Unconstrained evolution of analogue computational "QR" circuit with oscillating length representation

Yerbol Sapargaliyev[1], Tatiana G. Kalganova [1]

[1] School of Engineering and Design, Brunel University
Uxbridge, Middlesex, UB8 3PH, UK
{Yerbol.Sapar, Tatiana.Kalganova}@brunel.ac.uk

Abstract. The unconstrained evolution has already been applied in the past towards the design of digital circuits, and extraordinary results have been obtained, including generation of circuits with smaller number of electronic components. In this paper unconstrained evolution, blended with oscillating length genotype sweeping strategy, is applied towards the design of "QR" analogue circuit on the example of circuit that performs the cube root function. The promising results are obtained. The new algorithm has produced the excellent result in terms of quality of the circuit evolved and evolutionary resources required. It differs from previous ones by its simplicity and represents one of the first attempts to apply Evolutionary Strategy towards the analogue circuit design. The obtained result is compared with previous designs.

Keywords: Evolutionary hardware design, Evolutionary circuit diagnostics and testing.

1 Introduction

The Evolvable Hardware (EHW) is one of the most promising areas of today's electronics. Evolutionary Algorithm (EA) applied towards reconfigurable hardware enables to find a solution among global solution space. The EHW where the ultimate goal is a circuit design is called Evolutionary Electronics [1], [4]. The evolutionary electronics gives the alluring opportunity for an amateur in the field of Electronics to reach out the same results as professional, possessing mostly the knowledge of Darwinian's laws and the inspiration. The EA, navigated by a fitness values, provides randomly created and mutated chromosomes. Each chromosome encodes the structure for a circuit and has to be evaluated a fitness function assigning a fitness value. The fitness value shows how close the current hardware structure by its behavioral characteristics to the required one. The circuits evolved may have unconventional designs and less of all depend on personal knowledge of a designer.

For instance, using simulation software (extrinsic EHW), low-pass filters [2]-[9], high-pass filters [10] and amplifiers [3], [4], [5], [11] are successfully designed with the help of EA. Moreover, the structure and element parameters of reconfigurable hardware by itself could be set as an evolutionary target [13]. To evolve the base to be evolved for further evolutionary designs seems quite promising and interesting perspective, but implies the strict requirements towards the evolutionary technique. The evolutionary technique is a set of rules according to which sweeping strategies [4], the circuit growth strategy, parameters of EA and the circuit representation technique are managed.

In [18] the unconstrained evolution, both spatially and temporally, intrinsically has been applied towards the digital reconfigurable hardware -FPGA. By releasing the full repertoire of behaviors that FPGA can manifest, namely, allowing any connections among modules, letting the evolution to evolve the granularity of modules as well as the regimes of synchronization, evolution has been able to find a highly efficient electronic structure, which requires 1-2 orders less silicon area to achieve the same performance as conventional design does. Ones fully unconstraining the design methodology rules, the natural behavior of analogue elements started to be exploited inside a circuit.

In analogy to this approach, the absolutely unconstrained evolution is applied in [19] towards the originally analogue circuits. In this sense, the range of circuit-structure-checking rules at the netlist composition stage, prohibiting the invalid circuit graphs, are regarded as the main constraints for the design methodology.

In this paper, we utilize the same algorithm as in [2] trying to evolve the analogue circuit that performs the cube root function. Except [2], for this task no circuit was found in the published literature.

The framework of work is to demonstrate the capability of evolutionary process to evolve complex non-linear computational circuits with limited amount of computational resources used, not to apply the developed approach to real world applications. The accuracy of obtained results can be improved if the number of points considered during evaluation process has been increased.

Absolutely agreeing with the factors set in [5] as favorite for evolutionary circuit design: E-12 series of component parameters and modesty in element amount within a circuit, we also set as important the simplicity of the evolutionary technique. Last one makes the experiments to be reconstructed and the same results easier to retain. We propose the simplest oscillating length genotype (OLG) sweeping strategy [3] that together with unconstrained evolution gives excellent results. The evolutionary technique as well as the obtained results is compared with ones published previously on evolution of analogue filters.

The next section overviews the previous work in the area. Section III introduces the whole evolutionary technique. Section IV describes the experimental results together with comparison between the results. And, finally, the last section concludes the paper.

2 Previous Work

The importance of analogue evolutionary circuit design is well described in [3]. In Table 1 we included most of the works in evolutionary analogue circuit design. Most of the works start from evolving a passive low-pass filter. Last one is a convenient tool for probation of evolutionary technique and tuning the EA parameters towards the more sophisticated designs [2], [3], [13]. Our technique also was probed on this task [19] and excellent results were retained.

