

# Learning Trajectories of Dynamical Systems

Helle Hein and Ülo Lepik

*University of Tartu*

Liivi 2, Tartu, Estonia

E-mail: [helle.hein@ut.ee](mailto:helle.hein@ut.ee)

Received July 24, 2011; revised May 19, 2012; published online September 1, 2012

**Abstract.** The aim of the present paper is to describe the method that is capable of adjusting the parameters of a dynamical system so that the trajectories gain certain specified properties. Three problems are considered: (i) learning fixed points, (ii) learning to periodic trajectories, (iii) restrictions on the trajectories. An error function, which measures the discrepancy between the actual and desired trajectories is introduced. Numerical results of several examples, which illustrate the efficiency of the method, are presented.

**Keywords:** dynamical systems, learning trajectories, gradient descent method, sensitivity analysis.

**AMS Subject Classification:** 93C15; 68T05; 68Q32.

## 1 Introduction

A goal of dynamical systems theory is to modify the system so that its trajectories get certain properties. There are different possibilities for solving this problem. For instance, systems can be regularized by a special-purpose controller. The solution was proposed in [13] by Ott et al. for controlling chaos and has been developed further in numerous papers. A detailed analysis of the method is beyond the scope of this work.

Another approach to system modification is to adjust the parameters of the system minimizing the error function, which measures the discrepancy between the actual and desired trajectories. In most papers, this is realized using the gradient descent method. For instance, in [2] by Baldi, the parameter adaption method was applied for learning fixed points. Paper [19] by Toomarian and Barhen is dedicated to trajectory learning (two types of trajectories, a circle and figure eight are discussed). Zeng and Singh [20] controlled chaos of the Lorenz system by adapting its parameters. Oysal [14] solved the problem with the aid of dynamic wavelet network. Neural networks were also applied in [18] by Tomasiello for investigating the Duffing equation; the amplitude and frequency of the forcing term were adjusted with the purpose to avoid chaotic behaviour of the system. A novel approach was described by Li et al. in [9] where parameter

identification was accomplished by analogy with a chaotic ant swarm. Different methods for adaptive parameter estimation based on the synchronization of coupled chaotic systems were proposed in [10, 11, 17]. The problems discussed in this paper also could be solved by the optimal control theory, but generally the course of solution is more complicated. This approach is applied in [6]. As an example the forced Duffing equation is solved. For the control function the external force is taken. We have to find such a control which brings the system to a given point in the minimal time. In [6] by means of the optimal control theory it is shown that a solution to this problem exists only if the force is discontinuous (of the “bang–bang” type). Hirasawa et al. [7, 8] applied the Universal Learning networks for chaos control; for training the parameters the Lyapunov exponents were used (the parameters must be adjusted so that the maximum Lyapunov exponent is non-positive). In [1] an efficient method based on the ensemble Kalman filter is proposed. Making use of the Lyapunov functions Parlitz [15] and Chen et al. [3] applied synchronization method to estimate parameters of chaotic systems. Guan et al [5] used the unknown parameter observer method for this purpose.

For adjusting parameters with the aid of gradient descent methods it is expedient to calculate the sensitivity matrix [4, 16].

The conventional gradient descent method has some other shortcomings:

- (i) Usually the error function has several minima and the gradient method may bring us to some local minimum. To find the global minimum some special techniques (e.g. random search method) must be applied.
- (ii) The gradient method gives us a single-valued solution. In the real case the solution may be non-unique, whilst the error function may be zero for some regions of the parameters space.
- (iii) The diagram of the error function may be presented by a non-smooth curve (or surface). In this case the gradient method is not applicable.

By these reasons we suggest an alternative approach. Here the error function is put together via integrating numerically the equations of motion for many combinations of the adjustable parameters. Due to rapid algorithms for solving differential equations this process – in spite of the great number of equations – is not very time-consuming for contemporary computers. An advantage of this “direct method” is also the fact that we do not need to put together the sensitivity matrix, which is a time-consuming process. Practically application of this approach is justified if the number of adjustable parameters is small (one or two).

In this paper second order systems are considered. Here we have two main possibilities: (i) If the system has fixed points then after adjusting the parameters the motion converges to some of the fixed points. (ii) If the system does not have the fixed points then the motion of the adjusted system turns to a periodic motion.

The paper is organized as follows. In Section 2 the problem statement which is based mainly to the Baldi’s paper [2], is given. The problem of learning trajectories to fixed points is discussed in Section 3. In Section 4 it is

demonstrated how nonregular motion may be converted into a periodic one. For solving this problem a new method based on the analysis of the Poincare points is suggested. In Section 6 trajectories with restrictions are discussed. Results of five numerical examples are presented. The gradient approach was used for solving Examples 2 and 5. Examples 1, 3 and 4 were solved by the “direct method”.

