Evolutionary Synthesis and Control of Chaotic Systems

Roman Senkerik*. Ivan Zelinka*, Zuzana Oplatkova*, Donald Davendra*

** Department of Applied Informatics, Faculty of Applied Informatics, Tomas Bata University in Zlin,
Nad Stranemi 4511, 762 72 Zlin, Czech Republic (Tel: +420 57 603 5189;
e-mails: {senkerik, zelinka, oplatkova, davendra}@ fai.utb.cz).*

Abstract: This research deals with the synthesis and control of chaos by means of evolutionary algorithms. The main aim of this work is to show that evolutionary algorithms are capable of synthesis of new chaotic system and optimization of its control and to show a new approach of solving this problem and constructing new cost functions operating in "blackbox mode" without previous exact mathematical analysis of the system, thus without knowledge of stabilizing of the target state. Three different cost functions are presented and tested. The optimizations were achieved in several ways, each one for another desired periodic orbit. The evolutionary algorithm, Self-Organizing Migrating Algorithm (SOMA) was used in its four versions. For each version, repeated simulations were conducted to outline the effectiveness and robustness of used method and cost function. Presented results lend weight to the argument, that proposed cost functions give satisfactory results.

*Keywords**:** Chaos, Control, Evolutionary Algorithms, Optimization, Parameter estimation.*

1. INTRODUCTION

In recent years, the usage of soft-computing methods including evolutionary algorithms (EA) in the field of chaotic systems is increasing and brings about a lot of successful results.

Currently, evolutionary algorithms are known as powerful tools for almost any difficult and complex optimization problem. But the quality of obtained results through the optimizations mostly depends on proper design of the used cost function, especially when the EAs are used for optimization of chaos control. It is well known that chaos in general and also any technique to control of chaos are sensitive to parameter setting, initial conditions and in the case of optimization, they are also extremely sensitive to the construction of the used cost function.

This work presents the package of tools for the synthesis and subsequent control of chaotic systems by means of evolutionary algorithms. It utilizes the Pyragas's delayed feedback control technique (Pyragas, 1992; Just, 1999). Unlike the original OGY control method (Ott, et al., 1990) it can be simply considered as a targeting and stabilizing algorithm together in one package (Kwon, 1999). Another big advantage of Pyragas method is the amount of accessible control parameters. This is very advantageous for successful use of optimization of parameters setting by means of EA, leading to improvement of system behavior and better and faster stabilization to the desired periodic orbits. Some research in this field has been recently done using the evolutionary algorithms for optimization of local control of chaos (Richter & Reinschke, 2000; Richter, 2002).

This research is concerned with the investigation of the design of the "blackbox mode" cost functions securing the stabilization to desired UPO (unstable periodic orbit). The proposed "blackbox" mode approach is very advantageous and simple to implement in this case, because it allows the control the chaotic system without any previous mathematical analysis, thus without knowledge of exact UPOs position.

The control law is based on Pyragas method: Extended delay feedback control – ETDAS (Pyragas, 1995). This research is a continuation of previous experiments with application of EA to chaos control (Zelinka, et al., 2007; Zelinka, et al., 2006; Senkerik, et al. 2006).

2. EVOLUTIONARY SYNTHESIZED CHAOTIC SYSTEMS

In the past, experiments have been made as to how to synthesize various chaotic systems by means of further described methods. Results were published in (Zelinka, et al., 2008). This paper discusses the use of evolutionary algorithms on controlling of selected chaotic systems created via evolutionary processes. Surprisingly, we have found that some of them are barely controllable.

