)-1;
        ddata = data(:,i) - punt(:,j);
        ddata = [ddata(1,1)/xrange ; ddata(2,1)/yrange];
        ksi1 = norm(ddata,2); clear ddata;
        if ksi1 < grootte(j)*ksi_grens;
            ddata = data(:,i+1) - punt(:,j+1);
            ddata = [ddata(1,1)/xrange ; ddata(2,1)/yrange];
            ksi2 = norm(ddata,2); clear ddata;
            if ksi2 < grootte(j)*ksi_grens;
                orig(:,k(j),j) = data(:,i) - punt(:,j);
                afb(:,k(j),j) = data(:,i+1) - punt(:,j+1);
                k(j) = k(j)+1; break;
            else ;
            end
        else ;
        end
    end
end
```

C.3.3 Estimating B

```
function B = algorithm_b(serie,pn,A,ksi_grens,grootte,dp_grens);

periode = size(pn,2);

% Looking for points close to the orbits
[orig,afb] = zoekb(serie,pn,grootte,ksi_grens);

% Estimating B
for i = 1:periode;
    temp_afb = afb(:,:,i);
    while temp_afb(:,end) == [0;0];
        temp_afb = temp_afb(:,1:(end-1));
    end
    temp_orig = orig(:,1:size(temp_afb,2),i);
    n = size(temp_orig,2);
    x = temp_orig';
    y = temp_afb';
```

```
    ball = temp_afb - A(:,:,i)*temp_orig;
    nul = zeros(1,n);
    bave = mean(ball,2);
    B(:,:,i) = bave/dp_grens;
end
```

The function uses a function to find relative points:

```
function [orig,afb] = zoekb(serie,p,grootte,ksi_grens);

xrange = max(max(serie(1,:,:)))-min(min(serie(1,:)));
yrange = max(max(serie(2,:,:)))-min(min(serie(2,:)));

punt = [ p , p(:,1) ];
k = ones(size(p,2),1);
for i = 1:(length(serie)-1);
    for j = 1:size(punt,2)-1;
        dserie = serie(:,i,1) - punt(:,j);
        dserie = [dserie(1,1)/xrange ; dserie(2,1)/yrange];
        ksi1 = norm(dserie,2); clear dserie;
        if ksi1 < grootte(j)*ksi_grens;
            dserie = serie(:,i,2) - punt(:,j+1);
            dserie = [dserie(1,1)/xrange ; dserie(2,1)/yrange];
            ksi2 = norm(dserie,2); clear dserie;
            if ksi2 < grootte(j)*ksi_grens;
                orig(:,k(j),j) = serie(:,i,1) - punt(:,j);
                afb(:,k(j),j) = serie(:,i,2) - punt(:,j+1);
                k(j) = k(j)+1; break;
            else ;
            end
        else ;
        end
    end
end
```

C.3.4 Estimation adjustments

```
tol = 1e-4;                % Stopping criterium, absolute tolerance on dp
dpn = [1 ; 1];             % Initial fixed point adjustment
Cold = [1;1];

while max(abs(dpn)) > tol;
% Estimation of A and C
% The function zoeka was already used before
[orig,afb] = zoeka(data,pn,2,ksi_grens);
n = size(orig,2);
x = [ones(1,n) ; orig]';
y = afb';
c = x\y;
A = c(2:3,:)'
C = c(1,:)'
if C == Cold;
    disp('No improvement possible'); break;
end
Cold = C;

% New estimation fixed point
y1 = (afb(1,:)-A(1,:)*orig)';
y2 = (afb(2,:)-A(2,:)*orig)';
yn = [y1;y2];
x1 = [ones(1,n)*(1-A(1,1)) ; -ones(1,n)*A(1,2)]';
x2 = [-ones(1,n)*A(2,1) ; ones(1,n)*(1-A(2,2))]';
xn = [x1;x2];
dpn = xn\yn
if max(abs(dpn)) > tol;
    pn = pn+dpn;
```

```
else pn = pn;
end
end
```

C.4 The OGY algorithm

C.4.1 Fixed point

Here the example of the Hénon map is used.