## 2 Problem Statement

Consider a dynamical system

$$\frac{du}{dt} = F(t, u, W), \quad t \in [t_0, t_f] \tag{2.1}$$

with the initial condition  $u(t_0) = u_0$ . Here  $u(t)$  is the  $N$ -dimensional row vector, which represents the state variables of the system,  $W$  is the  $J$ -dimensional array of adjustable time-invariant parameters. Let us denote by  $W_0$  the vector of nominal parameters. It is assumed that the system (2.1) follows the prescribed trajectory  $\tilde{u}^*(t)$  over a certain interval. This trajectory is not usually defined for all  $u$  values and over the whole period;  $\tilde{u}^* \in u$  is a  $\Gamma$ -dimensional subset of the prescribed state variables and  $\Gamma \in N$ .

The error function

$$E = \int_{t_0}^{t_f} e[u(t), W, \tilde{u}^*(t)] dt, \tag{2.2}$$

measures the discrepancy between the trajectories  $u(t)$  and the desired values  $\tilde{u}^*(t)$ . For calculating (2.2), local error function  $e$  must be specified; parameter  $W$  obtains the values, where  $E = \min$ . The concrete form of function  $E$  is specified later on.

For calculating the minimum of  $E$ , the gradient learning and back-propagation methods [2] are applied. At first, the gradient

$$\frac{\partial E}{\partial W} \Big|_{W_0} = \int_{t_0}^{t_f} \frac{\partial e}{\partial u} \frac{\partial u}{\partial W} \Big|_{W_0} dt = \int_{t_0}^{t_f} \frac{\partial e}{\partial u} p dt \tag{2.3}$$

is computed. Here  $(p = \partial u / \partial W)_{W_0}$  denotes the trajectory sensitivity matrix [4]; it is  $N \times J$  matrix;  $\partial e / \partial u$  is the  $N$ -dimensional and  $(\partial E / \partial W)_{W_0}$  a  $J$ -dimensional vector. The approximation of the last integral depends on the character of the problem. In the case of learning fixed points (Example 2), the approximation is given in detail. Differentiating matrix  $p$ , we obtain

$$\frac{dp}{dt} = \frac{d\dot{u}}{dW} \Big|_{W_0} = \frac{\partial F}{\partial W} \Big|_{W_0} + \frac{\partial F}{\partial u} \Big|_{W_0} p. \tag{2.4}$$

Introducing matrix

$$L = \frac{\partial F}{\partial u} \Big|_{W_0}, \tag{2.5}$$

formula (2.4) can be put into the form

$$\frac{dp}{dt} = Lp + \left. \frac{\partial F}{\partial W} \right|_{W_0}. \quad (2.6)$$

Equation (2.6) can be integrated using different techniques (consult e.g. [2]). In this paper, we assume that time step  $\Delta t$  is very small and calculate  $dp/dt$  using the following formula:

$$\frac{dp}{dt} \approx \frac{1}{\Delta t} [p(t + \Delta t) - p(t)].$$

Next,  $p(t)$  is calculated from (2.6). New values for  $W$  are found from

$$W_{new} = W_{old} - \alpha \left( \frac{\partial E}{\partial W} \right)_{old}, \quad (2.7)$$

where  $\alpha$  is the learning rate. This procedure is repeated until the desired accuracy is reached. It is important to note that the gradient descent procedure may bring the system to some local minimum.

### 3 Learning Fixed Points

In this section, the case  $\partial F/\partial t = 0$  is considered. It is assumed that the system (2.1) has some fixed point  $u^*$ , for which  $F(u^*, W) = 0$ . From the equality, a few parameters can be evaluated. The remained parameters are to be learned so that the system would converge to value  $u^*$ .

If we denote  $u_f = u(t_f)$ , the error function  $e$  is defined as

$$e = 0.5(u_f - u^*)^2. \quad (3.1)$$

In the present case, function  $E$  takes the form of  $E = e(t_f - t_0)$ ; quantity  $p = \partial u_f / \partial W$  does not depend on time or  $\dot{p} = 0$ . From (2.6), we get that

$$p = -L^{-1} \left. \frac{\partial F}{\partial W} \right|_{W_0}, \quad (3.2)$$

where  $L^{-1}$  denotes the inverse matrix of  $L$ . According to (2.3), we obtain

$$\left. \frac{\partial e}{\partial W} \right|_{W_0} = (u_f - u^*)p. \quad (3.3)$$

Computer simulations were carried out for the following two examples.