In today's area of evolutionary computation, specialized methods exist, which allows synthesis of complex structures (mathematical formulas, etc.). Well known techniques are for example genetic programming, grammatical evolution or a novelty method called analytic programming (AP), see (Zelinka, et al., 2008; Zelinka 2002; Zelinka & Oplatkova, 2003; Oplatkova & Zelinka, 2006). Sometimes, when these methods are applied on the fitting of measured data, they are also called symbolic regression. The term symbolic regression (SR) represents a process, in which measured data is fitted by a suitable mathematical formula such as x2 + C, sin(x)+1/ex, etc., Mathematically, this process is quite well known and can be used when data of an unknown process is obtained. Historically SR has been in the preview of manual manipulation, however during the recent past, a large inroad has been made through the use of computers. Generally, there are two well-known methods, which can be used for SR by means of computers. The first one is called genetic programming or GP (Koza, 1998; Koza et al., 1999; Koza et al., 2003) and the other is grammatical evolution (O'Neill & Ryan, 2002; Ryan et al., 1998; O'Sullivan & Ryan, 2002).

AP was used in this work for the synthesis of a new chaotic system. Symbolic objects (e.g., variables, constants…) for manipulation and complex structure synthesis were selected from the well-known logistic equation:

$$x_{n+1} = rx_n(1-x_n) \tag{1}$$

This selection was based on the fact that the logistic equation is a well-known simplest system that can produce chaotic behavior. This equation is also well analyzed. It was expected that through evolutionary search, it would be possible to synthesize the logistic equation. Another reason behind the selection of the logistic equation is that results from designed experiments can be easily compared, verified and analyzed.

Basic set of objects used in symbolic regression are {x, A, +, –, *, /}. It is also important to note that evolutionary synthesis of chaotic systems is not restricted only to one-dimensional chaotic maps but can be applied in principle to synthesis of higher-dimensional and more complex chaotic systems. This declaration is based on many other successful complex examples accomplished by GP, GE and AP in the past.

The cost function used for chaos synthesis, when compared with other problems like chaos control (Zelinka, et al., 2006) or black-box optimization (Nolle et al., 2005), is quite a complex structure, which cannot be easily described by a few simple mathematical equations. For the detailed information, please refer to (Zelinka, et al., 2008)

Further text discusses controlling of selected chaotic system from evolutionary synthesized systems.

3. CONTROL OF CHAOS - PROBLEM DESIGN

3.1 Problem Selection and Case Studies

The example of a new chaotic system synthesized by means of AP has the form (2).

$$x_{n+1} = \frac{A\left(2A - 2x_n^{\ 2} - 3x_n\left(A - x_n + Ax_n\right)\right)}{-A + x_n - x_n^{\ 2}} \tag{2}$$

This system exhibits chaotic behaviour for the control parameter A in the ranges <0.1, 0.13> and <0.8, 1.2> (see Fig. 1 and 2).

This work primarily consists of three case studies. All of them are focused on the estimation of three accessible control parameters for the EDTAS method to stabilize desired UPO and the comparison of results for the used cost function. Desired UPOs are the following: p-1 (a fixed point) in the first case, p-2 (higher periodic orbit) in the second case and p-4 (also high periodic orbit) in the last case.

