SYNTHESIS OF FEEDBACK CONTROLLER FOR STABILIZATION OF CHAOTIC HÉNON MAP OSCILLATIONS BY MEANS OF ANALYTIC PROGRAMMING

Roman Senkerik[(a)], Zuzana Oplatkova[(a)], Ivan Zelinka[(b)], Donald Davendra[(b)], Roman Jasek[(a)]

[(a)]Tomas Bata University in Zlin , Faculty of Applied Informatics, Nam T.G. Masaryka 5555, 760 01 Zlin, Czech Republic
[(b)] Technical University of Ostrava, Faculty of Electrical Engineering and Computer Science, 17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic

[(a)]senkerik@fai.utb.cz, [(b)]ivan.zelinka@vsb.cz

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

Keywords: Chaos Control, Analytic programming, optimization, evolutionary algorithms.

1. 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. 2010a; 2010b), (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 this, current presented research shows a possibility as to 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 law is inspired by the Pyragas's delayed feedback control technique (Just 1999, Pyragas 1992). Unlike the original OGY (Ott, Grebogi and York) 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 research is an extension of previous research (Oplatkova et al. 2010a; 2010b, Senkerik et al., 2010c) 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).

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.

2. ANALYTIC PROGRAMMING

Basic principles of the AP were developed in 2001 (Zelinka et al. 2005). Until that time only Genetic Programming (GP) and Grammatical Evolution (GE) had existed. GP uses Genetic Algorithms (GA) while AP can be used with any EA, independently on individual representation. To avoid any confusion, based on the nomenclature according to the used algorithm, the name - Analytic Programming was chosen, since AP represents synthesis of analytical solution by means of EA.

The core of AP is based on a special set of mathematical objects and operations. The set of mathematical objects is a 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 termed the "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) (See Figure 1) (Zelinka et al. 2005, Lampinen and 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.

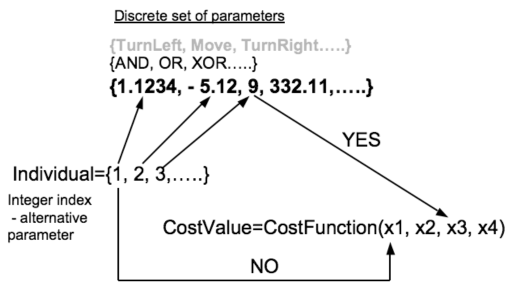

Figure 1: Discrete set handling

AP needs some EA (Zelinka et al. 2005) that consists of a 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 implies metaevolution.

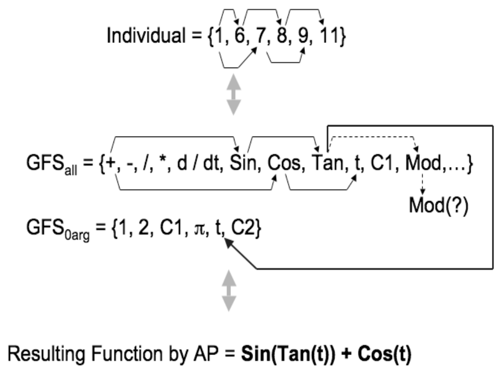

Figure 2: The main principles of AP

3. PROBLEM DESIGN

The brief description of used chaotic system and original feedback chaos control method ETDAS (Pyragas 1995) is given here. The ETDAS control technique was used in this research as an inspiration for synthesizing a new feedback control law by means of evolutionary techniques.

3.1. Selected Chaotic System

The chosen example of chaotic system was the two dimensional Hénon map in form (1):

$$\begin{aligned} x_{n+1} &= a - x_n^2 + by_n \\ y_{n+1} &= x_n \end{aligned} \tag{1}$$

This is a model invented with a mathematical motivation to investigate chaos. The Hénon map is a discrete-time dynamical system, which was introduced as a simplified model of the Poincaré map for the Lorenz system. It is one of the most studied examples of dynamical systems that exhibit chaotic behavior. The map depends on two parameters, *a* and *b*, which for the canonical Hénon map have values of $a = 1.4$ and $b = 0.3$. For these canonical values the Hénon map is chaotic (Hilborn 2000). The example of this chaotic behavior can be clearly seen from bifurcation diagram – Figure 3.