*Example 1.* Consider the Lorenz system:

$$\begin{aligned} \dot{x} &= -\sigma(x - y), \\ \dot{y} &= rx - y - xz, \\ \dot{z} &= xy - bz. \end{aligned} \quad (3.4)$$

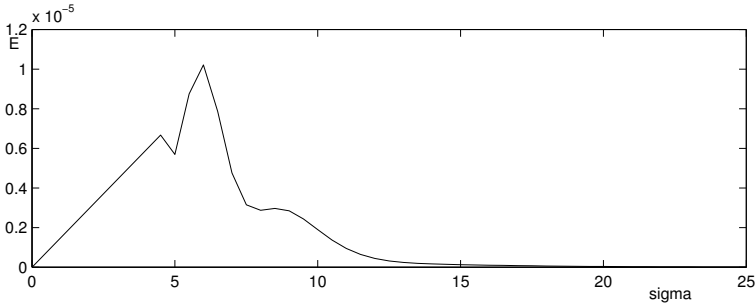
Here,  $u_1 = x$ ,  $u_2 = y$ ,  $u_3 = z$  are state variables and parameters  $\sigma$ ,  $r$ ,  $b$  have real positive values.

Let  $x_*, y_*, z_*$  denote the coordinates of the desired fixed point. According to (3.4), the condition  $\dot{x} = 0$  is satisfied if (i)  $x_* = y_*$  or (ii)  $\sigma = 0$  and  $x_* \neq y_*$ . Since  $\sigma$  is an essential parameter of the system (3.4) (the Prandtl number), we leave the second case as it is and assume that  $x_* = y_*$ . Fulfilling conditions  $\dot{y} = \dot{z} = 0$  for the fixed point, we find that  $r = 1 + z_*$ ,  $b = x_*^2/z_*$ .

We see that parameters  $r, b$  are prescribed and we can adjust only parameter  $\sigma$ . Hence, the error function (3.1) obtains the following form

$$e(\sigma) = \frac{1}{2} [(x_f(\sigma) - x_*)^2 + (y_f(\sigma) - y_*)^2 + (z_f(\sigma) - z_*)^2]. \tag{3.5}$$

Since there is only one parameter to be adjusted, it is simpler to calculate function  $e(\sigma)$  directly without calculation of matrix  $p$ .



**Figure 1.** Lorenz system; error estimate (3.5) as a function of parameter  $\sigma$ .

The computer simulations were carried out for  $\sigma = 10, r = 28, b = 8/3, x(0) = y(0) = z(0) = 2$ . This is a chaotic case. For the desired fixed point, we take  $x_* = y_* = 3, z_* = 5$ . The system was integrated by the fourth order Runge–Kutta method (with the precision  $10^{-7}$ ). Error function  $e(\sigma)$  is plotted in Fig. 1. As it follows from the diagram, the trajectories practically converge to the desired point if  $\sigma > 15$  and  $E(15) = 1.11 \times 10^{-5}$ . Simulation results of time history for  $\sigma = 15$  are shown in Fig. 2.

*Example 2.* Consider the modified Duffing equation:

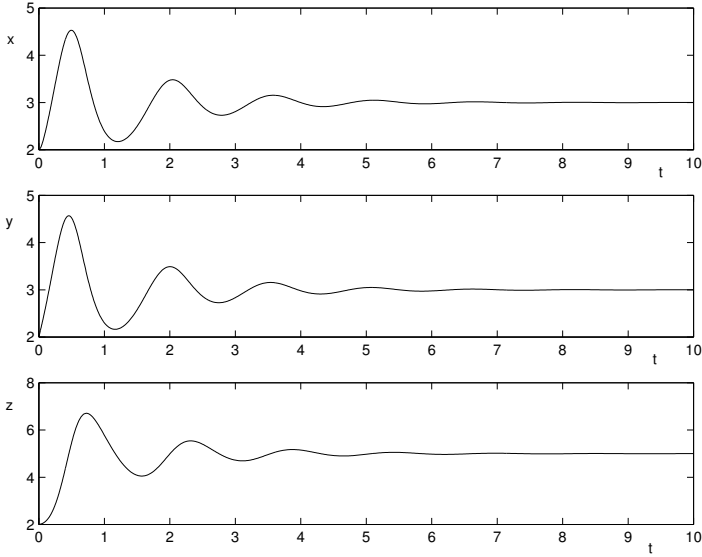
$$\begin{aligned} \dot{x} &= y, \\ \dot{y} &= -ay - bx - cx^3 - dx^5. \end{aligned} \tag{3.6}$$

In the fixed point  $\dot{x} = \dot{y} = 0$  and according to (3.6), we have  $y_* = 0$ . Parameter vector  $W$  contains four elements  $W = (a, b, c, d)$ . According to (2.5), matrix  $L$  takes the following form:

$$L = \left( \frac{\partial F}{\partial u} \right) \Big|_{t=t_f} = \begin{pmatrix} 0 & 1, \\ -\Delta & -a \end{pmatrix}, \quad \Delta = b + 3cx_f^2 + 5dx_f^4.$$