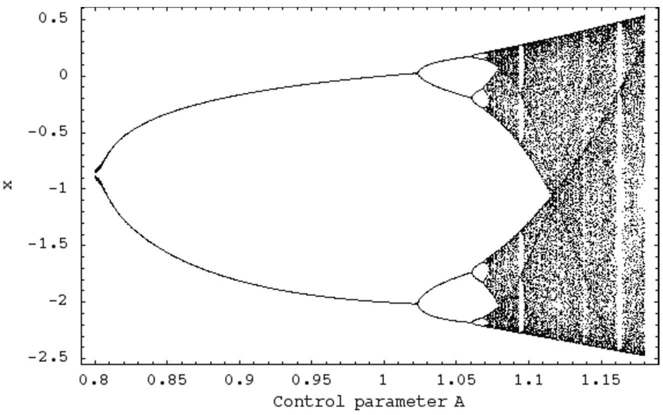

Fig. 1. Bifurcation diagram for A=<0.8, 1.2>

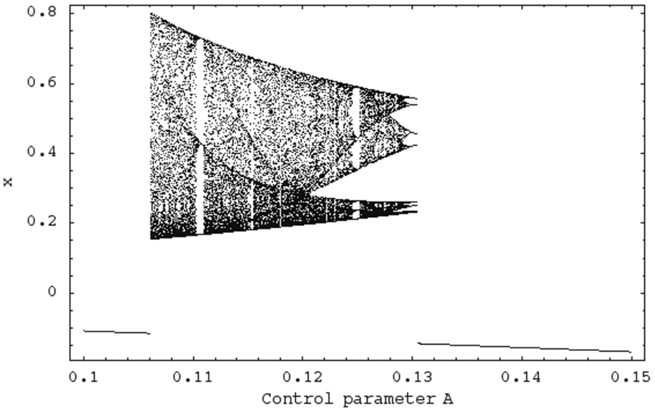

Fig. 2. Bifurcation diagram for A=<0.1, 0.15>

All simulations were 50 times repeated for each EA version to show and check robustness of the used method. The control method – ETDAS in the discrete form suitable for one dimensional Logistic equation is given in (3).

$$x_{n+1} = \frac{A\left(2A - 2x_n^{\ 2} - 3x_n\left(A - x_n + Ax_n\right)\right)}{-A + x_n - x_n^{\ 2}} + F_n$$
$$F_n = K\left[(1-R)S_{n-m} - x_n\right]$$
$$S_n = x_n + RS_{n-m} \tag{3}$$

Where *K* and *R* are adjustable constants, *F* is the perturbation, *S* is given by a delay equation utilizing previous states of the system and *m* is the period of *m*-periodic orbit to be stabilized. The perturbation F_n in equations (3) may have arbitrarily large value, which can cause diverging of the system outside the interval {-2.5, 0.5} (in case of control parameter A=<0.8, 1.2>). Therefore, F_n should have a value between $-F_{\max}$, $F_{\max}$ and EA should find an appropriate value of this limitation to avoid diverging of the system.

3.2 The Cost Function

Several unique designs of cost functions (CF) were developed and tested for the stabilization of p-1 orbit (fixed point), p-2 orbit and p-4 orbit. In previous research the CF had been calculated in general from the distance between desired state (desired UPO) and actual system output on simulation interval – τ. The minimal value of this cost function giving the best solution is zero. The aim of all the simulations was to find the best solution that returns the cost function value as close as possible to zero. This CF is given by (4).

$$CF = \sum_{t=0}^{\tau} \left| TS_t - AS_t \right| \tag{4}$$

Where: TS - target state, AS - actual state

Other cost functions (CF1 CF2 and CF4) had to be used for the stabilizing of the chaotic system in "blackbox mode" ie. without exact numerical value of target state. In this case it is not possible to use the simple rule of minimizing the area created by the difference between the required and actual state on the whole simulation interval – τ.

Our noval approach is based on searching for periodic orbits in chaotic attractor and stabilizing the system on these periodic orbits by means of applying the optimal value of controller adjustable constants (K, $F_{\max}$, R). It means that this new CF did not take into consideration any numerical target state, but the selected target behavior of system. Therefore, the new CFs are based on the searching for optimal values of controller parameters securing the stabilization on any type of selected UPO (p-1 orbit – stable state, p-2 orbit – oscillating between two values etc.). The slight disadvantage of this approach is that for each UPO (i.e. different behavior) a different CF is needed.

The proposal of CF1 used for in the first case (p-1 orbit) is based on the following simple rule. In discrete systems, the iteration $y(n)$ and $y(n+1)$ of output value must be the same. The idea was to minimize the area created by the difference between the n and $n+1$ output iteration on the whole simulation interval – τ, thus at the same time this proposal of CF should secure fast targeting into the close neighborhood of p-1 orbit and its stabilization. The CF$_1$ has form (5).

