Chapter 2
Cell Mapping Techniques for Tuning Dynamical Systems

Ángela Castillo and Pedro J. Zufiria

Abstract In this chapter, to be dedicated in the 90th birthday of Professor C. S. Hsu, several computational schemes are presented for the optimal tuning of the global behavior of nonlinear dynamical systems. Specifically, the maximization of the size of domains of attraction associated with invariants in parametrized dynamical systems is addressed. Cell Mapping (CM) techniques are used to estimate the size of the domains for different parameter values, and such size function is then maximized via several optimization methods. First, a genetic algorithm is tested whose performance shows to be good for determining global maxima at the expense of high computational cost. Secondly, an iterative scheme based on a Stochastic Approximation procedure (the Kiefer–Wolfowitz algorithm) is evaluated showing acceptable performance at low cost. Finally, several schemes combining neural network based estimations and model-based optimization procedures are addressed with promising results. The performance of the methods is illustrated with some applications including the well-known van der Pol equation with standard parametrization, and the tuning of a controller for saturated systems.

2.1 Introduction

Dynamical systems with tunable parameters are very common in many branches of science and engineering. For instance, dynamic and control systems (Brockett and Li 2003; Hu et al. 2002, 2005; Lewis 1987), robotic architectures (Arkin 1998; Fu et al. 1987), and learning schemes (Mitchell 1997; Moore and Naidu 1983) are processes whose behavior depends on the actual value of some characterizing parameters. In other words, the properties of dynamical systems are parameter

Á. Castillo (✉) • P.J. Zufiria
Universidad Politécnica de Madrid, Madrid, Spain
e-mail: angela.castillo@upm.es; pedro.zufiria@upm.es

J.-Q. Sun and A.C.J. Luo (eds.), *Global Analysis of Nonlinear Dynamics*,
DOI 10.1007/978-1-4614-3128-2_2,

dependent. The number equilibria and their stability in parametrized dynamical systems are widely studied in the framework of *bifurcation theory* (Seydel 1988).

The study and design of nonlinear dynamical systems based on global properties (such as the size of attraction domains) cannot be easily addressed by standard procedures. The study of attraction domains associated with asymptotically stable system invariants is fundamental for global analysis. Some approaches for computing such domains of attraction can be seen in Flashner and Guttalu (1988), Guttalu and Flashner (1988) and Xu et al. (1985). The tuning of these domains has great applicability in the design of controllers (Hu et al. 2002), neural network models (Cohen 1992), and the improvement of convergence in learning schemes (Moore and Naidu 1983).

In this chapter, several computational schemes are considered for maximizing the size of domains of attraction in certain parametrized dynamical systems. Cell mapping (CM) techniques (Hsu 1987) are employed to estimate the size of those regions, as presented in Castillo and Zufiria (2000, 2002, 2011). A genetic algorithm, a Kiefer–Wolfowitz stochastic approximation procedure and several neural network-based schemes are proposed as optimization methods for determining the optimal parameter values.

Cell mapping-based schemes have also been employed for the design and evaluation of optimal controllers (Hsu 1985; Hu et al. 1994a,b; Papa et al. 1997; Martínez-Marín and Zufiria 1999; Zufiria and Martínez-Marín 2003), but only in cases where the cost functional can be approximated and optimized via local procedures. In this chapter, a cell mapping-based global performance optimal control is presented.

The rest of this chapter is organized as follows. The following section presents the problem statement as well as the proposed procedure to address the problem. Section 2.3 details some modifications performed on the cell mapping technique for delineating and measuring the domains of attraction. The computational schemes for finding optimal parameter values are detailed in Sect. 2.4. Simulation examples are presented in Sect. 2.5. Concluding remarks appear in Sect. 2.6.

2.2 Problem Statement and Proposed Approach

We consider a family of dynamical systems defined by

$$\dot{x} = F(x, a), \quad x \in \mathbb{R}^n, \ a \in \mathcal{I}, \tag{2.1}$$

where $\mathcal{I}$ is a compact subset of $\mathbb{R}^p$. In addition, we consider a global performance index $J(a) \in \mathbb{R}$, which naturally will also depend on the system parameter vector $a \in \mathcal{I}$.

A parameter value $\bar{a} \in \mathcal{I}$ is to be found such that the performance index is maximized:

$$J(\bar{a}) = \max\{J(a), a \in \mathcal{I}\}. \tag{2.2}$$

It is well known that the existence of $\bar{a}$ is guaranteed, for instance, whenever $J(a)$ is a continuous function on the compact set $\mathcal{I}$. Also, if $J(a)$ is bounded (without necessarily being continuous) in $\mathcal{I}$, then the supremum $\sup\{J(a),\, a \in \mathcal{I}\}$ exists. Note that, although this supremum may not be reached for any value of a, numerical optimization schemes may provide values of a such that $J(a)$ is close enough to the supremum. In the rest of the paper, without loss of generality, we will talk about finding the maximum of J.

The performance index J may be difficult to compute since it is supposed to gather global features. Hence, the use of numerical approximations becomes relevant. First, one has to approximate (estimate) $J(a)$ for any given value of a. Second, J must be optimized, meaning that the maximum (or supremum) of $\{J(a),\, a \in \mathcal{I}\}$ must be determined, using such approximated (estimated) values, without the availability of any explicit algebraic expression for J. This condition is very restrictive and will determine the optimization algorithms that can be employed.

In this chapter, several algorithms are proposed which implement these two steps as follows:

- $J(a)$ is estimated making use of a cell mapping technique:

$$J(a) \approx J^{CM}(a), \qquad a \in \mathcal{I}. \tag{2.3}$$

  This approximation is characterized in the following section.
- An approximation of $\max\{J^{\mathrm{CM}}(a),\, a \in \mathcal{I}\}$ is computed using optimization methods which are appropriate for dealing with the mentioned restriction (non-availability of an algebraic expression). Precisely, Genetic Algorithms, a Kiefer–Wolfowitz scheme, and some Neural Network-based techniques have been employed.

2.2.1 Size of Domain of Attraction

The size of a domain of attraction is considered as the performance index J to be maximized. Let us consider that $\forall a \in \mathcal{I}$, system (2.1) has an attractor at the equilibrium point x^*, meaning that x^* will have an associated attraction domain $D(x^*, a)$ which will depend on the parameter value a.

From a practical point of view, a working region $\mathcal{H}$ is defined, usually being an n-dimensional rectangle $(\mathcal{H} = [a_1, b_1] \times [a_2, b_2] \times \ldots \times [a_n, b_n])$ which contains x^*. Hence, the portion of the domain of attraction included in such region $D_{\mathcal{H}} = D(x^*, a) \cap \mathcal{H}$ is to be considered. (Note that if $D(x^*, a) \subset \mathcal{H}$ then $D_{\mathcal{H}}(x^*, a) = D(x^*, a)$).

