# EVOLUTIONARY SYNTHESIS OF CONTROL LAW FOR HIGHER PERIODIC ORBITS OF CHAOTIC LOGISTIC EQUATION

[1]Roman Senkerik, [1]Zuzana Oplatkova, [2]Ivan Zelinka, [1]Roman Jasek

[1]Tomas Bata University in Zlin , Faculty of Applied Informatics
Nam T.G. Masaryka 5555, 760 01 Zlin, Czech Republic
{senkerik, oplatkova, jasek}@fai.utb.cz

[2]Department of Computer Science, Faculty of Electrical Engineering and Computer Science
VB-TUO, 17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic
ivan.zelinka@vsb.cz

## KEYWORDS

Chaos, Control, Logistic equation, Evolutionary computation, Analytic programming, SOMA, Differential Evolution.

## ABSTRACT

This research deals with a synthesis of control law for selected discrete chaotic system by means of analytic programming. The novelty of the approach is that a tool for symbolic regression – analytic programming - is used for the purpose of stabilization of higher periodic orbits – oscillations between several values of chaotic system. The paper consists of the descriptions of analytic programming as well as chaotic system and used blackbox type cost function. For experimentation, Self-Organizing Migrating Algorithm (SOMA) with analytic programming and Differential evolution (DE) as second algorithm for meta-evolution were used.

## INTRODUCTION

The interest about the interconnection between evolutionary techniques and control of chaotic systems is spread daily. First steps were done in (Senkerik et al., 2006; 2010a), (Zelinka et al., 2009), where the control law was based on Pyragas method: Extended delay feedback control – ETDAS (Pyragas, 1995). These papers were concerned to tune several parameters inside the control technique for chaotic system. Compared to previous research, this paper shows a possibility how to generate the whole control law (not only to optimize several parameters) for the purpose of stabilization of a chaotic system. The synthesis of control is inspired by the Pyragas's delayed feedback control technique (Just, 1999), (Pyragas, 1992). 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 the Pyragas method for evolutionary computation is the amount of accessible control parameters, which can be easily tuned by means of evolutionary algorithms (EA).

Instead of EA utilization, analytic programming (AP) is used in this research. AP is a superstructure of EAs and is used for synthesis of analytic solution according to the required behaviour. Control law from the proposed system can be viewed as a symbolic structure, which can be synthesized according to the requirements for the stabilization of the chaotic system. The advantage is that it is not necessary to have some "preliminary" control law and to estimate its parameters only. This system will generate the whole structure of the law even with suitable parameter values.

This work is focused on the expansion of AP application for synthesis of a whole control law instead of parameters tuning for existing and commonly used method control law to stabilize desired Unstable Periodic Orbits (UPO) of chaotic systems.

This work is an extension of previous research (Oplatkova et al., 2010a; 2010b), (Senkerik et al., 2010b) focused on stabilization of simple p-1 orbit – stable state. In general, this research is concerned to stabilize p-2 UPO – higher periodic orbits (oscillations between two values). Furthermore it implements checked blackbox approach cost function, thus without knowledge about exact UPO position in the chaotic attractor, which means that AP will synthesize suitable control law based only on the demanded type of chaotic system behavior.

Firstly, AP is explained, and then a problem design is proposed. The next sections are focused on the description of used cost function and evolutionary algorithms. Results and conclusion follow afterwards.

## ANALYTIC PROGRAMMING

Basic principles of the AP were developed in 2001 (Zelinka et al., 2005), (Zelinka et al., 2008), (Oplatkova et al., 2009). Until that time only genetic programming (GP) and grammatical evolution (GE) had existed. GP uses genetic algorithms while AP can be used with any evolutionary algorithm, independently on individual representation. To avoid any confusion, based on use of names according to the used algorithm, the name - Analytic Programming was chosen, since AP represents synthesis of analytical solution by means of evolutionary algorithms.