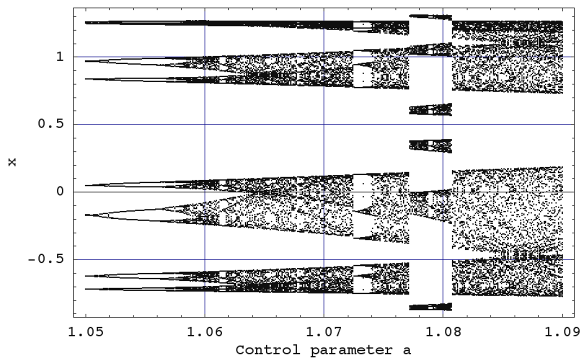

Figure 3: Bifurcation diagram of Hénon Map

Figure 3 shows the bifurcation diagram for the Hénon map created by plotting of a variable *x* as a function of the one control parameter for the fixed second parameter.

3.2. 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 (Pyragas 1995) 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):

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

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 τ_d is a time delay.

The original control method – ETDAS in the discrete form suitable for two-dimensional Hénon Map has the form (3):

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

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 (Pyragas 1992) control method (4) and its discrete form given in (5):

$$F(t) = K\left[x(t-\tau) - x(t)\right], \tag{4}$$
$$F_n = K(x_{n-m} - x_n). \tag{5}$$

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 (x_n to x_{n-9} .), thus to imitate inspiring control method for the successful synthesis of control law securing the stabilization of higher periodic orbits

3.3. Cost Function

Proposal for the cost function comes from the simplest Cost Function (CF). The core of CF could be used only for the stabilization of p-1 orbit. The idea was to minimize the area created by the difference between the required state and the real system output on the whole simulation interval – τ_i.

But another universal cost function had to be used for stabilizing of higher periodic orbit and having the possibility of adding penalization rules. It was synthesized from the simple CF and other terms were added. 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_i, due to many serious reasons, for example: degrading of the possible best solution by phase shift of periodic orbit.

This CF is in general based on searching for desired stabilized periodic orbit and thereafter calculation of the difference between desired and found actual periodic orbit on the short time interval - τ_s (40 iterations) from the point, where the first min. value of difference between desired and actual system output is found. Such a design of CF should secure the successful stabilization of either p-1 orbit (stable state) or higher periodic orbit anywise phase shifted. The CF_{Basic} has the form (6).

$$CF_{Basic} = pen_1 + \sum_{t=\tau 1}^{\tau 2} \left|TS_t - AS_t\right|, \tag{6}$$

where:
TS - target state, AS - actual state
τ_1 - the first min value of difference between TS and AS
τ_2 – the end of optimization interval ($\tau_1 + \tau_s$)
pen₁= 0 if $\tau_i - \tau_2 \geq \tau_s$;
pen₁= 10*($\tau_i - \tau_2$) if $\tau_i - \tau_2 < \tau_s$ (i.e. late stabilization).

4. USED EVOLUTIONARY ALGORITHMS

This research used two evolutionary algorithms: Self-Organizing Migrating Algorithm (Zelinka 2004) and Differential Evolution (Price and Storn 2001, 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).

4.1. Self Organizing Migrating Algorithm – SOMA

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 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. To the same group of algorithms, Particle Swarm Optimization (PSO) algorithm can also be classified sometimes called swarm intelligence. In the case of SOMA, there is no velocity vector as in PSO, only the position of individuals in the search space is changed during one generation, referred to as 'migration loop'.

The rules are as follows: In every migration loop the best individual is chosen, i.e. individual with the minimum cost value, which is called the Leader. An active individual from the population moves in the direction towards the Leader in the search space. At the end of the crossover, the position of the individual with minimum cost value is chosen. If the cost value of the new position is better than the cost value of an individual from the old population, the new one appears in new population. Otherwise the old one remains there. The main principle is depicted in Figures 4 and 5.