A value $\bar{a} \in \mathcal{I}$ of the parameter is to be found such that $vol(D_{\mathcal{H}}(x^*, \bar{a})) = \max\{vol(D_{\mathcal{H}}(x^*, a)), a \in \mathcal{I}\}$. In this expression, $vol(\cdot)$ is the mathematical function which defines the volume of a region, so that the performance index can be defined as

$$J(a) = vol(D_{\mathcal{H}}(x^*, a)), \quad a \in \mathcal{I}. \tag{2.4}$$

As mentioned above, the existence of $\bar{a}$ in the compact set $\mathcal{I}$ is guaranteed if $J(a)$ is a continuous function.

The definition of $J(a) = vol(D_{\mathcal{H}}(x^*, a)) = \lambda(D_{\mathcal{H}}(x^*, a))$ is based on measure theoretic concepts. Precisely, since $D(x^*, a)$ is an open set and provided $\mathcal{H}$ is compact, then $D_{\mathcal{H}}(x^*, a)$ is Lebesgue measurable. Nevertheless, we cannot guarantee $J(a)$ to be a continuous function, even if $F \in C^1(\mathbb{R}^n \times \mathcal{I})$. Fortunately, since $\mathcal{H}$ is bounded, $\sup\{J(a),\ a \in \mathcal{I}\}$ does exist.

The procedure proposed for the maximization of this specific performance index J can be summarized as follows:

(a) First, the volume of $D_{\mathcal{H}}(x^*, a)$ is estimated. This estimation can be performed using the cell mapping technique (CM) (Hsu 1987), a computational method for the global analysis of nonlinear dynamical systems. The use of CM requires the selection of a rectangle in the state space region under consideration. This rectangle does eventually define our working region $\mathcal{H}$. Hence, the global analysis will be restricted to such region. This rectangle is divided into cells, its complementary set being called *sink cell*. Based on that division of the state space, CM can be applied to determine equilibria, limit cycles, and periodic solutions located within the prescribed rectangle. Furthermore, if some attractor is found, CM provides an approximation to its attraction domain in the rectangle (it will be denoted $D^{\mathrm{CM}}(x^*,\ a)$), through the so-called *cellular attraction domain*.

Once CM has been applied, the computation of the volume of a cellular domain is not difficult because it consists of cells, and cell dimensions are known values. When referred to the problem considered here, CM can be applied on $\mathcal{H}$, and if x^* is located, the volume of its associated cellular domain will be considered as an estimation of the volume of $D_{\mathcal{H}}(x^*, a)$.

Hence, the function to be maximized is the unknown

$$J(a) = vol(D_{\mathcal{H}}(x^*, a)), \tag{2.5}$$

and the function actually being optimized is its estimation

$$J^{CM}(a) = vol(D^{CM}(x^*, a)), \tag{2.6}$$

which has no explicit algebraic expression.

(b) An optimization method to maximize J^{CM} is required. For that purpose, an algorithm able to maximize functions without explicit algebraic expression is required. Three types of schemes are considered below.

2.2.2 Characterization of J^{CM}

As mentioned above $J(a)$ is estimated making use of the cell mapping technique:

$$J(a) \approx J^{CM}(a), \quad a \in \mathcal{I}. \tag{2.7}$$

In this section, the statistical properties of J^{CM} as an estimator of J are studied in order to support the applicability of the optimization schemes employed in this work.

The estimator J^{CM} gathers several approximating steps:

- First, a partition of the state space into cells is defined, so that the domain is approximated by a set of cells. This first step restricts the possible values of the approximate domain size to be a multiple of the cell volume.
- In addition, the cell mapping carries out two additional approximations for the (efficient) computation of the trajectories, which also depend on the cell size:
  - The approximation due to the computation of trajectories only in a finite interval.
  - The approximation due to the use of numerical methods for computing such trajectories.

We consider first the approximation due to the cell partitioning of the state space, by neglecting errors in the computation of trajectories, since this error can be analytically studied. Following the cell mapping definition, those cells in the partition whose center point belongs to $D_{\mathcal{H}}(x^*, a)$ will count as part of the approximation of $D_{\mathcal{H}}(x^*, a)$. Such a set of cells, which is denoted as $D^{\mathrm{P}}(x^*, a)$ (where P stands for "Partition" into cells), can be seen as an approximation of the Lebesgue measure of $D_{\mathcal{H}}(x^*, a)$, which is defined by

$$\lambda(D_{\mathcal{H}}(x^*, a)) = \inf_{\mathcal{C}} \sum_{c_i \in \mathcal{C}} vol(c_i), \tag{2.8}$$

where $\mathcal{C}$ is a countable collection of cells whose union covers $D_{\mathcal{H}}(x^*, a)$. Therefore, we have that $J(a) = vol(D_{\mathcal{H}}(x^*, a))$ would be approximated by $vol(D^{\mathrm{P}}(x^*, a))$ if the errors in the computation of trajectories were neglected.

Note that $D^{\mathrm{P}}(x^*, a)$ depends on $x_o = (a_1, a_2, \ldots, a_n)$, the selected origin for $\mathcal{H}$, and on the cell size h. As mentioned above, a cell will belong to $D^{\mathrm{P}}(x^*, a)$ if the middle point of such a cell is included in $D_{\mathcal{H}}(x^*, a)$. Based on a geometric reasoning, one can expect that the error of the approximation comes from the inclusion or exclusion of those cells c_i containing points in the boundary of $D(x^*, a)$, that is, $c_i \in C_{\mathrm{B}}$. These cells may or may not be included in $D^{\mathrm{P}}(x^*, a)$, depending on the fact that their center point belongs to $D_{\mathcal{H}}(x^*, a)$. Note that x_o can be seen as a random variable. Hence, the inclusion or exclusion of $vol(c_i)$ in $vol(D^{\mathrm{P}}(x^*, a))$ adds up an error to the estimator which can also be seen as a random variable ϵ_{c_i}. Although potentially $\epsilon_{c_i} \in [-V, V]$ ($V = h^n$ being the volume of a cell), for small values of h and smooth domain boundaries (which can be approximated by hyperplanes), one can expect $\epsilon_{c_i} \in [-\frac{V}{2}, \frac{V}{2}]$ having a symmetric distribution (which under some assumptions can be considered to be uniform). In any case, $E(\epsilon_{c_i}) = 0$ and $Var(\epsilon_{c_i}) \leq \frac{V^2}{4}$.