The core of AP is based on a special set of mathematical objects and operations. The set of mathematical objects

Proceedings 25th European Conference on Modelling and Simulation ©ECMS Tadeusz Burczynski, Joanna Kolodziej Aleksander Byrski, Marco Carvalho (Editors)
ISBN: 978-0-9564944-2-9 / ISBN: 978-0-9564944-3-6 (CD)

is set of functions, operators and so-called terminals (as well as in GP), which are usually constants or independent variables. This set of variables is usually mixed together and consists of functions with different number of arguments. Because of a variability of the content of this set, it is called here "general functional set" – GFS. The structure of GFS is created by subsets of functions according to the number of their arguments. For example $GFS_{all}$ is a set of all functions, operators and terminals, $GFS_{3arg}$ is a subset containing functions with only three arguments, $GFS_{0arg}$ represents only terminals, etc. The subset structure presence in GFS is vitally important for AP. It is used to avoid synthesis of pathological programs, i.e. programs containing functions without arguments, etc. The content of GFS is dependent only on the user. Various functions and terminals can be mixed together (Zelinka et al., 2005), (Zelinka et al., 2008), (Oplatkova et al., 2009).

The second part of the AP core is a sequence of mathematical operations, which are used for the program synthesis. These operations are used to transform an individual of a population into a suitable program. Mathematically stated, it is a mapping from an individual domain into a program domain. This mapping consists of two main parts. The first part is called discrete set handling (DSH) (Figure 1) (Zelinka et al., 2005), (Lampinen & Zelinka, 1999) and the second one stands for security procedures which do not allow synthesizing pathological programs. The method of DSH, when used, allows handling arbitrary objects including nonnumeric objects like linguistic terms {hot, cold, dark…}, logic terms (True, False) or other user defined functions. In the AP DSH is used to map an individual into GFS and together with security procedures creates the above mentioned mapping which transforms arbitrary individual into a program.

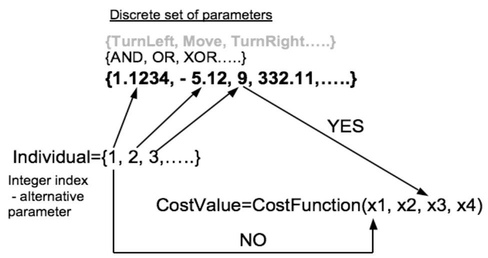



Figure 1: Discrete set handling

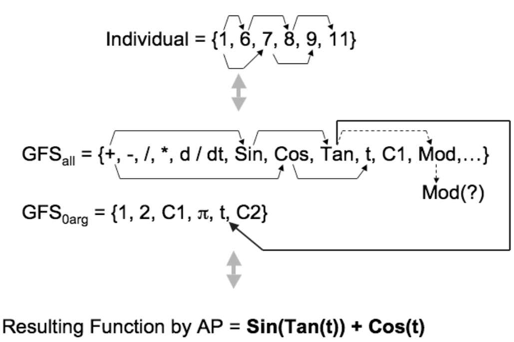



Figure 2: Main principles of AP

AP needs some evolutionary algorithm (Zelinka, 2004) that consists of population of individuals for its run. Individuals in the population consist of integer parameters, i.e. an individual is an integer index pointing into GFS. The creation of the program can be schematically observed in Figure 2. The individual contains numbers which are indices into GFS. The detailed description is represented in (Zelinka et al., 2005), (Zelinka et al., 2008), (Oplatkova et al., 2009).

AP exists in 3 versions – basic without constant estimation, $AP_{nf}$ – estimation by means of nonlinear fitting package in Mathematica environment and $AP_{meta}$ – constant estimation by means of another evolutionary algorithms; meta means metaevolution.

## PROBLEM DESIGN

The brief description of used chaotic systems and original feedback chaos control method, ETDAS is given. The ETDAS control technique was used in this research as an inspiration for synthesizing a new feedback control law by means of evolutionary techniques.

### Selected chaotic system

The chosen example of chaotic systems was the one-dimensional Logistic equation in form (1).

$$x_{n+1} = rx_n\left(1 - x_n\right) \quad (1)$$

The logistic equation (logistic map) is a one-dimensional discrete-time example of how complex chaotic behaviour can arise from very simple non-linear dynamical equation (Hilborn, 2000). This chaotic system was introduced and popularized by the biologist Robert May (May, 2001). It was originally introduced as a demographic model as a typical predator – prey relationship. The chaotic behaviour can be observed by varying the parameter *r*. At *r* = 3.57 is the beginning of chaos, at the end of the period-doubling behaviour. At *r* > 3.57 the system exhibits chaotic behaviour. The example of this behavior can be clearly seen from bifurcation diagram – Figure 3.