Next, we calculate matrix  $\frac{\partial F}{\partial W}$ :

$$\frac{\partial F}{\partial W} \Big|_{t=t_f} = \begin{pmatrix} 0 & 0 & 0 & 0, \\ -y_f & -x_f & -x_f^3 & -x_f^5 \end{pmatrix}.$$



**Figure 2.** Time response of the Lorenz system for  $\sigma = 15$ .

Sensitivity matrix  $p$  is calculated according to (3.2):

$$p = -\frac{1}{\Delta} \begin{pmatrix} y_f & x_f & x_f^3 & x_f^5 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

The gradient approximation matrix is approximated with the aid of (2.3) and (3.3):

$$\left. \frac{\partial E}{\partial W} \right|_{t=t_f} = -\frac{(t_f - t_0) * (x_f - x_*)}{\Delta} (y_f, x_f, x_f^3, x_f^5). \tag{3.7}$$

The parameter adjustment is performed according to the following algorithm:

Step 1: Assume vector of initial parameters  $W_0 = (a_0, b_0, c_0, d_0)$ , time interval is  $(t_0, t_f)$  and learning rate  $\alpha$  are the initial conditions for the system.

Step 2: Integrate the system (3.6), calculate the error function (3.1) and get quantities  $x_f, y_f$ .

Step 3: Calculate  $\frac{\partial E}{\partial W}$  according to (3.7) and adjust the parameters according to (2.7).

Step 4: Repeat steps 2 and 3 until the error  $E$  satisfies the prescribed precision.

The computer simulation was carried out for  $a_0 = -0.1, b_0 = 4.0, c_0 = -5.0, d_0 = 1.0, x(0) = 0, y(0) = 1$ . Values  $x_* = 2, y_* = 0$  were taken for the desired fixed point. The learning rate was equal to 0.2 and the error precision  $10^{-6}$ , respectively. The parameter values after the gradient descent learning were  $a = 4.8660, b = -4.6069, c = -14.1378$  and  $d = 3.8262$ . The time response of the system for the values is demonstrated in Fig. 3.

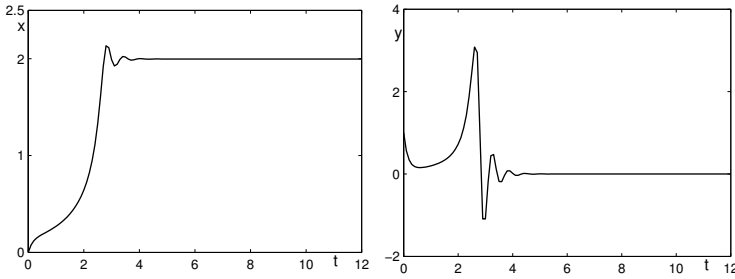


Figure 3. Time response of the Duffing system after gradient learning.

### 4 Converting the Motion into Periodic Trajectories by Adjusting the Parameters

The trajectories of the system (2.1) are generally non-periodic. Our aim is to adjust parameters  $W$  so that the trajectory is periodic or close enough to a periodic trajectory. To solve the problem, a method which is based on the Poincare sections is proposed. A Poincare section is obtained when an  $N$ -dimensional trajectory is cut by several  $\mu$ -dimensional hyperplanes where  $\mu < N$ . Due to the section, a set of points in the hyperplane is derived. We shall call these points the Poincare points. For a periodic motion, the set consists of a single point. Hence, in order to make the trajectory periodic, the dispersion of the Poincare points should be minimized. This can be done in the following way.

Let us denote the coordinates of the Poincare points by  $U_{lm}$ , where  $1 \leq l \leq K$ ,  $1 \leq m \leq \mu$  (index  $l$  denotes the number of the point,  $m$  – the number of the coordinate;  $K$  is the total number of the Poincare points). It is reasonable to give some time to the system for adaptation and leave the first  $\nu$  points out. Therefore, the mean and dispersion of the Poincare points are

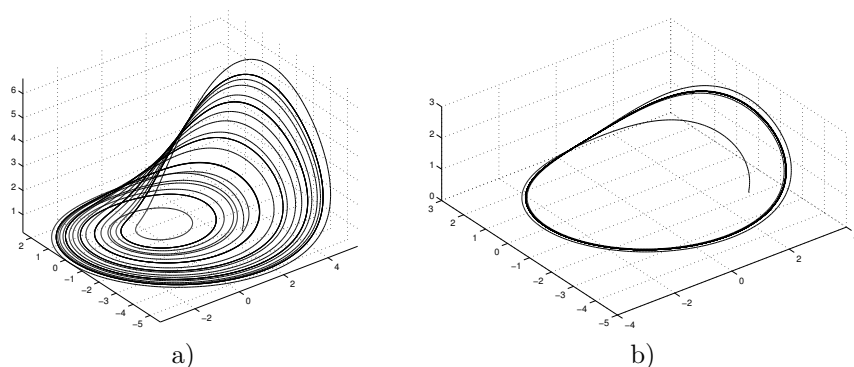
$$M_m = \frac{1}{K - \nu} \sum_{l=\nu}^K U_{lm}, \quad D = \frac{1}{K - \nu - 1} \sum_{l=\nu}^K \sum_{m=1}^{\mu} (U_{lm} - M_m)^2.$$