```
% Parameters
p = a;
dp_grens = 0.05*a;
fp = max(roots([a (1-b) -1]))   % Theoretical fixed point

% Linearize:
A = [-2*a*fp , 1;
    b , 0];
B = [0 ; fp];
lam = eig(A);                   % Eigenvalues of A

% Setting unstable eigenvalues to zero
i = 1;
while i <= length(lam);
    if abs(lam(i)) >= 1
        k(i) = 0; i=i+1;
    else k(i) = lam(i); i=i+1;
    end
end

% Pole placement
C = ctrb(A,B);
if size(B,1) == rank(C)         % Controllability demand
    K = acker(A,B,k);
    K = K';
else error('Matrices A and B are not controllable!')
end

% Application to 1000 iterations
Xfp = [fp;b*fp]
X = [0;0];                      % Initial condition
regelstap = [0];

a = p;
for i = 1:1000;
    dp = -K'*(X(:,i)-Xfp);
    if abs(dp) <= dp_grens
        a = p + dp;
        regelstap = [regelstap , dp];
    else dp = 0; a = p + dp;
        regelstap = [regelstap , dp];
    end
    X(1,i+1) = X(2,i)+1-a*X(1,i)^2;
    X(2,i+1) = b*X(1,i);
end
tijd_X = 0:(length(X)-1);

% Visualization
figure(1)
subplot(2,1,1)
hold on, grid
plot(tijd_X,X(1,:),'b.','MarkerSize',5)
title('System response','FontSize',18)
ylabel('Response x','FontSize',15)
subplot(2,1,2)
```

```
hold on, grid
plot(tijd_X,regelstap,'r.','MarkerSize',5)
xlabel('timestep','FontSize',15),ylabel('Control effort \deltab','FontSize',15);
```

C.4.2 Higher periodic orbits

File to create the matrix $\Phi_{i,j}$:

```
function phi = makephi(A,u,i,j);

periode = size(A,3);

if j > u-1;
    phi = zeros(0,size(A,1));
else
    while i+j > periode;
        i = i-periode;
    end
    phi = A(:,:,i+j);
    while j < u-1;
        j = j+1;
        while i+j > periode;
            i = i-periode;
        end
        phi = A(:,:,i+j) * phi;
    end
end
```

File to create the matrix C:

```
function [C,u] = makec(A,B);

periode = size(A,3);

% Calculation number of stable and unstable eigenvectors
prod = A(:,:,periode);
for i = 1:periode-1;
    prod = prod*A(:,:,periode-i);
end
clear i;
d = eig(prod);
s = 0; u = 0;
for i = 1:length(d);
    if abs(d(i)) < 1;
        s = s+1;
    elseif abs(d(i)) > 1;
        u = u+1;
    else ;
    end
end
clear i prod d;

% Calculation stable eigenvectors for every point
for i = 1:periode;
    stab = A(:,:,i);
    j = i-1;
    for n = 1:periode-1;
        if j == 0;
            j = periode;
        else;
        end
        stab = stab*A(:,:,j);
        j = j-1;
    end
    [v,d] = eig(stab);
    m=1;
```

```
    for n=1:length(d);
        if abs(d(n,n)) < 1;
            vs(:,m,i) = v(:,n);
            m=m+1;
        else;
        end
    end
end
clear i j n m v d stab;

% Creation of C (by using makephi)
for i = 1:periode;
    iu = i+u;
    iumin = i+u-1;
    while iu > periode;
        iu = iu-periode;
    end
    while iumin > periode;
        iumin = iumin-periode;
    end
    temp = [ B(:,:,iumin) , vs(:,:,iu) ];
    if u-1 < 1;
        ;
    else for j = 1:u-1;
            temp = [chaos09phi(A,u,i,u-j) , temp];
        end
    end
    C(:,:,i) = temp;
end
clear i j temp;
```

File to create control matrix K:

```
function K = makek(A,C,u);

periode = size(C,3);
kappa = zeros(1,size(C,1)); kappa(1,1) = 1;
for i = 1:periode;
    temp = kappa*inv(C(:,:,i))*chaos09phi(A,u,i,0);
    K(:,:,i) = temp';
end
clear temp i;
```

OGY control of a period m orbit then becomes as simple as:

```
% Calculation of K
[Cm,um] = makec(Am,Bm);
Km = makek(Am,Cm,um);

% Application of OGY
regel = m;
for i = 1:1000;
    for j = 1:regel;
        dp(j,1) = -Km(:,:,j)'*(X(:,i)-pm(:,j));
        if abs(dp(j,1)) > dp_grens;
            dp(j,1) = 0;
        else ;
        end
    end
    test = find(dp);
    [e,f]=min(abs(dp(test))); dp=dp(test(f));
    if dp == [];
        dp = 0;
    else ;
    end
    a = p + dp;
    regelstap = [regelstap , dp];
```