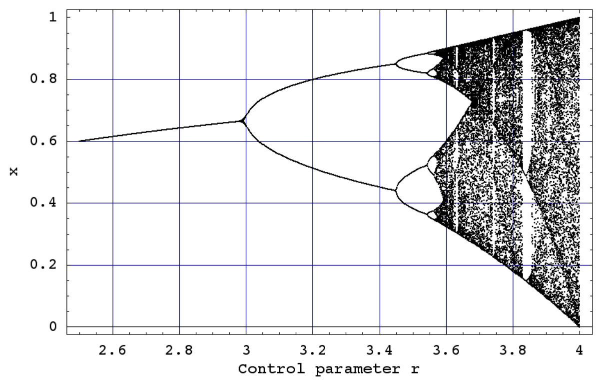



Figure 3: Bifurcation diagram of Logistic equation

**ETDAS control method**

This work is focused on explanation of application of AP for synthesis of a whole control law instead of demanding tuning of EDTAS method control law to stabilize desired Unstable Periodic Orbits (UPO). In this research desired UPO is only p-2 (higher periodic orbit – oscillation between two values). ETDAS method was obviously an inspiration for preparation of sets of basic functions and operators for AP.

The original control method – ETDAS has form (2).

$$F(t) = K\left[(1-R)S(t-\tau_d) - x(t)\right]$$
$$S(t) = x(t) + RS(t-\tau_d) \qquad (2)$$

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 $\tau_d$ is a time delay.

The original control method – ETDAS in the discrete form suitable for one-dimensional logistic equation has the form (3).

$$x_{n+1} = rx_n(1-x_n) + F_n$$
$$F_n = K\left[(1-R)S_{n-m} - x_n\right]$$
$$S_n = x_n + RS_{n-m} \qquad (3)$$

Where: $m$ 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 {0, 1.0}. Therefore, $F_n$ should have a value between $-F_{\max}$, $F_{\max}$. In this preliminary study a suitable $F_{\max}$ value was taken from the previous research. To find the optimal value also for this parameter is in future plans.

Previous research concentrated on synthesis of control law only for p-1 orbit (a fixed point). An inspiration for preparation of sets of basic functions and operators for AP was simpler TDAS control method (4) and its discrete form suitable for logistic equation given in (5).

$$F(t) = K\left[x(t-\tau) - x(t)\right] \qquad (4)$$

$$F_n = K(x_{n-m} - x_n) \qquad (5)$$

Compared to this work, the data set for AP presented in the previous research required only constants, operators like plus, minus, power and output values $x_n$ and $x_{n-1}$. Due to the recursive attributes of delay equation $S$ utilizing previous states of the system in discrete ETDAS (3), the data set for AP had to be expanded and cover longer system output history, thus to imitate inspiring control method for the successful synthesis of control law securing the stabilization of higher periodic orbits

**COST FUNCTION**

Based on the numerous simulations and experiences with evolutionary optimization of chaos control and considering the longer data set for AP causing the searching for the best solution more difficult, the blackbox approach was selected. Numerous experiments proved that this type of CF is less chaotic, nonlinear and erratic.

In the previous research, the CF had been calculated in general from the distance between desired state (desired UPO) and actual system output on a part of simulation interval – $\tau$. The minimal value of this cost function giving the best solution is zero. The aim of all simulations was to find the best solution that returns the cost function value as close as possible to zero. The simplified CF is given by (6).

$$CF = \sum_{t=tstart}^{\tau} \left|TS_t - AS_t\right| \qquad (6)$$

Where: TS - target state, AS - actual state

Other cost functions (CF2) 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 – $\tau$ or its arbitrary part.

Our 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 feedback perturbation $F_n$. It means that this new CF did not take any numerical target state into consideration, but the selected target behavior of system. Therefore, the new CF is based on the searching for optimal feedback perturbation $F_n$ 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 CF2 used in the case of p-2 orbit is based on the following simple rule. The iteration *y*(*n)* and *y(n+2)* must have the same value. But this rule is also valid for the case of – p-1 orbit, where in discrete systems, the iteration *y(n)* and *y(n+1)* of output value must be the same. 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 (7)

$$\frac{1}{\left|y(n+1) - y(n)\right| + c} \qquad (7)$$

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 (8).

$$CF_2 = p1 + \sum_{t=0}^{\tau} |y(n+2) - y(n)| + \frac{1}{|y(n+1) - y(n)| + c} \quad (8)$$

Where: p1 = penalization

In the proposed CF there had to be included penalization, which should avoid the finding of solutions, where the stabilization on saturation boundary values {0, 1} 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.