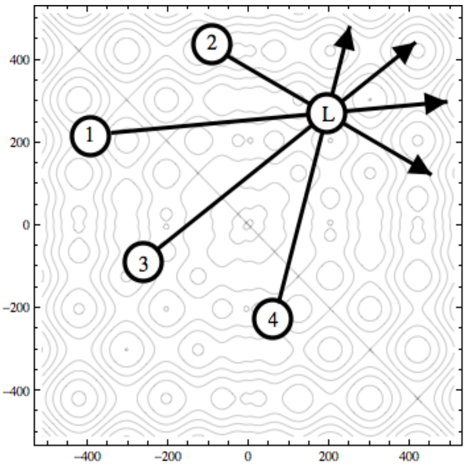

Figure 4: Principle of SOMA, movement in the direction towards the Leader

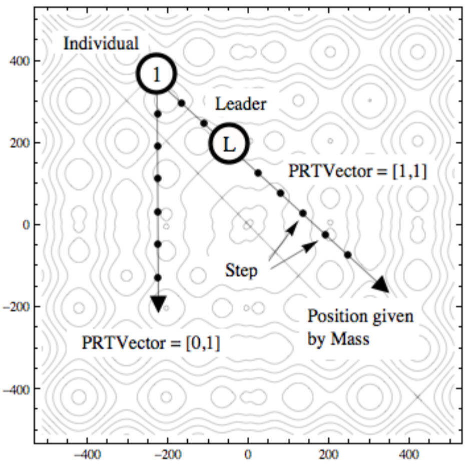

Figure 5: Basic principle of crossover in SOMA, PathLength is replaced here by Mass

4.2. 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. Description of used DERand1Bin strategy is presented in (7). Please refer to (Price and Storn 2001, Price 2005) for the description of all other strategies.

$$u_{i,G+1} = x_{r1,G} + F \bullet \left(x_{r2,G} - x_{r3,G}\right) \tag{7}$$

5. SIMULATION RESULTS

As described in section 2 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} (See Table 1 and Table 2).

Table 1. SOMA settings for AP

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

Table 2. DE settings for meta-evolution

Parameter	Value
PopSize	40
F	0.8
CR	0.8
Generations	150
Max. CF Evaluations (CFE)	6000

The Analytic Programming used following setting-up:
Basic set of elementary functions for AP:
GFS_{2arg}= +, -, /, *, ^
GFS_{0arg}= $data_{n-9}$ to $data_n$, K

Total number of cost function evaluations for AP was 5345, for the second EA it was 6000, together 32.07 millions per each simulation. The novelty of this approach represents the synthesis of feedback control law F_n (8) (perturbation) for the Hénon Map inspired by original ETDAS control method.

$$x_{n+1} = a - x_n^2 + by_n + F_n \tag{8}$$

Following two presented simulation results represent the best examples of synthesized control laws. Based on the mathematical analysis, the real p-2 UPO for unperturbed logistic equation has following values: x_1 = - 0.5624, x_2 = 1.2624.

Description of the two selected simulation results covers direct output from AP – synthesized control law without coefficients estimated; further the notation with simplification after estimation by means of second algorithm DE and corresponding CF value.

5.1. Example 1

The first example of a new synthesized feedback control law F_n (perturbation) for the controlled Hénon map (8) inspired by original ETDAS control method (3) has the form (9) – direct output from AP and form (10) – with estimated coefficients by means of the second EA.

$$F_n = -\frac{x_{n-1}\left(K_2 - x_{n-3} - x_n\right)}{K_1} \tag{9}$$

$$F_n = 0.342699 x_{n-1}\left(0.7 - x_{n-3} - x_n\right) \tag{10}$$

Simulation depicted in Figure 6 lends weight to the argument, that AP is able to synthesize a new control law securing very quick and very precise stabilization. The CF Value was $3.8495.10^{-12}$, which means that average error between actual and required system output was $9.6237.10^{-14}$ per iteration.