Dispersion  $D$  can be interpreted as an error function. Hereof, the values of the parameters should be found where  $D = \min$ . If the minimal value of  $D$  is not close enough to zero, it is not possible to find periodic trajectories to the given problem.

*Example 3.* Let us examine the Roessler system:

$$\begin{aligned} \dot{x} &= -y - z, \\ \dot{y} &= x + ay, \quad t \in [0, 200], \\ \dot{z} &= b + xz - cz \end{aligned}$$

for  $x(0) = 2.5$ ,  $y(0) = -2.5$ ,  $z(0) = 0.5$ ,  $a = 0.4$ ,  $b = 2$  and  $c = 4$ . The three-dimensional phase diagram of the Lorenz system for the current case is plotted in Fig. 4a. The motion is clearly non-periodic.



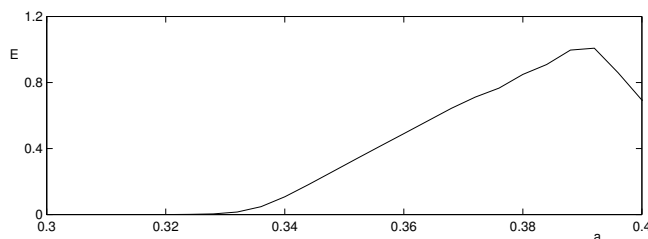
**Figure 4.** Results for the Rössler system: a) the three-dimensional phase diagram for  $a = 0.4$ , b) adapted Rössler system  $a = 0.25$ .

We assume that parameters  $b$  and  $c$  are prescribed and only  $a$  is an adjustable parameter. The Poincaré section is carried out with plane  $x = x(0)$ . The coordinates of Poincaré points are denoted  $Y(j)$ ,  $Z(j)$ ,  $j = 1, 2, \dots, K$ . The computer simulation was executed for  $\nu = 10$ . The system was integrated by the Runge–Kutta method for different values of parameter  $a$ . The dispersion  $D(a)$  was calculated from the formulae

$$Y_m = \frac{1}{K - \nu} \sum_{l=\nu}^K Y(l), \quad Z_m = \frac{1}{K - \nu} \sum_{l=\nu}^K Z(l),$$

$$D = \frac{1}{K - \nu - 1} \sum_{l=\nu}^K \sum_{m=1}^{\mu} [(Y(l) - Y_m)^2 + (Z(l) - Z_m)^2]. \quad (4.1)$$

According to the computations, the number of the Poincaré points was  $K = 32$ . The diagram  $D = D(a)$  is plotted in Fig. 5.



**Figure 5.** Error function  $D(a)$  for the Rössler system.

As it follows from Fig. 5, the dispersion for  $a < 0.3$  is very small:  $D(0.25) = 4.92 \times 10^{-9}$ ,  $D(0.3) = 1.83 \times 10^{-7}$ , but  $D(0.4) = 0.83$ . Therefore, for  $a < 0.3$  the system can be regarded as periodic. This assertion is confirmed by Fig. 4b, where the three-dimensional phase diagram of the modified system is presented.



Example 4. Consider the forced Duffing equation:

$$\begin{aligned} \dot{x} &= y, \\ \dot{y} &= -ay - bx - cx^3 + f \cos t, \quad t \in [0, 300] \end{aligned} \tag{4.2}$$

for  $x(0) = 0$  and  $y(0) = 1$ . For  $a = 0.25$ ,  $b = -1$ ,  $c = 1$ ,  $f = 0.3$ , the motion is chaotic (Fig. 6). The upper and lower diagrams present the time history and the phase diagram, respectively. Our aim is to adjust parameter  $a$  so that the motion becomes periodic.