## USED EVOLUTIONARY ALGORITHMS

This research used two evolutionary algorithms: Self-Organizing Migrating Algorithm (Zelinka, 2004), Differential Evolution (Price, 2005). Future simulations expect a usage of soft computing GAHC algorithm (modification of HC12) (Matousek, 2007) and a CUDA implementation of HC12 algorithm (Matousek, 2010).

### Self Organizing Migrating Algorithm - SOMA

Self Organizing Migrating Algorithm (SOMA) is a stochastic optimization algorithm that is modelled on the social behaviour of cooperating individuals (Zelinka, 2004). It was chosen because it has been proven that the algorithm has the ability to converge towards the global optimum (Zelinka, 2004). SOMA works on a population of candidate solutions in loops called *migration loops*. The population is initialized randomly distributed over the search space at the beginning of the search. In each loop, the population is evaluated and the solution with the highest fitness becomes the leader *L*. Apart from the leader, in one migration loop, all individuals will traverse the input space in the direction of the leader. Mutation, the random perturbation of individuals, is an important operation for evolutionary strategies (ES). It ensures the diversity amongst the individuals and it also provides the means to restore lost information in a population. Mutation is different in SOMA compared with other ES strategies. SOMA uses a parameter called PRT to achieve perturbation. This parameter has the same effect for SOMA as mutation has for genetic algorithms.
The novelty of this approach is that the PRT Vector is created before an individual starts its journey over the search space. The PRT Vector defines the final movement of an active individual in search space.
The randomly generated binary perturbation vector controls the allowed dimensions for an individual. If an element of the perturbation vector is set to zero, then the individual is not allowed to change its position in the corresponding dimension.
An individual will travel a certain distance (called the PathLength) towards the leader in *n* steps of defined length. If the PathLength is chosen to be greater than one, then the individual will overshoot the leader. This path is perturbed randomly.

### Differential evolution

DE is a population-based optimization method that works on real-number-coded individuals (Price, 2005). For each individual $\vec{x}_{i,G}$ in the current generation G, DE generates a new trial individual $\vec{x}'_{i,G}$ by adding the weighted difference between two randomly selected individuals $\vec{x}_{r1,G}$ and $\vec{x}_{r2,G}$ to a randomly selected third individual $\vec{x}_{r3,G}$. The resulting individual $\vec{x}'_{i,G}$ is crossed-over with the original individual $\vec{x}_{i,G}$. The fitness of the resulting individual, referred to as a perturbed vector $\vec{u}_{i,G+1}$, is then compared with the fitness of $\vec{x}_{i,G}$. If the fitness of $\vec{u}_{i,G+1}$ is greater than the fitness of $\vec{x}_{i,G}$, then $\vec{x}_{i,G}$ is replaced with $\vec{u}_{i,G+1}$; otherwise, $\vec{x}_{i,G}$ remains in the population as $\vec{x}_{i,G+1}$. DE is quite robust, fast, and effective, with global optimization ability. It does not require the objective function to be differentiable, and it works well even with noisy and time-dependent objective functions.

## RESULTS

As described in section about Analytic Programming, AP requires some EA for its run. In this paper $AP_{meta}$ version was used. Meta-evolutionary approach means usage of one main evolutionary algorithm for AP process and second algorithm for coefficient estimation, thus to find optimal values of constants in the evolutionary synthesized control law.
SOMA algorithm was used for main AP process and DE was used in the second evolutionary process. Settings of EA parameters for both processes were based on performed numerous experiments with chaotic systems and simulations with $AP_{meta}$ (Table 1 and Table 2).