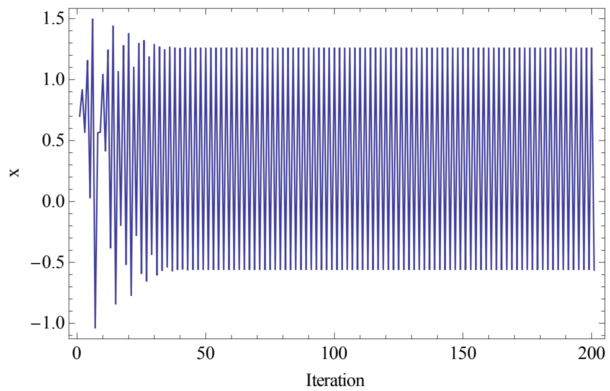

Figure 6: Simulation results - the first example

5.2. Example 2

The second example of a new synthesized feedback control law F_n (perturbation) for the controlled Hénon map (8) inspired by original ETDAS control method (3) has the form (11) - direct output from AP and form (12) – with estimated coefficients.

$$F_n = \frac{x_{n-5}x_{n-1}\left(K_1 + x_{n-3}\right)\left(-\frac{x_{n-7}}{K_3} - x_n\right)}{K_2\left(-\frac{x_{n-7}}{x_{n-5}x_{n-2}x_n} + x_{n-6} + x_{n-4} - x_{n-1}\right)x_n} \tag{11}$$

$$F_n = -\frac{x_{n-5}x_{n-1}\left(x_{n-3} - 25.168\right)\left(-0.5402x_{n-7} - x_n\right)}{4.4124\left(-\frac{x_{n-7}}{x_{n-5}x_{n-2}x_n} + x_{n-6} + x_{n-4} - x_{n-1}\right)x_n} \tag{12}$$

Simulation output representing successful and quick stabilization of Hénon map is depicted in Figure 7. The CF Value was 0.7540 (average error 0.01885 per iteration).

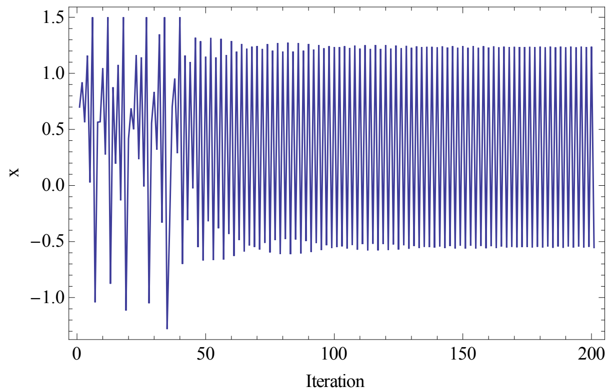

Figure 7: Simulation results - the second example

6. CONCLUSION

This paper deals with a synthesis of a control law by means of AP for stabilization of selected chaotic system at higher periodic orbit. Hénon map as an example of two-dimensional discrete chaotic system was used in this research.

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.

Obtained 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 stabilization.

Presented two simulation examples show two different results. Extremely precise stabilization and simple control law in the first case and not very precise and slow stabilization and relatively complex notation of chaotic controller in the second case. This fact lends weight to the argument, that AP is a powerful symbolic regression tool, which is able to strictly and precisely follow the rules given by cost function and synthesize any symbolic formula, in the case of this research – the feedback controller for chaotic system. 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.

ACKNOWLEDGMENTS

This work was supported by the grant NO. MSM 7088352101 of the Ministry of Education of the Czech Republic and by grants 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.

Oplatková, Z., Zelinka, I.: 2009. *Investigation on Evolutionary Synthesis of Movement Commands*, Modelling and Simulation in Engineering, Vol. 2009.

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.

Price, K. and Storn, R. (2001), *Differential evolution homepage*, [Online]: http://www.icsi.berkeley.edu/~storn/code.html, [Accessed 30/04/2011].

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., Davendra, D., Oplatkova, Z, 2010a, *Evolutionary Design of Chaos Control in 1D*, In. Zelinka I., Celikovski S., Richter H., Chen G.: Evolutionary Algorithms and Chaotic Systems, SpringerVerlag Berlin, pp.165 - 190.

Senkerik R., Zelinka I., Davendra D., Oplatkova Z., 2010b, *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., 2010c, *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, pp. 911–942