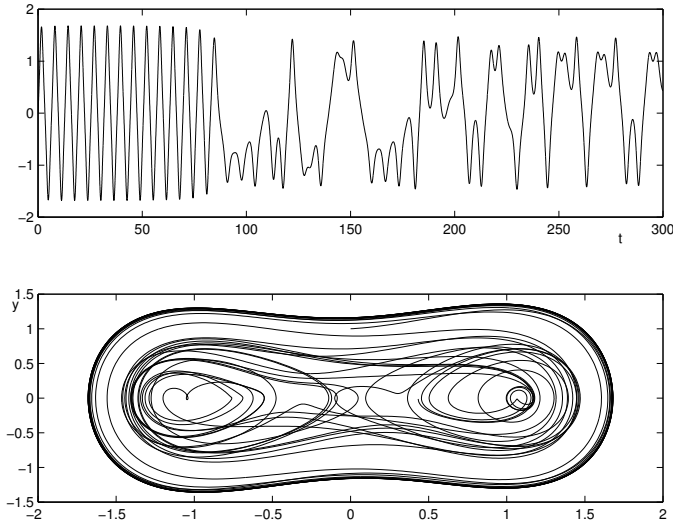


Figure 6. Chaotic motion of the Duffing system for  $a = 0.25$ .

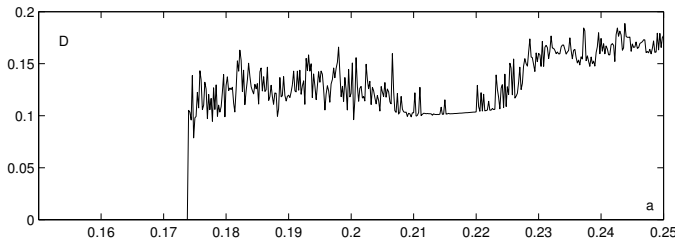
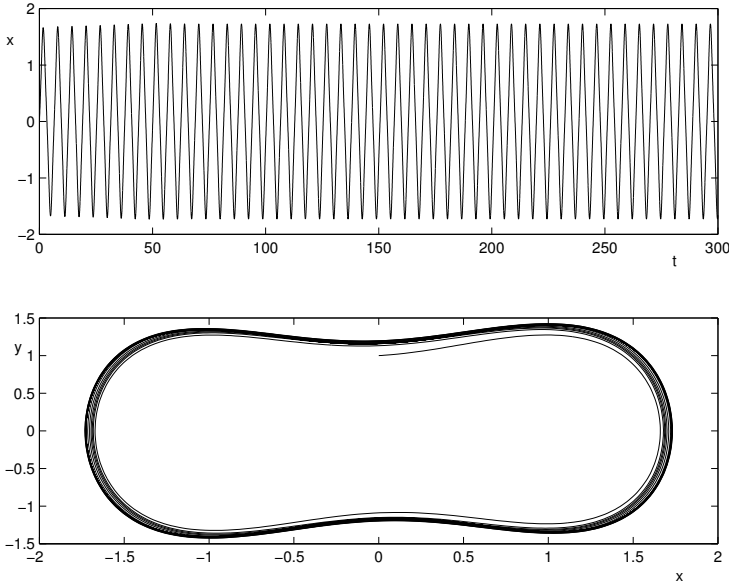


Figure 7. The Duffing system; error estimate (4.1) as a function of parameter  $a$ .

Let us consider the Poincare section  $x = 0$ . The formulae (4.1) holds if we replace  $Z(l) = Z_m = 0$ . Computations were carried out for  $\nu = 5$ ; function  $D(a)$  is plotted in Fig. 7. According to the diagram,  $D \approx 0$  if  $a < 0.17$ . The computer simulation showed that  $D(0.17) = 5.91 \times 10^{-7}$ ; however,  $D(0.174) = 0.115$ ; in the latter case, the motion is chaotic. The results for the



**Figure 8.** Adapted Duffing system ( $a = 0.17$ ).

adapted value  $a = 0.17$  are plotted in Fig. 8 (time history and phase diagram). The local minimum exists at  $a = 0.213$ , where  $D = 3.24 \times 10^{-10}$ . The area for the minimum is quite restricted since  $D(0.211) = 0.030$  and  $D(0.215) = 0.008$ .

In Fig. 7, curve  $D(a)$  has a very complicated form (several local minima). Therefore, it can be assumed that the gradient descent method does not work well in the current case.

### 5 Restrictions for the Trajectories

Let us discuss the case where the trajectories of the problem (2.1) are closed between two limit curves

$$u_{\gamma}^{-}(t) \leq u_{\gamma}(t) \leq u_{\gamma}^{+}(t), \quad t \in [t_0, t_f]. \tag{5.1}$$

Here,  $\gamma$  denotes the number of the state variable  $u$  for which the restriction (5.1) is imposed. If  $u_{\gamma}^{-}(t) = u_{\gamma}^{+}(t)$ , we get the case where the system (2.1) follows the prescribed trajectory over the time interval; the case was discussed in [10]. We perturb the parameters  $W$  until inequalities (5.1) are satisfied. For that purpose, function  $\varphi_{\gamma}$  is introduced

$$\varphi_{\gamma}(t) = \begin{cases} u_{\gamma}(t) - u_{\gamma}^{+}(t), & \text{if } u_{\gamma}(t) > u_{\gamma}^{+}(t), \\ u_{\gamma}(t) - u_{\gamma}^{-}(t), & \text{if } u_{\gamma}(t) < u_{\gamma}^{-}(t), \\ 0, & \text{elsewhere.} \end{cases} \tag{5.2}$$