Table 1: SOMA settings for AP

| | |
|---|---|
| PathLength | 3 |
| Step | 0.11 |
| PRT | 0.1 |
| PopSize | 50 |
| Migrations | 4 |
| Max. CF Evaluations (CFE) | 5345 |

Table 2: DE settings for meta-evolution

| | |
|---|---|
| PopSize | 40 |
| F | 0.8 |
| CR | 0.8 |
| Generations | 150 |
| Max. CF Evaluations (CFE) | 6000 |

Basic set of elementary functions for AP:
GFS2arg= +, -, /, *, ^
GFS0arg= $data_{n-9}$ to $data_n$, K

Total number of 35 simulations was carried out. The most simulations were successful and have given new synthesized control law, which was able to stabilize the system at required behaviour (p-2 orbit) within short simulation interval of 200 iterations. Total number of cost function evaluations for AP was 5345, for the second EA it was 6000, together 32.07 millions per each simulation. See Table 3 for simple CF values statistic.

Table 3: Cost Function values

| | |
|---|---|
| Min | 149.004 |
| Max | 347.57 |
| Average | 203.633 |

The novelty of this approach represents the synthesis of feedback control law $F_n$ (9) (perturbation) for the Logistic equation inspired by original ETDAS control method.

$$x_{n+1} = r x_n (1 - x_n) + F_n \tag{9}$$

Following Table 4 contains examples of synthesized control laws. Obtained simulation results were classified into 3 groups, based on level of approaching to real p-2 UPO, which for unperturbed logistic equation has following values: $x_1$ = 0.3737, $x_2$ = 0.8894. More about this phenomenon is written in conclusion section.
Table 4 covers identification number of UPO approaching level group, direct output from AP – synthesized control law without coefficients estimated, further the notation with simplification after estimation by means of second algorithm DE, corresponding CF value, orbit values between which system oscillates, and identification of figure with simulation results.

Table 4: Simulation results

| Group | Control Law | Control Law with coefficients | CF Value | Orbit Values | Figure |
|---|---|---|---|---|---|
| 1 | $F_n = K_1 x_{n-4} x_{n-2}^{\;2}$ | $F_n = 0.39286 x_{n-4} x_{n-2}^{\;2}$ | 195.881 | 0.98 – 0.44 | 4a |
| 1 | $F_n = -\dfrac{K_1 + x_{n-6}}{K_2 + x_{n-7}} - x_{n-3}$ | $F_n = -\dfrac{x_{n-6} - 50.0535}{x_{n-7} + 47.367} - x_{n-3}$ | 198.685 | 0.98 – 0.44 | 4b |
| 2 | $F_n = x_{n-1}^{\;K_1}$ | $F_n = x_{n-1}^{\;19.464}$ | 149.061 | 0.94 – 0.21 | 4c |
| 3 | $F_n = x_n^{\frac{K_1 x_{n-2}}{K_2 - x_{n-2} + x_{n-1}}}$ | $F_n = x_n^{\frac{12.5103 x_{n-2}}{0.09052 - x_{n-2} + x_{n-1}}}$ | 188.251 | 0.91 – 0.36 | 4d |