The total size of the cellular domain, $vol(D^{\mathrm{P}}(x^*, a))$, is computed as the sum of the volumes of interior and boundary cells. The error associated with the total

volume estimation, ε, can be computed as the sum of the errors corresponding to the boundary cells $c_i \in C_B$:

$$\epsilon = \sum_{c_i \in C_B} \epsilon_{c_i}. \tag{2.9}$$

It is important to note that ϵ_{c_i} are not independent from each other, their joint distribution strongly depending on the geometry of $D_{\mathcal{H}}(x^*, a)$ with respect to the cell partitioning. In any case, the variance of ε can be bounded $Var(\epsilon) \leq (\#C_B)^2 \frac{V^2}{4}$, where $\#C_B$ stands for the cardinality of C_B. (Note that $\#C_B = \mathcal{O}(h^{-(n-1)})$ so that $Var(\epsilon) = \mathcal{O}(h^2)$ and will tend to zero as $h \to 0$.)

Coming back to the cell mapping technique, besides the state space partition, it computes an approximation of system trajectories in order to approximate invariants and attraction domains. Such approximations include errors due to the state space partition and also errors due to the computation of trajectories, as explained above. Therefore, the cell mapping domain $D^{\mathrm{CM}}(x^*, a)$ provides an approximation of $D_{\mathcal{H}}(x^*, a)$ which in general will depend on x_o and h as well as on the numerical methods employed for the computation of trajectories. Hence, one can also denote it as $D^{\mathrm{CM}(xo,\, h,\, p)}(x^*, a)$, where p stands for the parameters characterizing the numerical approximation of trajectories. Hence, one can characterize the overall approximation of $J(a)$ as follows

$$J^{CM}(a) = J^{CM(x_o,h,p)}(x^*, a) = vol(D^{CM(x_o,h,p)}(x^*, a)) = J(a) + \xi, \tag{2.10}$$

where ξ is also a random variable which gathers ε as well as numerical trajectory computation errors.

Concerning the convergence of the approximation, one can prove that for the simple cell mapping (SCM)

$$\lim_{h \to 0} vol(D^{CM(x_o,h,p)}(x^*, a)) = vol(D_{\mathcal{H}}(x^*, a)), \tag{2.11}$$

that is, $\lim_{h \to 0} J^{\mathrm{CM}(xo,\, h,\, p)}(x^*, a) = J(a)$. This means that even if all the sources of error are considered (due to the three types of approximations mentioned above), convergence to the true value can be guaranteed as the cell size h tends to zero (see Riaza and Zufiria 1999).

Note that in general $J^{\mathrm{CM}}(a)$ has no explicit algebraic expression, and its distribution cannot be characterized analytically. Also, the approximation error ξ (and consequently, $J^{\mathrm{CM}}(a)$) will follow a distribution which depends on specific features of the problem under analysis. In order to get some insight into the statistical properties of J^{CM}, some Monte Carlo simulations have been carried out. In particular, Figs. 2.1–2.3 show the results of the application of the Monte Carlo technique: a rough estimation of the distribution of the random variable $J^{\mathrm{CM}}(a)$ has been obtained for three different problems.

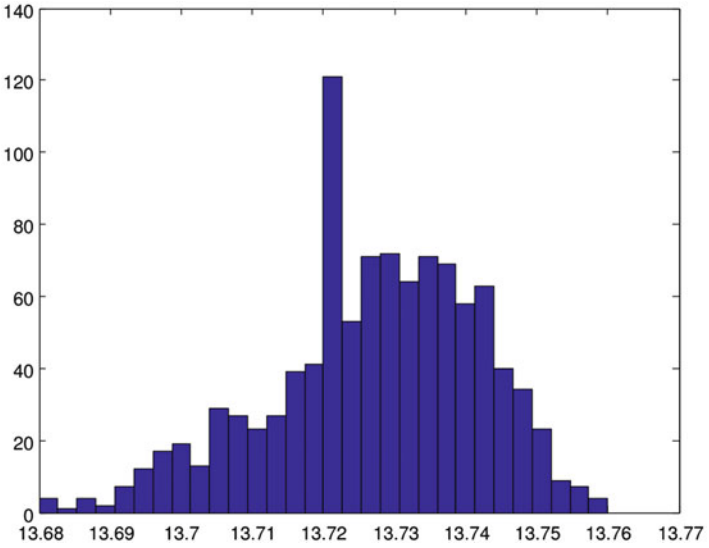

Fig. 2.1 Estimation distribution for van der Pol with $\beta = 1$

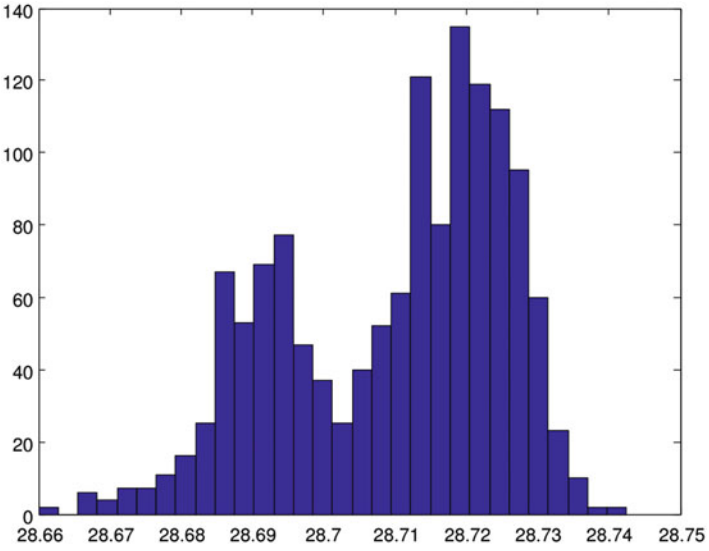

Fig. 2.2 Estimation distribution for van der Pol with $\beta = 5$

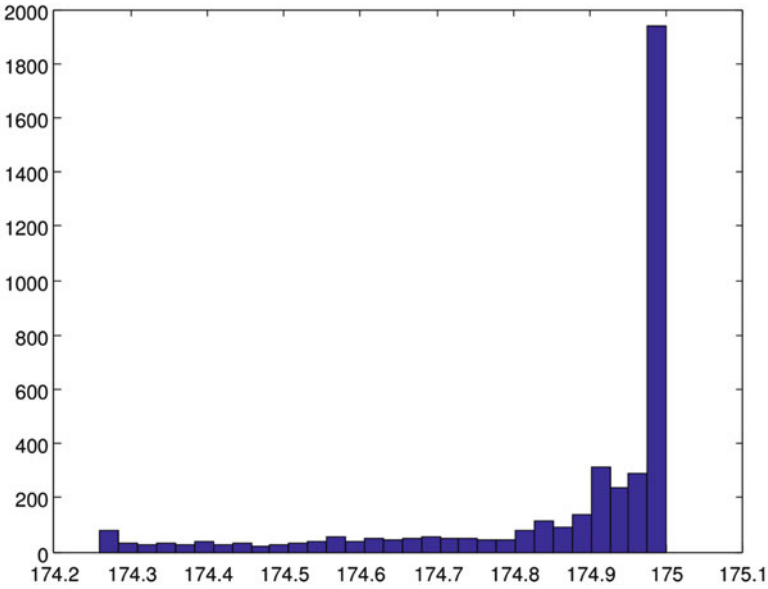

Fig. 2.3 Estimation distribution for saturated control system

2.3 Modifications to CM Implementation

Several remarks about the application and improvement of CM for estimating the size of domains of attraction are presented in this section.