The error function is taken in the form of

$$E = \int_{t_0}^{t_f} e(t) dt = \frac{1}{2} \sum_{\gamma} \int_{t_0}^{t_f} \varphi_{\gamma}^2(t) dt.$$

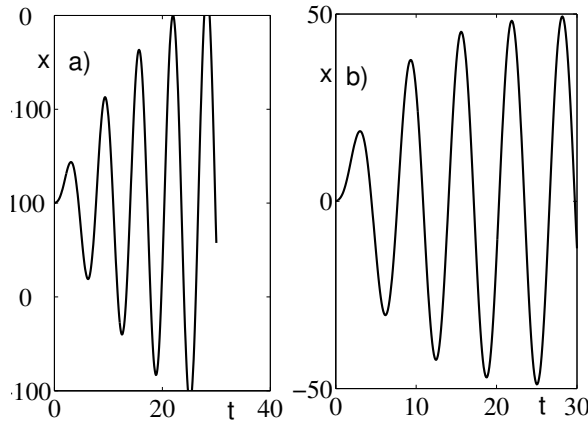
The summation is carried out only for such values  $\gamma$  that occur in the inequalities (5.1). Since

$$\frac{\partial e}{\partial u_{\gamma}} = \gamma_{\gamma} \frac{\partial \varphi_{\gamma}}{\partial u_{\gamma}} = \varphi_{\gamma},$$

Equation (2.3) obtains the form of

$$\frac{\partial E}{\partial W} = \int_{t_0}^{t_f} \varphi_{\gamma} p dt, \tag{5.3}$$

where sensitivity matrix  $p$  is calculated according to (2.6).



**Figure 9.** Time response of the Duffing system in the case of restrictions (5.4); (a) before learning, (b) after learning.

*Example 5.* Let us solve (4.2) for  $c = 0$  and  $x(0) = 0, y(0) = 1, t \in [0, 30]$ . The restrictions (5.2) are taken in the form

$$\varphi_1(t) = \begin{cases} x(t) - 50, & \text{if } x > 50, \\ x(t) + 50, & \text{if } x < -50, \\ 0, & \text{elsewhere.} \end{cases} \tag{5.4}$$

The solution for  $a = 0.1, b = 1, f = 15$  is plotted in Fig. 9(a). Since the restrictions (5.4) are violated, we must adjust these parameters.

According to (5.3), we have

$$\frac{\partial E}{\partial a} = \int_{t_0}^{t_f} \varphi(t) p_{11}(t) dt, \quad \frac{\partial E}{\partial b} = \int_{t_0}^{t_f} \varphi(t) p_{12}(t) dt, \quad \frac{\partial E}{\partial f} = \int_{t_0}^{t_f} \varphi(t) p_{13}(t) dt. \tag{5.5}$$

For evaluation integrals (5.3) and (5.5), the Simpson's rule was applied. A computer simulation gave the following values for the adjusted parameter:  $a = 0.298$ ,  $b = 1.030$ ,  $f = 15.00$ . The time history after learning is presented in Fig. 9(b).

## 6 Discussion

In order to obtain the trajectories with certain properties, the parameters of a dynamic system should be assigned such values that would minimize the local error  $e$  or the integral error  $E$ . This can be done by the traditional gradient descent method. The approach was realized in Examples 2 and 5. It should be mentioned that the approach has some shortcomings. First, we must put together and integrate the sensitivity matrix which is a time-consuming procedure. Secondly, the gradient descent algorithm gives us a single-valued solution, whilst in real cases the error function can be zero for some regions of the parameters to be adjusted (see Figs. 1, 5, 7). Thirdly, the error function  $E$  may have many local minima which are characterized by non-smooth curve (consult Fig. 7). In such a case, the gradient descent is not applicable.

If the number of the parameters to be adjusted is small (one or two), the gradient method should be set aside; it is suggested to evaluate the error function directly. Here, we integrated the system of dynamical equations for many combinations of the adjustable parameters. In this way, the error function was composed. Since nowadays many fast subprograms exist based on the Runge-Kutta method, the computing time for the operation is relatively small. Using one of these, Examples 1, 3, 4 were solved: the intervals of the parameters, for which the desired solution holds, were calculated (consult Figs. 1, 5, 7). Finally, a few different algorithms and computer programs for defining periodic orbits can be found in [12], but as a rule they require significantly more computing time.

## Acknowledgement

Financial support from the Estonian Science Foundation under Grant ETF 8830 is gratefully acknowledged.