$$CF_1 = pen1 + \sum_{t=0}^{\tau} \left| y(n+1) - y(n) \right| \tag{5}$$

Where: pen1 = *penalization*

The next proposal of CF2 used for in the second case (p-2 orbit) is based on the following rule. The iteration $y(n)$ and $y(n+2)$ must have the same value. But this rule is also valid for the previous case of – p-1 orbit. Thus another condition had to be added. It says that in the case of p-2 orbit there must be some difference between the n and $n+1$ output iteration. Considering the fact of minimizing the CF the value this condition had to be rewritten into this suitable form (6)

$$\frac{1}{\left| y(n+1) - y(n) \right| + c} \tag{6}$$

Where: c – small constant 1.10^{-16} which was added to prevent the evolutionary optimization from crashing, since upon finding the suboptimal solution stabilized at p-1 orbit it returns the division by zero. The CF$_2$ has the form (7).

$$CF_2 = pen1 + \sum_{t=0}^{\tau} \left| y(n+2) - y(n) \right| + \frac{1}{\left| y(n+1) - y(n) \right| + c} \tag{7}$$

Where: pen1 = *penalization*

The last CF used for the third case (p-4 orbit) is based on the following design. The iteration $y(n)$ and $y(n+4)$ must have the same value. But this is also valid for the two previous cases – p-1 and p-2 orbit. Also, in this case another condition had to be added. It states that in the case of p-4 orbit there must be some difference between the $n+1$ and $n+3$ output iteration. The CF$_4$ has the form (8).

$$CF_4 = pen1 + \sum_{t=0}^{\tau} \left| y(n+4) - y(n) \right| + \frac{1}{\left| y(n+3) - y(n+1) \right| + c} \tag{8}$$

Where: pen1 = *penalization*

In all proposed CFs there had to be included penalization, which should avoid the finding of solutions, where the stabilization on saturation boundary values {-2.5, 0.5} or oscillation between them (i.e. artificial p-2 orbit) occurs. This penalization was calculated as the sum of the number of iterations, where the system output reaches the saturation boundary value.

These three "blackbox" mode CFs (5), (7) and (8) are tested in this work. The difference between them can be clearly seen in Fig. 3 - 5, which shows the dependence of CF values on the adjustable parameters K. From these figures, it is obvious that together with growing complexity of the used CF, the nonlinearity and unpredictability of CF surface also increases. Thus this is the answer for the question as to why EA were used.

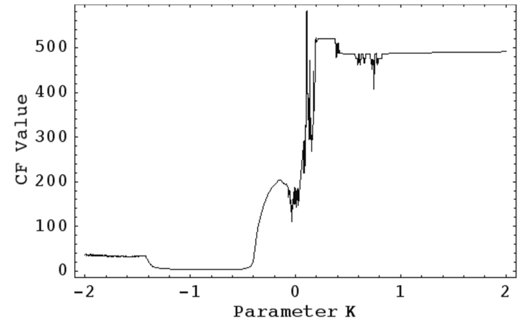

Fig. 3. Dependence of CF value on parameters K for p-1 orbit, $x_{initial}$ = -0.5, CF$_1$, $F_{\max}$ = 0.5, R = 0.2530

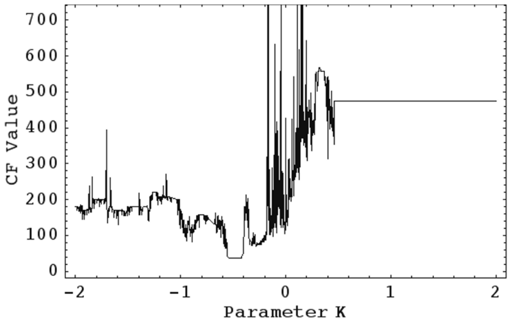

Fig. 4. Dependence of CF value on parameters K for p-2 orbit, $x_{initial}$ = -0.5, CF$_2$, $F_{\max}$ = 2537, R = 0.2676

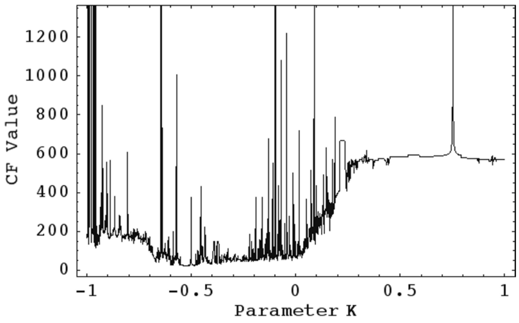

Fig. 5. Dependence of CF value on parameters K for p-4 orbit, $x_{initial}$ = -0.5, CF$_4$, $F_{\max}$ = 8964, R = 0.4569