```
    X(1,i+1) = X(2,i)+1-a*X(1,i)^2;
    X(2,i+1) = b*X(1,i);
    clear test e f;
end
```

C.4.3 Delay coordinates

When OGY is implemented using delay coordinates as in section 5.2.3, K has to be recalculated. Using the same A and new defined matrices B^1 and B^2, the code becomes:

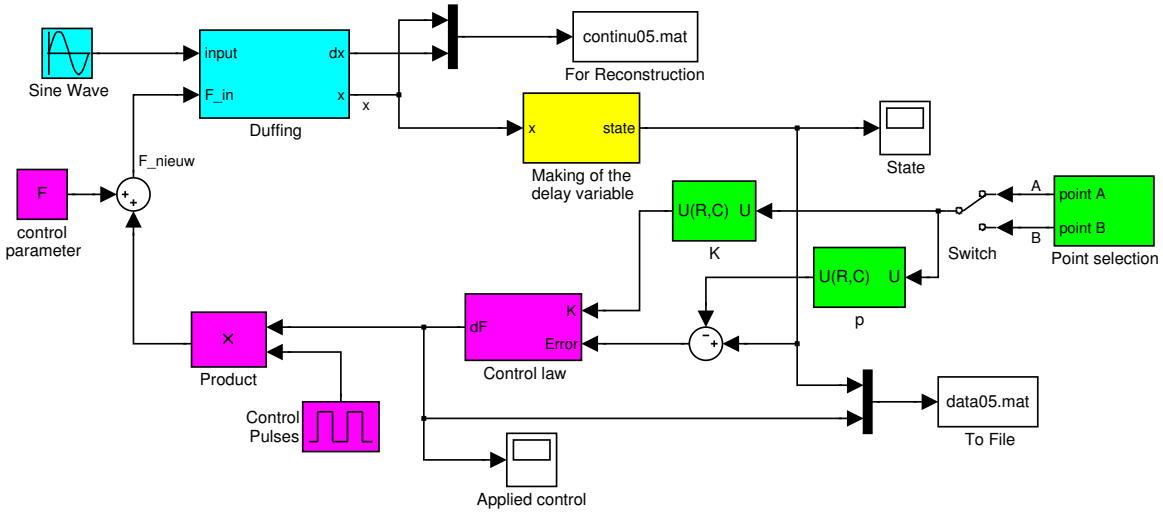
```
% Defining new A and B
An = [A , B_2;
    zeros(1,size(A,2)) , 0];
Bn = [B_1 ; 1];