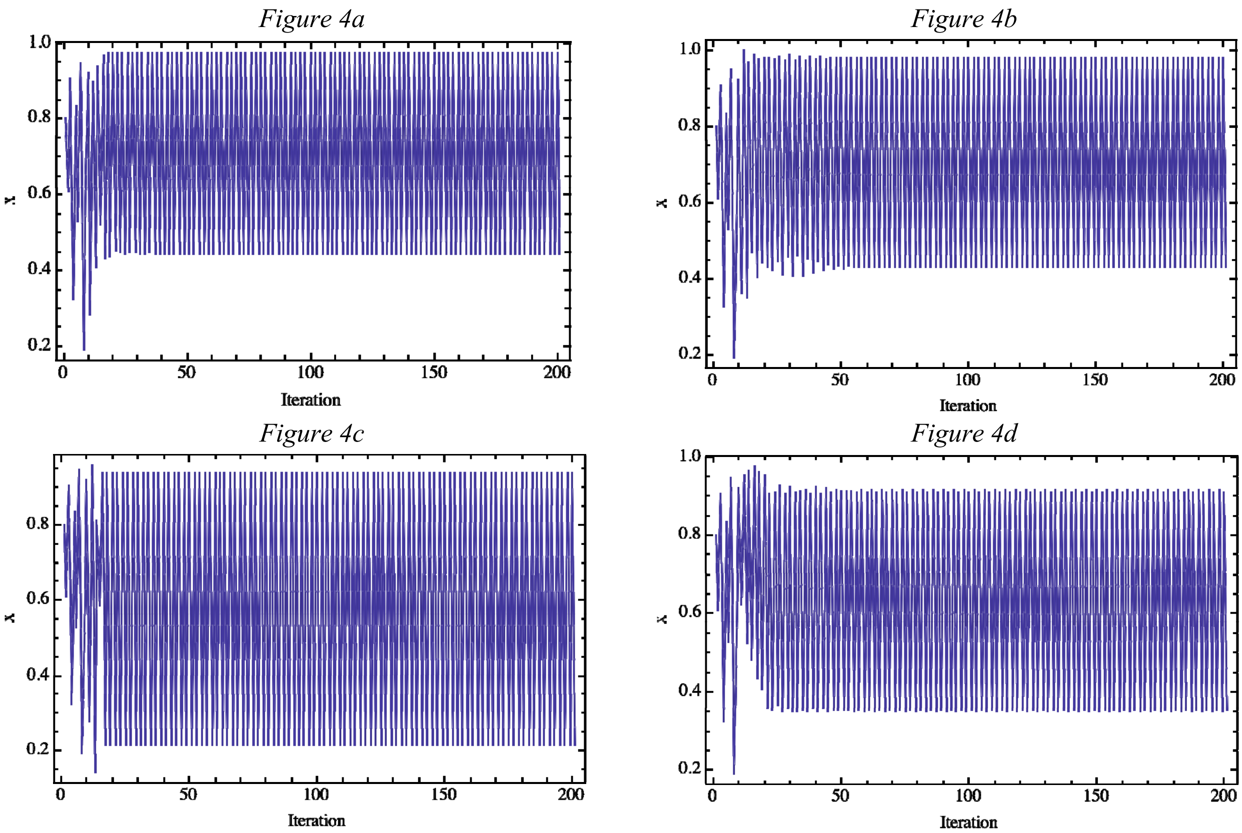



Figure 4: Examples of results – stabilization of chaotic system by means of control laws given in Table 4.

## CONCLUSION

This paper deals with a synthesis of a control law by means of AP for stabilization of selected chaotic system - Logistic equation at higher periodic orbit. In this presented approach, the analytic programming was used instead of tuning of parameters for existing control technique by means of EA's as in the previous research. Presented results reinforce the argument that AP is able to solve this kind of difficult problems and to produce a new synthesized control law in a symbolic way securing desired behaviour of chaotic system and quick stabilization.

An interesting phenomenon was discovered in simulation results. Since there was no information about exact position of p-2 orbit in the chaotic attractor transferred into evolutionary process and cost function was designed to operate in blackbox mode, thus on the basis of selection of desired system behaviour, AP synthesized control laws, which can be classified based on level of approaching to real p-2 UPO. It is very interesting, that these control laws are able to stabilize the chaotic system on optional artificial periodic orbits. Most of common control method was developed for stabilization only on real UPO with low energy costs. The question of energy costs and more precise stabilization will be included into future research together with development of better cost functions, different AP data set, and performing of numerous simulations to obtain more results and produce better statistics, thus to confirm the robustness of this approach.

## AKNOWLEDGMENT

This work was supported by the grants: grant of Ministry of Education of the Czech Republic MSM 7088352101 and grant of Grant Agency of Czech Republic GACR 102/09/1680 and by European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

## REFERENCES

Hilborn R.C., 2000. Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers, Oxford University Press, 2000, ISBN: 0-19-850723-2.

Just W., 1999, "Principles of Time Delayed Feedback Control", In: Schuster H.G., Handbook of Chaos Control, Wiley-Vch, ISBN 3-527-29436-8.

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

Lampinen J., Zelinka I., 1999, "New Ideas in Optimization – Mechanical Engineering Design Optimization by Differential Evolution", Volume 1, London: McGraw-hill, 1999, 20 p., ISBN 007-709506-5.

Matousek R., 2007, „GAHC: Improved GA with HC station", In WCECS 2007, San Francisco, pp. 915-920. ISBN: 978-988-98671-6-4.

Matousek R., 2010, „HC12: The Principle of CUDA Implementation". In MENDEL 2010, Mendel Journal series, pp. 303-308. ISBN: 978-80-214-4120- 0. ISSN: 1803- 3814.

May R.M., 2001, "Stability and Complexity in Model Ecosystems", Princeton University Press, ISBN: 0-691-08861-6.