3.2 Optimizing Algorithm used for Chaos Control

For the experiments described here, stochastic optimization algorithm SOMA (Zelinka, 2004) has been used. It was chosen because it has been proven that this algorithm has the ability to converge towards the global optimum. SOMA works with groups of individuals (population) whose behavior can be described as a competitive – cooperative strategy. The construction of a new population of individuals is not based on evolutionary principles (two parents produce offspring) but on the behavior of social group, e.g. a herd of animals looking for food. This algorithm can be classified as an algorithm of a social environment.

4. EXPERIMENTAL RESULTS

Four versions of SOMA were used for all simulations. See Table 1 for relation between each version and index corresponding to Tables. Parameters for the optimizing algorithm (see Table 2) were set up in such a way, as in order to reach the same value of maximal CF evaluations for all used versions. Each version of SOMA has been applied 50 times in order to find the actual optimum.

The primary aim here is not to show which version is better or worse but to show that the EA can in reality be used for easy to implement control of unknown new synthesized chaotic system, when the "blackbox mode" cost functions are properly defined. The ranges of all estimated parameters were these:

$-2 \le K \le 2$, $0 \le F_{\max} \le 0.5$ and $0 \le R \le 1.0$

Table 1 Used Versions of SOMA

Index	Algorithm / Version
1	SOMA AllToOne
2	SOMA AllToRandom
3	SOMA AllToAll
4	SOMA AllToAllAdaptive

Table 2 Parameter Settings for SOMA

Parameter	ATO / ATR	ATA / ATAA
PathLength	3	3
Step	0.33	0.33
PRT	0.1	0.1
PopSize	25	10
Migrations	25	7
Max. CF Evaluations (CFE)	5400	5670

The best solution for each version of SOMA are shown in Tables 3 – 5 together with other optimization results like Average IStab value for 50 repeated simulations (Avg. IStab). The corresponding figures 6 – 11 show simulation of the best individual solution and the complex simulation of all 200 solutions.

The elementary mathematical analysis of controlled chaotic system was made in order to find the exact positions of UPOs. But these numerical positions of UPOs were not used in any optimization or control process. They were used only for the checking of quality of stabilization and for the simplification of computation of the number of iterations required for exact stabilization during statistical analysis of the obtained results.

4.1 Control of chaos, p-1 orbit, CF$_1$

The first case is focused on the stabilization of p-1 orbit. Unperturbed chaotic system has this p-1 orbit: x_F = -1.0772 (A = 1.15). For the results in this case please refer to Table 3. All SOMA versions gave similar results from the point of view of the CF value for the best solution and Avg Istab value. Fig. 6 displays the best individual solution with the lowest CF value (SOMA ATAA) Fig. 7 gives the simulation of all 200 solutions obtained by all versions of EA. Based on obtained results, it may be stated that the control parameters estimated in the optimizations ensured fast reaching of a desired state without any knowledge about its exact position for all 200 simulations. On average, about 55 iterations are required for successful stabilization.

Table 3 Results for p-1 Orbit, Optimization by Means of CF$_1$

EA	1	2	3	**4**
K	-0.6713	-0.6773	-0.6646	-0.6635
$F_{\max}$	0.4998	0.4992	0.4996	0.5
R	0.2605	0.2597	0.2532	0.2530
CF Value	2.9798	2.9839	2.9801	2.9796
Avg. IStab	**55**	**55**	**55**	**55**