When CM is applied to analyze a dynamical system, some fake solutions can appear. They are cellular invariants which do not correspond to invariants from the original continuous system. These spurious invariants can show up due to either slow dynamics or near to periodic solutions and equilibria. The effects of this type of cellular solutions represent the main difficulty found in the application of CM to our problem.

For instance, when approximating the domain of attraction of a single equilibrium point x^* known to be the only invariant in a given region, CM might detect a fixed cell where x^* is located and another cellular invariant formed by k cells around that fixed cell. This is clearly a fake cellular solution having its own cellular attraction domain. A mistake will be made if only the cellular domain from the fixed cell is taken as an approximation to the attraction domain of x^*. The appearance of fake solutions might be avoided by changing some characteristics of the CM. If this does not work, the two cellular domains can be joined providing the required approximation.

This process of joining domains starts by distinguishing spurious solutions. Then, one must associate their cellular domain of attraction with the corresponding

true invariant set. For doing so, the distance from a cellular invariant to each one of the equilibria of the continuous system is computed first. If the minimum of those distances is lower than a prescribed value, the invariant is considered spurious, its cellular domain being associated with the nearest equilibrium point. If the minimum distance is greater than the reference value, the invariant is not treated as a fake solution.

In some cases the procedure introduced above may not avoid the effects due to fake solutions. The success of this procedure is dependent on the type of dynamical systems under consideration. For instance, in those systems which are known to have a single attractor, all cells in the cellular attraction domains could be taken as associated with the attractor.

2.4 Optimization Techniques

In this section, the main features of the different optimization schemes are outlined.

2.4.1 Genetic Algorithm

Genetic algorithms are search algorithms based on the dynamics of natural selection and natural genetics (Davis 1991; Goldberg 1989). They are mainly employed for finding extreme points in functions where other methods do not work due to the complexity or limitations in the search space. For instance, genetic algorithms are appropriate in case that only raw function evaluations can be performed but no additional information about the function (structure, derivatives, etc.) is available. This is what happens in the type of problems treated in this chapter, where the only available information is the value $vol(D^{\mathrm{CM}}(x^*, a_0))$ for a given $a_0 \in \mathcal{I}$.

In general, a genetic algorithm starts from an initial population composed of individuals (parameter values in our problem). Each individual is represented by a code and evaluated by a *fitness function* (J^{CM} in the problem treated here). The algorithm develops processes of *selection*, *crossover*, *mutation*, and *substitution*. It tries to improve population fitness, generation by generation, and it finishes when a certain percentage of identical individuals is reached, providing then the best-found individual, which approximates the optimum we are pursuing. For more information on genetic algorithms, see Davis (1991), Goldberg (1989) and references therein.

The genetic algorithm used in the simulation examples of this chapter is a basic particular case of the standard procedure: individuals are represented by binary strings of 16 bits (in our case, an individual refers to a parameter value of the dynamical system). Real coding could also be used, being especially suited for multiparameter problems. Every generation is composed by 15 individuals; the fitness function is taken as the function to be maximized (in our problem it is $J^{\mathrm{CM}}(a) = vol(D^{\mathrm{CM}}(x^*, a))$.

The algorithm starts generating randomly 15 individuals which form the initial population. Every bit value, 0 or 1, is selected with a probability of 1/2. Then, in the *selection* process, individuals with a high fitness will have a high probability of being selected. This probability is computed as a sum of two weighed terms. One of them is the inverse of the number of individuals in a generation; the other one is the normalized individual fitness (this is obtained dividing each fitness value by the sum of the fitness values of all individuals in the actual generation).

Crossover follows the selection process. It constructs couples with adjoining individuals in the list provided by the selection. Then, it interchanges pieces of string between individuals in the same couple after a cross site has been randomly selected. This affects only 60% of couples.

New individuals from crossover go through a *mutation*. This is accomplished by a change (with a probability of 0. 3) in one of their bits chosen randomly.

Insertion of new individuals into the population is performed such that individuals with a low fitness value will have a high probability of being replaced. Replacement probabilities are obtained by subtracting the selection probabilities from 1.

After each insertion, the replacement probabilities are recalculated. The new individual will have zero probability to be replaced again, and the rest of probabilities will be normalized.

When crossover, mutation, and substitution have finished for all couples, the process restarts for the new generation.

The algorithm finishes when 90% of individuals in a generation are identical. Since the number of generations needed to fit such requirement can be too high, an upper bound in the number of computed generations will be defined. This number is chosen looking for a trade-off between obtaining a good approximation of the optimal point and minimizing the computational time (i.e., calculating the smallest number of fitness values). If the predefined maximum number of generations is reached, the algorithm selects the individual showing the highest fitness value during the whole process.

2.4.2 Kiefer–Wolfowitz Scheme

Loosely speaking, the Kiefer–Wolfowitz (K–W) algorithm is a stochastic version of the well-known steepest descent (SD) optimization method for cases in which the function to be maximized is not directly available. The SD algorithm is given by the following dynamical systems:

- Discrete form

$$a_{k+1} = a_k + \epsilon_k f_a(a_k). \tag{2.12}$$

- Continuous form

$$\dot{a} = f_a(a), \tag{2.13}$$

where $f(\cdot)$ is the function to be maximized and $f_a(\cdot)$ is its derivative (one dimensional case being considered).

SD can be applied if $f(\cdot)$ and $f_a(\cdot)$ have known analytical expressions. In case those functions were not available but noisy measurements of $f(\cdot)$ could be provided, then an option is estimating $f(\cdot)$ and also its derivative. Incorporating this idea to the SD method, the K–W algorithm appears, being defined by the following difference equation:

$$a_{k+1} = a_k + \epsilon_k(\hat{f}(a_k + h) - \hat{f}(a_k - h))/2h, \tag{2.14}$$

where $\hat{f}(\cdot)$ represents a random variable which estimates $f(\cdot)$. Note that in our problem f is J and $\hat{f}$ is given by J^{CM}; hence, the K–W scheme fits very well with our data availability and associated computational costs.