Oplatková, Z., Zelinka, I.: 2009. Investigation on Evolutionary Synthesis of Movement Commands, Modelling and Simulation in Engineering, Volume 2009 (2009), Article ID 845080, 12 pages, Hindawi Publishing Corporation, ISSN: 1687-559.

Oplatkova Z., Senkerik R., Zelinka I., Holoska J., 2010a, Synthesis of Control Law for Chaotic Henon System - Preliminary study, ECMS 2010, Kuala Lumpur, Malaysia, p. 277-282, ISBN 978-0-9564944-0-5.

Oplatkova Z., Senkerik R., Belaskova S., Zelinka I., 2010b, Synthesis of Control Rule for Synthesized Chaotic System by means of Evolutionary Techniques, Mendel 2010, Brno, Czech Republic, p. 91 - 98, ISBN 978-80-214-4120-0.

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

Price K., Storn R. M., Lampinen J. A., 2005, "Differential Evolution : A Practical Approach to Global Optimization", (Natural Computing Series), Springer; 1 edition.

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, 1995, pp. 323-330.

Senkerik R., Zelinka I., Navratil E., 2006, "Optimization of feedback control of chaos by evolutionary algorithms", in proc 1st IFAC Conference on analysis and control of chaotic systems, Reims, France, pp. 97 - 102.

Senkerik R., Zelinka I., Davendra D., Oplatkova Z., 2010a, "Utilization of SOMA and differential evolution for robust stabilization of chaotic Logistic equation", Computers & Mathematics with Applications, Volume 60, Issue 4, pp. 1026-1037.

Senkerik R., Oplatkova Z., Zelinka I., Davendra D.,Jasek R., 2010b, "Synthesis Of Feedback Controller For Chaotic Systems By Means Of Evolutionary Techniques,", Proceeding of Fourth Global Conference on Power Control and Optimization, Sarawak, Borneo, 2010,.

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, 2004, ISBN 3-540-20167X.

Zelinka I.,Oplatkova Z, Nolle L., 2005. *Boolean Symmetry Function Synthesis by Means of Arbitrary Evolutionary Algorithms-Comparative Study*, International Journal of Simulation Systems, Science and Technology, Volume 6, Number 9, August 2005, pages 44 - 56, ISSN: 1473-8031.

Zelinka I., Senkerik R., Navratil E., 2009, "Investigation on evolutionary optimization of chaos control", Chaos, Solitons & Fractals, Volume 40, Issue 1, pp. 111-129.

Zelinka, I., Guanrong Ch., Celikovsky S., 2008. Chaos Synthesis by Means of Evolutionary algorithms, International Journal of Bifurcation and Chaos, Vol. 18, No. 4 (2008) 911–942

## AUTHOR BIOGRAPHIES

**ROMAN SENKERIK** was born in the Czech Republic,

and went to the Tomas Bata University in Zlin, where he studied Technical Cybernetics and obtained his MSc degree in 2004 and Ph.D. degree in Technical Cybernetics in 2008. He is now a lecturer at the same university (Applied Informatics, Cryptology, Artificial Intelligence, Mathematical Informatics). His email address is: senkerik@fai.utb.cz

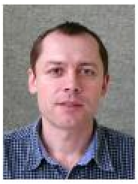
**ZUZANA OPLATKOVA** was born in Czech

Republic, and went to the Tomas Bata University in Zlin, where she studied technical cybernetics and obtained her MSc. degree in 2003 and Ph.D. degree in 2008. She is a lecturer (Artificial Intelligence) at the same university. Her e-mail address is: oplatkova@fai.utb.cz

**IVAN ZELINKA** was born in the Czech Republic, and

went to the Technical University of Brno, where he studied Technical Cybernetics and obtained his degree in 1995. He obtained Ph.D. degree in Technical Cybernetics in 2001 at Tomas Bata University in Zlin. Now he is a professor (Artificial Intelligence, Theory of Information). Email address: ivan.zelinka@vsb.cz .

**ROMAN JASEK** was born in the Czech Republic, and



went to the Palacky University Olomouc, where he studied Informatics and obtained his degree in 1992. He obtained Ph.D. degree in Information Technologies at Charles University in Prague. Now he is an associate professor and head of department of Informatics and Artificial Intelligence at TBU in Zlin. His research interests: Information Security, Artificial Intelligence. Email address: jasek@fai.utb.cz .