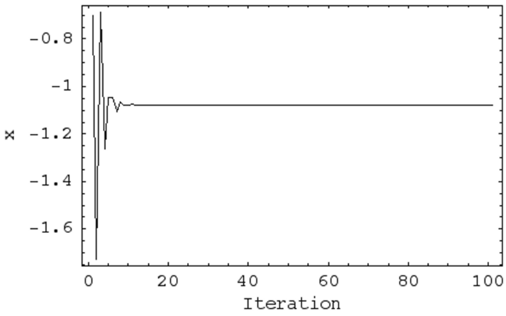

Fig. 6. Best ind. solution: p-1 orbit, CF_1, SOMA ATAA

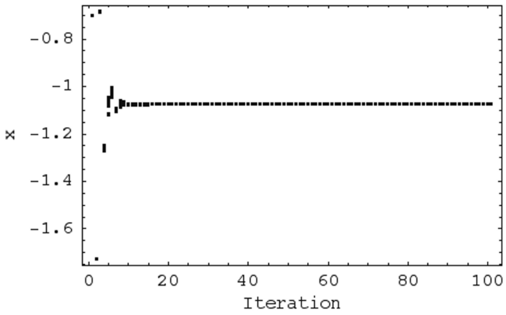

Fig. 7. Simulation of all 200 solutions, p-1 orbit, CF_1

4.2 Control of chaos, p-2 orbit, CF_2

This case is focused on the stabilization of p-2 orbit. Unperturbed new chaotic system has this p-2 orbit (A = 1.15): x_1 = -2.0282, x_2 = 0.1232. For the best results of each SOMA version please refer to Table 4. The simulation of the best solution is depicted in Fig. 8 (SOMA ATAA). The results show similar features as in the previous case. From Fig. 9 it follows that this design of CF secures precise but slightly slower stabilization on desired UPO. For stabilization, on average, about 141 iterations are required.

Table 4 Results for p-2 Orbit, Optimization by Means of CF_2

EA	**1**	2	3	4
K	-0.4377	-0.4291	-0.4511	-0.4384
F_{max}	0.2537	0.2570	0.2499	0.2539
R	0.2676	0.2713	0.2637	0.2678
CF Value	36.2311	36.2298	36.2336	36.2315
Avg. IStab	**147**	**139**	**140**	**139**

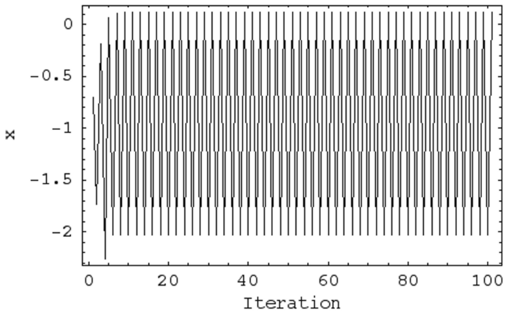

Fig. 8. Best ind. solution: p-2 orbit, CF_2, SOMA ATO

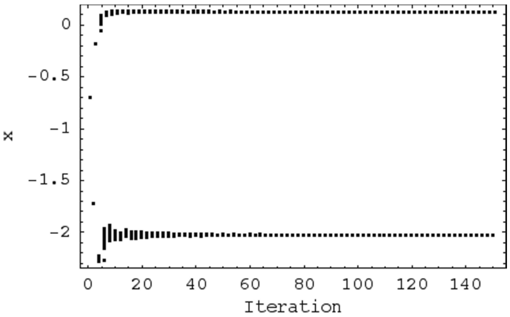

Fig. 9. Simulation of all 200 solutions, p-2 orbit, CF_2

4.3 Control of chaos, p-4 orbit, CF_4

The last case is focused on stabilization of the following p-4 orbit: x_1 = -0.4290, x_2 = -2.1716, x_3 = 0.2521, x_4 = -1.5247. See Table 5 for the results of this optimization. The simulations of the best individual solution (SOMA ATA) are depicted in Fig. 10. From Fig. 11 it follows that increasing order of UPO and complexity of CF surface can cause the phenomenon occurrence of different best individual solutions given by four SOMA versions thus the inability of reaching "exact" stabilization of p-4 orbit for all 200 solutions.