Stochastic approximation theory assures that, under certain conditions on the step size and the random variables characterizing the estimation procedure, this algorithm converges to a local maximum of $f(\cdot)$. This asseveration is based on the fact that the equation given above follows in mean the differential equation (2.13), the continuous version of the SD method. More details about K–W algorithm can be found in Kushner and Yin (1997).

2.4.3 Neural Network-Based Schemes

One of the main uses of supervised neural networks (NN) is the approximation of the explicit expression of a function $f(\cdot)$ from which only raw sampling values are known (Hassoun 1995).

Hence, the application of an NN to our problem will provide an analytical approximation (let us say, g) of the function $J^{\mathrm{CM}}(\cdot)$, so that its derivatives are easily computed. The next step is to maximize the NN output value using an efficient optimization method, for instance, the steepest descent algorithm or any other traditional scheme, which can make use of the analytical expression of the function to be maximized and its derivatives. Note that these derivatives can be computed following a scheme that is similar to the back-propagation algorithm. One only needs to take into account that:

- The derivative takes the form $\frac{\partial g}{\partial x} = \sum_j \delta_j w_j$.
- δ_j can be computed recursively from δ_{j+1}, satisfying the same relationship as in the backpropagation algorithm.
- δ^s (output error) is equal to 1 (for the case of linear output).

Different schemes can be defined depending on the way that the training procedure and optimization of the NN output are combined. For instance, the optimization procedure can be carried out after an elaborated training, this procedure will be labeled as NN(1) for comparative purposes. Besides that, the optimization procedure can be alternated in the training process, being this scheme named NN(2).

Alternatively, an online scheme can be implemented for getting initial approximations of the maximum to be successively refined. Then, new data can be computed in the neighborhood of this initial maximum estimate, in order to refine such approximation. In addition, some function values that are far from the initial estimate can also be incorporated, in order to avoid getting stuck in local maxima. This procedure has been labeled as NN(3).

2.5 Simulation Examples

In this section, the effectiveness of the proposed techniques is tested on three different dynamical systems. First, a dynamical system having a cubic term with a complicated parameter dependency, second the well-known van der Pol equation with standard parametrization, and finally the tuning of a controller with actuator saturation (this problem depends on two parameters).

2.5.1 Example 1

In this example, the method is applied to a nonlinear system with a complicated dependency on a parameter. The equations defining such system are

$$\begin{aligned} \dot{x}_1 &= (1+\cos^2 a)x_2, \\ \dot{x}_2 &= -x_2 + (a^2 - 10a + 5)(x_1 - {x_1}^3). \end{aligned} \tag{2.15}$$

The selected equilibrium point for the analysis is $x^* = (0, 0)$. The rectangle where CM is focused is $\mathcal{H} = [-4, 4] \times [-4, 4]$, and $\mathcal{I} = [1, 9]$ is the set of parameter values where an optimum is looked for. Since (0, 0) is an attractor $\forall a \in \mathcal{I}$, we can look for $\bar{a} \in \mathcal{I} / \ vol(D((0,0), \bar{a}) \cap \mathcal{H}) = \max\{vol(D((0,0), a) \cap \mathcal{H}), a \in \mathcal{I}\}$.

As it has been explained in previous sections, we work with approximated values for the function $J(\cdot) = vol(D(x^*, \ \cdot) \cap \mathcal{H})$, the approximations being provided by CM. Hence, $J^{\mathrm{CM}}(\cdot) = vol(D^{\mathrm{CM}}(x^*, \ \cdot))$ will denote the function determined by the CM approximations. In this example, the simple cell mapping (Hsu 1987) is used as cellular method, classical fourth-order Runge–Kutta as numerical integration method and the region $\mathcal{H}$ is split into 81×81 cells.

The genetic algorithm chosen to maximize the mentioned function has the structure and characteristics described in Sect. 2.4.1. In this case the maximum number of generations will be fixed to 5, so that the number of fitness values to be calculated is bounded by $15 \times 5 = 75$.

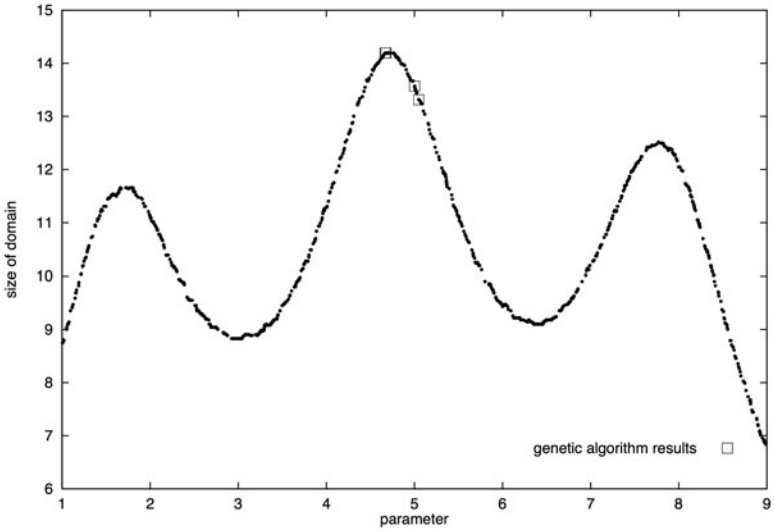

Fig. 2.4 Representation of function $J^{\mathrm{CM}}(\cdot)$

The algorithm has been run four times, providing as a result values of the parameter a in the range $[4.\,6, 5.\,1]$ with corresponding J^{CM} values in $[13.\,3, 14.\,2]$. It took the algorithm around 7 min for providing each estimation of the $\bar{a}$ optimal value.

For the purpose of efficient evaluation, J^{CM} was computed for 1,000 values taken randomly in $\mathcal{I} = [1, 9]$ (this process took around 2 h and 30 min). The representation of those points can be seen in Fig. 2.4.

The results provided by the genetic algorithm are close to the real optimal value of J^{CM} in $\mathcal{I} = [1, 9]$ (Fig. 2.4). This shows the good performance of the procedure in this particular case. One of the cellular attraction domains computed in this example can be seen in Fig. 2.5.

The simulation results for the different optimization methods are displayed in the following table.

	Time	Solution
Genetic	420 s	[4.6,5.1]
K–W	45 s	4.71
NN(1)	32 s	4.69
NN(2)	32 s	4.69
NN(3)	240 s	4.76

It is important to note that the highest computational cost is associated with the function evaluation process. Taking this into account, each of the studied schemes has specific features to be explained below. Hence, this table is only informative and does not have precise comparative purposes.