## References

- [1] J.D. Annan and J.C. Hargreaves. Efficient parameter estimation for a highly chaotic system. *Tellus*, **56**:520–526, 2004. <http://dx.doi.org/10.1111/j.1600-0870.2004.00073.x>.
- [2] B. Baldi. Gradient descent learning algorithm overview: a general dynamic system perspective. *IEEE Trans. Neural Networks*, **6**:182–195, 1995. <http://dx.doi.org/10.1109/72.363438>.
- [3] S.H. Chen, J. Hu, C.H. Wang and J.H. Lu. Adaptive synchronization of uncertain Roessler hyperchaotic system based on parameter identification. *Phys. Lett. A*, **321**:50–55, 2004. <http://dx.doi.org/10.1016/j.physleta.2003.12.011>.

- [4] P.M. Frank. *Introduction to System Sensitivity Theory*. Academic Press, 1978.
- [5] X.P. Guan, H.P. Peng, L.X. Li and Y.Q. Wang. Parameter identification and control of Lorenz system. *Acta Phys. Sinica*, **50**:26–29, 2001.
- [6] H. Hein and U. Lepik. Regularization of the trajectories of dynamical systems by adjusting parameters. *World Academy of Science, Engineering and Technology*, **43**:318–321, 2008.
- [7] K. Hirasawa, X. Wang, J. Murata and J. Hu. Chaos control using maximum Lyapunov number of universal learning network. In *Proceedings of IEEE International Conference on Systems, Man, Cybernetics*, pp. 1702–1707, San Diego, 1998.
- [8] K. Hirasawa, X. Wang, J. Murata, J. Hu and C. Jin. Universal learning network and its application to chaos control. *Neural Networks*, **13**:239–253, 2000. [http://dx.doi.org/10.1016/S0893-6080\(99\)00100-8](http://dx.doi.org/10.1016/S0893-6080(99)00100-8).
- [9] L. Li, Y. Yang, H. Peng and X. Wang. Parameters identification of chaotic systems via chaotic ant swarm. *Chaos Solitons Fractals*, **28**:1204–1211, 2006. <http://dx.doi.org/10.1016/j.chaos.2005.04.110>.
- [10] I.P. Marino and J. Miguez. An approximate gradient descent method for joint parameter estimation and synchronization of coupled chaotic systems. *Phys. Lett.*, **351**:262–267, 2006. <http://dx.doi.org/10.1016/j.physleta.2005.11.005>.
- [11] I.P. Marino and J. Miguez. Monte Carlo method for multiparameter estimation in coupled chaotic systems. *Phys. Rev. E*, **76**:057203, 2007. <http://dx.doi.org/10.1103/PhysRevE.76.057203>.
- [12] H.E. Nusse and J.A. Yorke. *Dynamics: Numerical Explorations*. Springer, 1998.
- [13] E. Ott, C. Grebogi and J.A. Yorke. Controlling chaos. *Phys. Rev. Lett.*, **64**:1196–1199, 1990. <http://dx.doi.org/10.1103/PhysRevLett.64.1196>.
- [14] Y. Oysal. An intelligent control of chaos in Lorenz system with a dynamic wavelet network. *Lecture Notes in Comput. Sci.*, **3215**:581–587, 2004. [http://dx.doi.org/10.1007/978-3-540-30134-9\\_78](http://dx.doi.org/10.1007/978-3-540-30134-9_78).
- [15] U. Parlitz. Estimating model parameters from time series by autosynchronization. *Phys. Rev. Lett.*, **76**:1232–1235, 1996. <http://dx.doi.org/10.1103/PhysRevLett.76.1232>.
- [16] E. Rozenwasser and R. Yusupov. *Sensitivity of Automatic Control Systems*. CRC Press, 1999.
- [17] H. Sakaguchi. Parameter evaluation from time sequences using chaos synchronization. *Phys. Rev. E*, **65**:027201, 2002. <http://dx.doi.org/10.1103/PhysRevE.65.027201>.
- [18] S. Tomasiello. An application of neural networks to a non-linear dynamics problem. *J. Sound Vibration*, **272**:461–467, 2004. <http://dx.doi.org/10.1016/j.jsv.2003.06.016>.
- [19] N.B. Toomarian and J. Barhen. Learning a trajectory using adjoint functions and teacher forcing. *Neural Networks*, **5**:473–484, 1992. [http://dx.doi.org/10.1016/0893-6080\(92\)90009-8](http://dx.doi.org/10.1016/0893-6080(92)90009-8).
- [20] Y. Zeng and S.N. Singh. Adaptive control of chaos in Lorenz system. *Dynamics and Control*, **7**:143–154, 1997. <http://dx.doi.org/10.1023/A:1008275800168>.