Table 5 Results for p-4 Orbit, Optimization by Means of CF_4

EA	1	2	**3**	4
K	-0.5118	-0.5477	-0.5238	-0.5326
F_{max}	0.1119	0.8991	0.8964	0.8744
R	0.4724	0.4652	0.4569	0.4533
CF Value	24.3821	24.5044	24.3666	24.4140
Avg. IStab	**125**	**133**	**127**	**130**

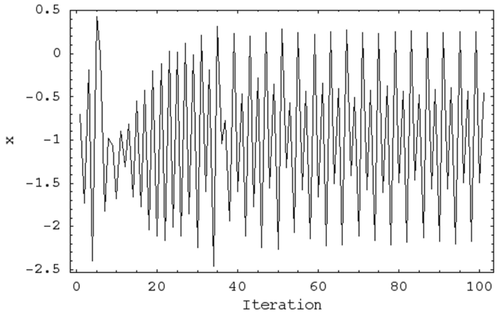

Fig. 10. Best ind. solution: p-4 orbit, CF_4, SOMA ATA

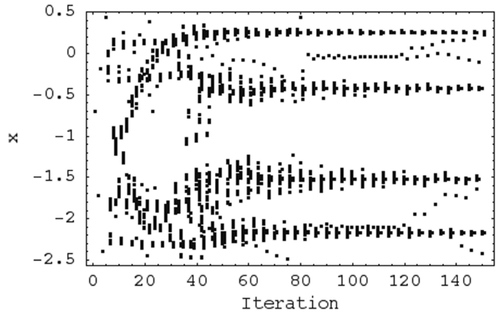

Fig. 11. Simulation of all 200 solutions, p-4 orbit, CF_4

5. CONCLUSIONS

Based on obtained results, it may be claimed that all simulations gave promising results and thus EAs are capable of solving this “blackbox mode” problem. In this work three different CFs were introduced and tested in the task of fast targeting and stabilization of desired periodic orbits of evolutionary synthesized chaotic system. The proposed “blackbox” mode approach is very advantageous and simple to implement in the case of new evolutionary synthesized chaotic system, because of its ability to control the chaotic system without any previous mathematical analysis, thus without knowledge of exact UPOs position.

As can be seen from the optimization results presented here, they are very sensitive to the construction of the used CF. In the first and the second case study, the desired p-1 and p-2 orbit was reached precisely and in the first case also rapidly, but in the in the third case the EA had found lot of solutions which did not secure such precise stabilization. The answer for this problem can be clearly seen from Figs 3 – 5. The first figure (Fig. 3) is related to the p-1 orbit. Here the relatively wide area around the global optimum is highly visible, thus the EA rapidly converges towards this global optimum and all simulations show very fast and precise stabilization of the chaotic system. In the case of p-2 orbit the situation is slightly different. The CF surface is more nonlinear and the area around global optimum is quite tiny. However, the EA was able to find satisfactory solutions with precise and quite fast stabilization. The growing nonlinearity appears with the increase of cost function evaluation required, thus the computational time. In the last case (p-4 orbit), the CF surface is a highly nonlinear and erratic CF and contains a lot of local minima. The area around the global optimum is very tiny and is surrounded by a lot of local minimas, thus it is a harder task for EA to find it. Here occurs the phenomenon that the global optimum represents solution with slow but very precise stabilization, whereas the solutions given by local optimums secures very fast but not precise stabilization on desired UPO. This problem is caused by the inclusion of initial chaotic part into CF value before stabilization. This can be solved by a change of setting-up of EA, change of CF design and implementing the experiences with complex CF designed for higher periodic orbits from previous research. Moreover, the worse ability of stabilization on higher periodic orbits is a well known property of ETDAS control method.

This paper presents a simple tool as to how to check the controllability of new synthesized chaotic system without any numerical analysis. The desired order of UPO is reached by the selection of CF. According to all results shown here it is planned that the main activities will be focused on testing more complex cost functions (not only based on very simple mathematical rules) and other criteria for precision distinguishing of stabilized UPO, together with searching for better settings of EA. These promising results give the possibility for “blackbox mode” evolutionary searching for controller parameters in the task of continuous time systems (Lorenz, Rössler), where the exact mathematical analysis of UPOs position in the chaotic attractor is very demanding.