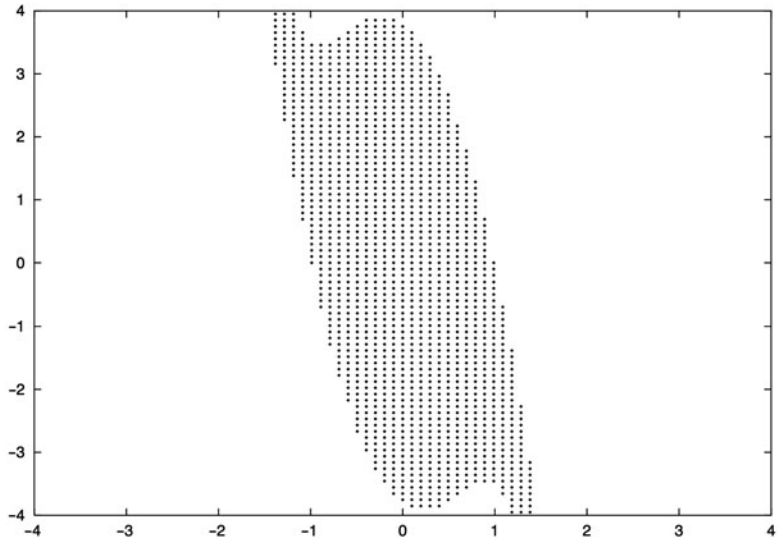

Fig. 2.5 Domain of attraction (Example 1)

As expected, the computational cost of the genetic algorithm applied to this example is remarkably high, but it always provides an estimate of the global optimum, although it may be not too accurate (ranging from 4. 6 to 5. 1).

Regarding K–W algorithm it has been checked that this method requires much less computational time than the GA method, but it presents some risk of getting stuck in a local maximum (20% of failure).

The computational requirements of the NN(1) are low when using a reduced set of data, whereas a proper random selection of such data provides good results in general. The whole computational cost (32 s) can be decomposed in 28 s for obtaining data (evaluating the function), 3.8 s for network training and 0.02 s for each SD iteration (very low cost). Once the network has been trained, the global maximum can be easily determined by trying different initial conditions of the SD method at very low cost.

The NN(2) method, which incorporates SD iterations during the NN training process, has a similar computational cost to NN(1). The advantage of this method is that it can incorporate some stopping criterion based on the sequence of maxima provided by the SD method during the whole process. This means that the algorithm may provide good results in early NN training stages without waiting for a completely trained NN.

The NN(3) method being online oriented (such as GA and K–W) is computationally expensive due to the need of a higher number of function evaluations for maximum refinement purposes. This method also provides information concerning the minimum number of data required for a good function approximation. This information can be employed, for instance, in methods NN(1) and NN(2) (as occurred in this simulation example).

Finally, it should be taken into account that all the NN methods provide an analytical expression for the function approximation, an additional information not obtained through the applications of the other alternative methods. Therefore, SD can be applied in all NN-based methods a posteriori, using different initial conditions, at low cost, thus avoiding the problem of local maxima.

2.5.2 Example 2

This example deals with the van der Pol equation and considers negative parameter values. In that range the behavior is well known, having an unstable limit cycle, with the origin as an attractor whose attraction domain happens to be the region delimited by the limit cycle.

The equations of the system are

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -\beta(1 - x_1^2)x_2 - x_1. \end{aligned} \tag{2.16}$$

The selected equilibrium point for the analysis is $x^* = (0, 0)$. The rectangle where CM is focused is $\mathcal{H} = [-8, 8] \times [-8, 8]$, and $\mathcal{I} = [0.\ 1,\ 5]$ is the set of parameter values where an optimum is searched for. Since (0, 0) is an attractor $\forall \beta \in \mathcal{I}$, we can look for $\bar{\beta} \in \mathcal{I} / \ vol(D((0,0), \bar{\beta})) = \max\{vol(D((0,0), \beta)), \beta \in \mathcal{I}\}$. In this example $D((0,0), \bar{\beta}) \subset \mathcal{H}, \forall \beta \in \mathcal{I}$, so the search process maximizes the whole domain of attraction under consideration.

As in example 1, $J^{CM}(\cdot) = \widehat{f}(\cdot)$ is characterized by using the SCM. Again, the fourth order Runge–Kutta method is applied and $\mathcal{H}$ is divided into 161 $\times$161 cells.

The selected genetic algorithm follows again the structure and features mentioned in Sect. 2.4.1, having 5 as the maximum number of generations.

The algorithm has been run, spending around 20 min for providing each estimation of the $\bar{\beta}$ optimal value. The adjoining cell mapping technique (ACM) (Guttalu and Zufiria 1993; Zufiria and Guttalu 1993) was also tested being less time consuming but providing worse estimates of the sizes of attraction domains. Looking for a compromise between SCM and ACM, the use of the hybrid cell mapping technique seems to be very promising (Riaza and Zufiria 1999). The SCM-based procedure obtained values of the parameter β in the range [4. 97, 4. 99] with corresponding J^{CM} values in [28. 86, 28. 91]. Hence, it seems that the optimal value is reached at the upper extreme of $\mathcal{I}$. As a second part of the analysis in this example we searched for the parameter value which minimizes the size of $D((0, 0), \cdot)$ (i.e., maximizes the function $vol(\mathcal{H}) - J^{\mathrm{CM}}(\cdot)$. The genetic algorithm provided values in [0. 58, 0. 69].

J^{CM} has been computed for 500 values chosen uniformly in $\mathcal{I} = [0.\ 1,\ 5]$ (see Fig. 2.6). About 4 h and 30 min have been necessary for the whole computational process. It can be seen in Fig. 2.6 that the genetic algorithm provides fair approximations to the minimizing parameter value in $\mathcal{I} = [0.\ 1,\ 5]$. Nevertheless,

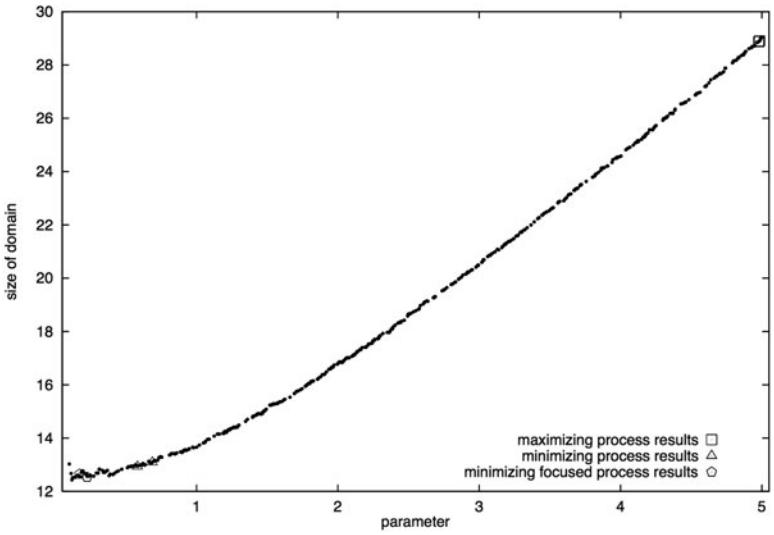

Fig. 2.6 Representation of function $J^{\mathrm{CM}}(\cdot)$

the location of such minimum could be improved via a further analysis focused in the interval [0. 1, 1]. In this case, the obtained values are in [0. 17, 0. 23] with function values in the range [12. 53, 12. 66]. These latest values are good approximations of the minimum in $\mathcal{I} = [0.\ 1,\ 5]$.