% Calculation of K
ev = findev(An);
i = 1;
while i <= length(ev);
    if abs(ev(i)) >= 1
        k(i) = 0; i=i+1;
    else k(i) = ev(i); i=i+1;
    end
end
C = ctrb(An,Bn);
if size(Bn,1) == rank(C);
    K = acker(An,Bn,k);
    K = K';
end
```

The application of OGY is then calculated with the following code:

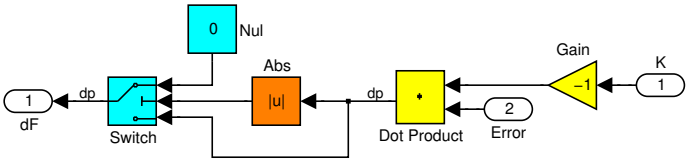
```
% Using fp = fixed point, and K = control matrix
p = F;
regelstap = [0];
period = 2*pi/omega;
eind = 10000;
nop_end = ceil(eind/period);
for i = 1:nop_end;
    tijd = [(i-1)*period , i*period-delay , i*period];
    dp = -K'*(rdata(:,end) - fp);
    if abs(dp) > dp_grens
        dp = 0; F = p + dp;
    else F = p + dp;
    end
    [t,x] = ode45(@duffing,tijd,begin,options,ksi,omega,F);
    regelstap = [regelstap , dp];
    rdata(:,i+1) = [ x(3,1) ; x(2,1) ; F];
    rtijd(1,i+1) = t(end);
    begin = x(end,:);
end
```

Appendix D

Simulink® models

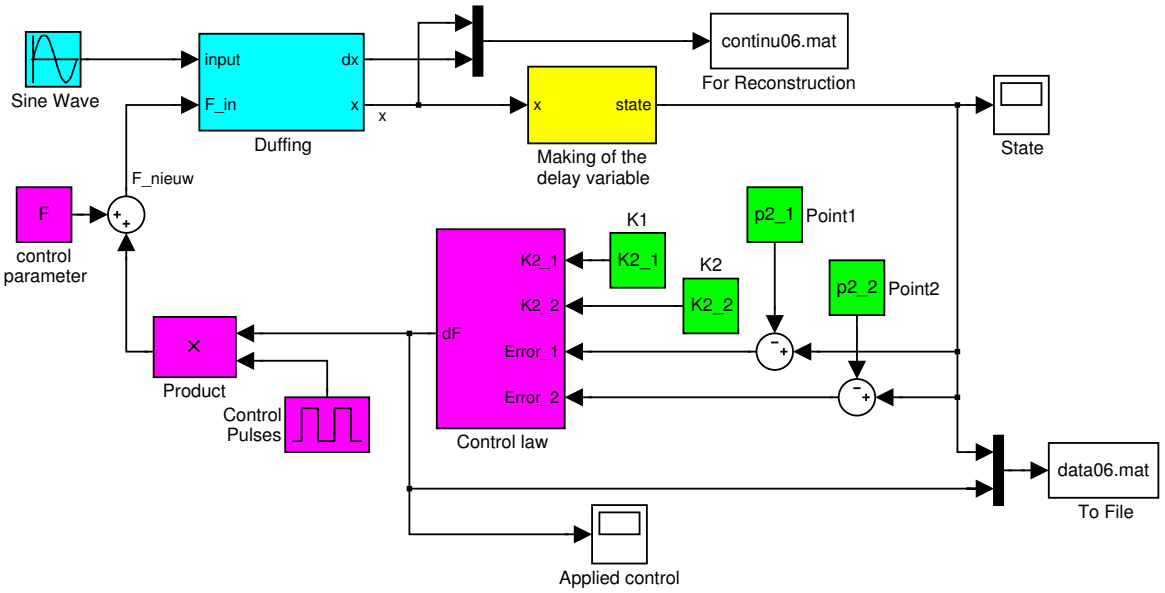


(a) Complete Simulink® model

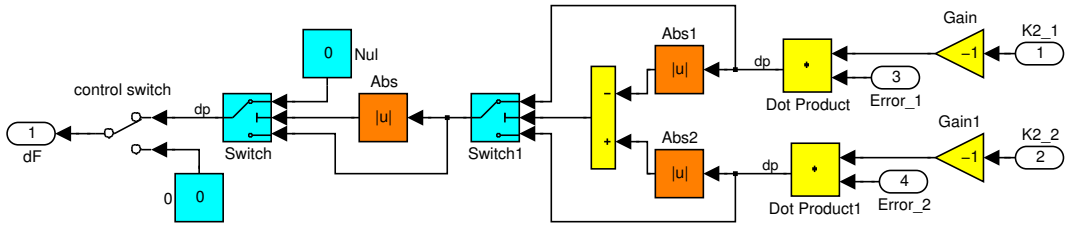


(b) Submodel: OGY control algorithm

Figure D.1: Specific Simulink® models for period 1 orbits

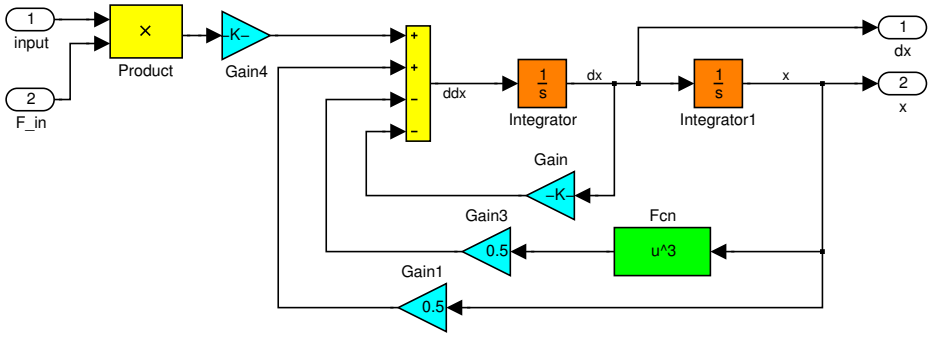


(a) Complete Simulink® model

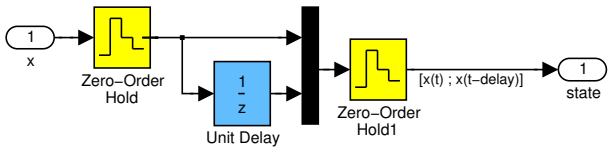


(b) Submodel: OGY control algorithm

Figure D.2: Specific Simulink® models for period 2 orbit



(a) Duffing oscillator



(b) Construction of delay coordinate vector

Figure D.3: Other Simulink® submodels