REFERENCES

Just W., (1999), Principles of Time Delayed Feedback Control, In: Schuster H.G., *Handbook of Chaos Control*, Wiley-Vch.

Koza J. R., Keane M. A., Streeter M. J., (2003), Evolving Inventions, *Scientific American*, p. 40-47

Koza J.R., (1998), *Genetic Programming II*, MIT Press

Koza J.R.,Bennet F.H., Andre D., Keane M., (1999), *Genetic Programming III*, Morgan Kaufnamm pub.

Kwon O. J., (1999), Targeting and Stabilizing Chaotic Trajectories in the Standard Map, *Physics Letters A*. vol. 258, pp. 229-236.

O'Neill M., Ryan C., (2002), *Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language*, Kluwer Academic Publishers.

Oplatkova Z., Zelinka I., (2006), Investigation on Artificial Ant using Analytic Programming, *GECCO 2006 11th conference on Genetic Programming*, Seatle, WA, USA, 8-12.7.2006.

O'Sullivan J., Ryan C., (2002), An Investigation into the Use of Different Search Strategies with Grammatical Evolution, *Proceedings of the 5th European Conference on Genetic Programming*, p.268 – 277.

Ott E., C. Greboki, J.A. Yorke, (1990), Controlling Chaos, *Phys. Rev. Lett*. vol. 64, pp. 1196-1199.

Pyragas K., (1992), Continuous control of chaos by self-controlling feedback, *Physics Letters A*, 170, 421-428.

Pyragas K., (1995), Control of chaos via extended delay feedback, *Physics Letters A*, vol. 206, pp. 323-330.

Richter H., (2002), An Evolutionary Algorithm for Controlling Chaos: The Use of Multi - Objective Fitness Function, *Lecture Notes in Computer Science*, vol. 2439, pp. 308-320.

Richter H., Reinschke K. J., (2000), Optimization of local control of chaos by an evolutionary algorithm, *Physica D*, vol. 144, pp. 309-334.

Ryan C., Collins J.J., O'Neill M. (1998), Grammatical Evolution: Evolving Programs for an Arbitrary Language, *Lecture Notes in Computer Science,* Vol. 1391, pp 83-96.

Senkerik R., Zelinka I., Navratil E., (2006), Optimitazion of Feedback Control of Chaos by Evolutionary Algorithms, *CHAOS'06, In Proc. 1st IFAC Conference on Analysis and Control of Chaotic Systems*, Reims, France, 28-30 June 2006, pages 97-102.

Zelinka I., (2002), Analytic programming by Means of Soma Algorithm, *ICICIS'02, First International Conference on Intelligent Computing and Information Systems*, Egypt, Cairo.

Zelinka I., (2004), “SOMA – Self Organizing Migrating Algorithm”, In: *New Optimization Techniques in Engineering*, (B.V. Babu, G. Onwubolu (eds)), chapter 7, 33, Springer-Verlag,

Zelinka I., Chen G., Celikovsky S., (2008), Chaos Synthesis by Means of Evolutionary Algorithms, *International Journal of Bifurcation and Chaos*, Vol 18, No 4, p. 911 – 942

Zelinka I., Oplatkova Z., (2003), Analytic programming – Comparative Study, *CIRAS'03, The second International Conference on Computational Intelligence, Robotics, and Autonomous Systems*, Singapore, 2003

Zelinka I., Senkerik R., Navratil E. (2007), Investigation on Evolutionary Optimitazion of Chaos Control, Chaos, Solitons & Fractals, doi:10.1016/j.chaos.2007.07.045.

Zelinka I., Senkerik R., Navratil E., (2006), Investigation on Real Time Deterministic Chaos Control by Means of Evolutionary Algorithms, *CHAOS'06, In Proc. 1st IFAC Conference on Analysis and Control of Chaotic Systems*, Reims, France, 28-30 June 2006, pages 211-217.