The results of the maximizing process and the minimizing (general and focused) process are also represented in Fig. 2.6. The domain obtained for $\beta = 5$, corresponding to the maximum value, appears in Fig. 2.7.

2.5.3 *Example 3: Control Tuning*

This example considers the design of a control system with saturated actuators:

$$\begin{aligned} &\dot{x} = Ax + B \cdot \mathrm{sat}(Fx), \\ &A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ B = \begin{bmatrix} 0 \\ 5 \end{bmatrix},\ F = [f_1, f_2]. \end{aligned} \tag{2.17}$$

where $\mathrm{sat}(u) = \mathrm{sign}(u) \cdot \min\{1, |u|\}$. For fixed feedback vector $F = [-2, -1]$, this system has $x^* = (0,\ 0)$ as an asymptotically stable equilibrium point, and it has been employed by several authors (see, for instance Hu et al. 2002, 2005) to illustrate different schemes to estimate the domain of attraction via the use of Lyapunov functions, being such estimates quite conservative. Here, the problem

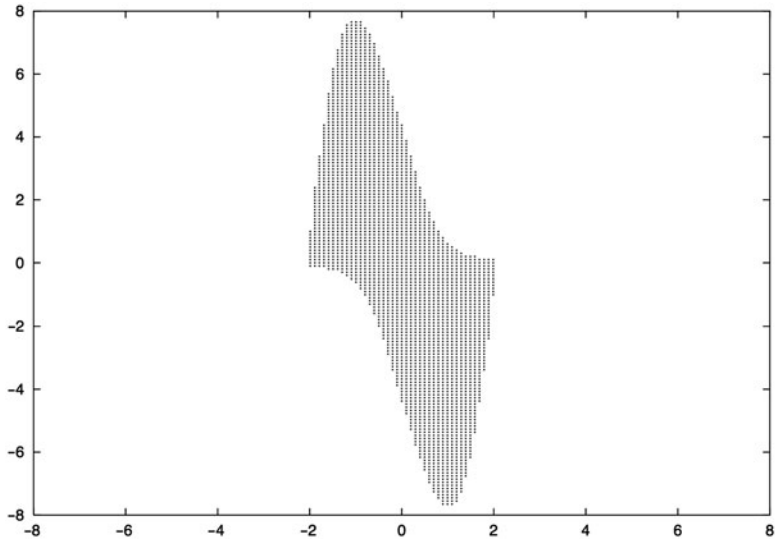

Fig. 2.7 Domain of attraction for $\beta = 5$ (Example 2)

of tuning feedback f_1 and f_2 components has been addressed in order to maximize such domain of attraction.

The system can be defined piecewise as follows:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} x_2 \\ x_1 + 5(f_1 x_1 + f_2 x_2) \end{bmatrix}, \tag{2.18}$$

when $| f_1 x_1 + f_2 x_2 | \leq 1$, and

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} x_2 \\ x_1 + 5 \ \ \text{sign}(f_1 x_1 + f_2 x_2) \end{bmatrix} \tag{2.19}$$

elsewhere.

Note that the origin (0, 0) will always be an equilibrium point of the system. Assuming $f_2 < 0$, we get that for $f_1 < -0.2$ the origin is asymptotically stable, and the system has two additional (unstable) equilibria at $(-5, 0)$ and (5, 0). Therefore, the analysis focuses on the region $(f_1, f_2) \in (-\infty, -0.2) \times (-\infty, 0)$.

When applying the different optimization approaches described in this chapter, the results are quite sensitive to the initial conditions (seeds for the genetic scheme, and initial conditions for Kiefer–Wolfowitz and NN schemes) since the domain CM-estimate reaches a maximum in a parameter region including $(f_1, f_2) \in [-1.8, -0.6] \times [-2.0, -1.0]$ as it can be seen at Fig. 2.8. In any case, the three

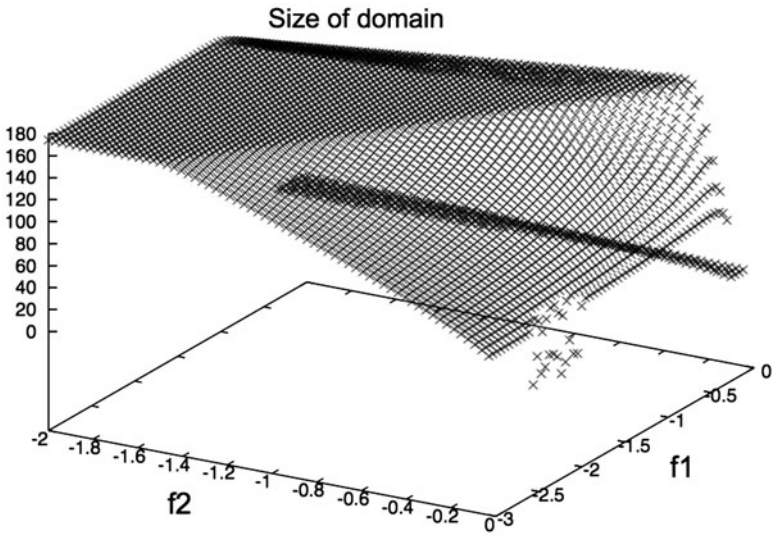

Fig. 2.8 Representation of the cost function in Example 3

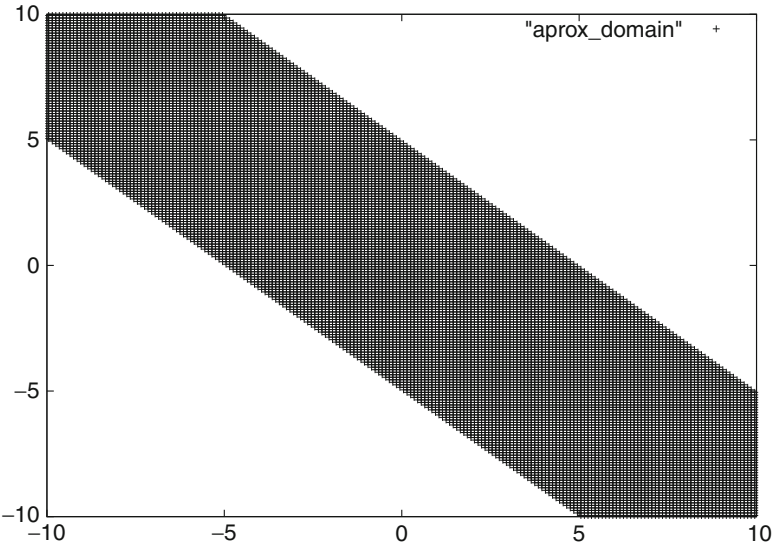

Fig. 2.9 Domain of attraction for tuned feedback $(f_1, f_2) = (-1.47, -1)$ (Example 3)

methods provide solutions within such region. Note that taking $(f_1, f_2) = (-2, -1)$ as in Hu et al. (2002, 2005) provides a domain CM-estimate smaller than when choosing any pair of values in the mentioned optimal region.

Figure 2.9 shows the obtained domain of attraction for $(f_1, f_2) = (-1.47, -1)$ which is clearly larger than the previous existing estimates in the literature.

2.6 Concluding Remarks

Several computational methods to optimize domains of attraction in parametrized dynamical systems have been introduced in this chapter. These methods employ estimates provided by an adaptation of the cell mapping technique for the global analysis of such systems. Three different schemes of optimization are used: genetic algorithms, Kiefer–Wolfowitz algorithm, and Neural Network-based methods. The good performance of the proposed procedures has been illustrated in three particular examples, including the van der Pol nonlinear oscillator and the tuning of a controller with actuator saturation.

Acknowledgments This work has been partially supported by project MTM2007-62064 of the Plan Nacional de I+D+i, MEyC, Spain, project MTM2010-15102 of Ministerio de Ciencia e Innovación, Spain, and by projects Q09 0930-182 and Q10 0930-144 ofthe Universidad Politécnica de Madrid (UPM), Spain.

References

Arkin RC (1998) Behavior-based robotics. The MIT Press, Cambridge

Brockett RW, Li H (2003) A light weight rotary double pendulum: Maximizing the domain of attraction. In: Proceedings of 42 IEEE CDC, Maui, Hawaii USA, pp 3299–3304

Castillo Á, Zufiria PJ (2000) A computational method for maximizing domains of attraction in dynamical systems. Eighth international conference on information processing and management of uncertainty in knowledge-based systems (IPMU2000), pp 165–171, Madrid

Castillo Á, Zufiria PJ (2002) Hybrid schemes for parameter tuning of nonlinear dynamical systems. First international NAISO congress on neuro fuzzy technologies, 100027-01-AC-073. ICSC NAISO Academic Press Canada/The Netherlands, pp 74–80

Castillo Á, Zufiria PJ (2011) Computational schemes for optimizing domains of attraction in dynamical systems. In: Proceedings of the ASME 2011 international design engineering technical conferences & computers and information in engineering conference, DETC2011-48158, 10 pp, Washington, DC

Cohen MA (1992) The construction of arbitrary stable dynamics in nonlinear neural networks. Neural Network 5:83–103

Davis L (1991) Handbook of genetic algorithms. VNR Computer Library, New York

Flashner H, Guttalu RS (1988) A computational approach for studying domains of attraction for non-linear systems. Int J Non-Linear Mech 23(4):279–295

Fu KS, González RC, Lee CSG (1987) Robotics: Control, sensing, vision and intelligence. McGraw-Hill Book Company, New York

Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley Publishing company, Boston

Guttalu RS, Flashner H (1988) A numerical method for computing domains of attraction for dynamical systems. Int J Numer Meth Eng 26:875–890

Guttalu RS, Zufiria PJ (1993) The adjoining cell mapping and its recursive unraveling, Part II: Application to selected problems. Nonlinear Dynam 4:309–336

Hassoun MH (1995) Fundamentals of artificial neural networks. MIR Press, Cambridge

Hsu CS (1985) A discrete method of optimal control based upon the cell state space concept. J Optim Theor Appl 46:547–569

Hsu CS (1987) Cell-to-cell mapping. Springer, New York

Hu T, Goebel R, Teel AR, Lin Z (2005) Conjugate Lyapunov functions for saturated linear systems. Automatica 41:1949–1956

Hu T, Lin Z, Chen BM (2002) An analysis and design method for linear systems subject to actuator saturation and disturbance. Automatica 38:351–359

Hu HT, Tai HM, Shenoi S (1994a) Fuzzy controller design using cell mappings and genetic algorithms. In: Wang PP (ed) Advances in fuzzy theory and technology, vol 2. Bookwrights, Durham, NC, pp 243–264

Hu HT, Tai HM, Shenoi S (1994b) Incorporating cell map information in fuzzy controller design. In: Proceedings of the third IEEE international conference on fuzzy systems, pp 394–399

Kushner HM, Yin GG (1997) Stochastic approximation algorithms and applications. Springer, New York

Lewis FL (1987) Optimal control. Wiley, New York

Martínez-Marín T, Zufiria PJ (1999) Optimal control of nonlinear systems through hybrid Cell-Mapping/Artificial Neural-Networks techniques. Int J Adapt Contr Signal Process 13:307–319

Mitchell TM (1997) Machine learning. WCB/McGraw-Hill, Boston

Moore KL, Naidu DS (1983) Maximal domains of attraction in a Hopfield neural network with learning. In: Proceedings of American control conference, pp 2894–2896

Papa M, Tai HM, Shenoi S (1997) Cell mapping for controller design and evaluation. IEEE Contr Syst Mag 17(2): 52–65

Riaza R, Zufiria PJ (1999) Adaptive cellular integration of linearly implicit differential equations. J Comput Appl Math 111:305–317

Seydel R (1988) From equilibrium to chaos: Practical bifurcation and stability analysis. Elsevier, New York

Xu J, Guttalu RS, Hsu CS (1985) Domains of attraction for multiple limit cycles of coupled van der Pol equations by simple cell mapping. Int J Non-Linear Mech 20(5/6):507–517

Zufiria PJ, Guttalu RS (1993) The adjoining cell mapping and its recursive unraveling, Part I: Description of adaptive and recursive algorithms. Nonlinear Dynam 4:207–226

Zufiria PJ, Martínez-Marín T (2003) Improved optimal control methods based upon the adjoining cell mapping technique. J Optim Theor Appl 118(3):657–680

http://www.springer.com/978-1-4614-3127-5

Global Analysis of Nonlinear Dynamics

Sun, J.-Q.; Luo, A.C.J. (Eds.)
2012, XIV, 287 p. 128 illus., 50 illus. in color., Hardcover
ISBN: 978-1-4614-3127-5