The considerable results were obtained in [2]. They used Genetic Programming (GP) circuit-constructing program trees approach with four kinds of circuit-constructing functions. They also used automatically defined functions and potentially enabled certain substructures to be reused. They got three kinds of computational circuits including cube root function with very precise performance. The main drawback of this experience is the large computing power required as well as the complicacy of the methodology.

Works [4] and [6] gave the comparison between GP and Genetic Algorithm. The first work was made as analogy to biology concept with comparison of different types of variable length chromosome strategies, while in the second one there was intrinsic evolution of real hardware for robustness purposes.

According to [4], sweeping strategies refer to the way in which the different dimensionalities of the genome space are sampled by the EA. There are three kinds of sweeping strategies introduced by Zebulum et al in [4]: Increasing Length Genotypes (ILG), Oscillating Length Genotypes (OLG) and Uniformly Distributed Initial Population (UDIP). The OLG strategy is a variation of the ILG strategy in which the genotypes are also allowed to decrease in size. The main purpose of this strategy is to create pathways from large to smaller genotypes with similar fitness values. In UDIP instead of starting with a population of small genotypes, the initial genotype sizes are now randomly assigned to values ranging from one to the maximum number of genes. In [4] all tree strategies were applied to evolving the LCR (filters) and QR (amplifiers) circuits, and ILG and OLG strategies have shown superior results.

Table 1. Recent advanced on the evolution of analogue circuits, SA is Simulated Annealing

		Year	Type of EA	Circuit-structure-checking rules	Parameter optimization	Sweeping strategy
Koza et al [2]		1997	GP	Partially	No	ILG
Lohn, Colombano [3]		2000	GA	Yes	No	ILG
Goh, Li [5]		2000	GA	Yes	No	ILG
Zebulum, et al [4]		1998	GP,GA	Yes	No	ILG,OLG,UDIP
Grimbleby [7]		1999	GA	Data n/a	numerical	ILG
Dastidar, et al [11]		2005	GA	Yes	GA	OLG
Ando, Iba [6]		2003	GP,GA	Yes	GP,GA	Data n/a
Sripramong et al [12]		2002	ES+SA	Yes	hill-climbing	Fixed
Sapargaliyev, Kalganova	LCR circuit [19]	2006	ES	No	No	OLG
	QR circuit (proposed)	2008	ES	No	No	OLG

As could be noticed from the Table 1, the following gaps do still exist, which we try to fill with work presented:

- The most of previous research in analogue circuit design used the circuit-structure-checking rules for avoiding invalid circuit graphs.

- The most of works in low-pass filter design used the ILG strategy as a sweeping strategy; however in work [4] the OLG has shown excellent results for analogue circuit design, and the best for low-pass filter design.

In the frame of this work we will try to fill these gaps in someway.

3 Unconstrained evolution of “QR” circuits

Representation. We use for circuit building only tree types of elements: Qn which is n-p-n bipolar transistor, Qp which is p-n-p bipolar transistor and R – resistors. The linear circuit representation is proposed for use, similar to one that exploited in [4], that is every element of a circuit is represented as a particular gene, and each gene consists of 4 loci corresponding to element’s features: element’s name, node numbers to each pin and parameter (only for R).

Resistor encoding:	Bipolar transistor encoding:
Re N1 N2 Pa	Qx N1 N2 N3

Fig. 1. A gin coding a resistor (left) and a bipolar transistor (right): Re-loci is the resistor’s name; Qx-loci is the transistor’s name; N1, N2, N3 -loci are the nodes for the first, the second and the third pins; Pa-loci is the element’s parameter. Amount of variations is shown over each loci.

On Fig.1 (left) is a view of a gene coding a resistor, Fig.1 (right) is representation for a bipolar transistor. The gene looks exactly the same as an element line in the PSPICE netlist, so, there is no necessity to convert a genotype into a netlist.

For resistor Pa-loci we set 64 possible values, according to E-12 order, which means there are 12 parameters per decade. That is we covered 5 decades plus 4 “just-in-case” parameters located in upper and lower neighboring decades.

Unconstrained evolution of “QR” circuit. In [19] we called absolutely unconstrained evolution of an analogue circuit the process of circuit netlist generation during which no circuit-structure-checking rules applied and all the circuits are counted as valid graphs except ones that have elements with dangling nodes and with isolated sub-circuits. Note that the term “unconstrained” is applied to the duration of evolutionary process, i.e. the way of obtaining the target circuit that differs from the way how the final valid circuit structure is obtained. In other words, the automatic design of logic circuit contains of two main stages: 1) unconstrained evolution of logic circuit; 2) generation of final valid circuit structure.

There are two main kinds of invalidities that unacceptable for most of simulation software and that just prohibited by most of researchers in the area (Table 1): the nodes without DC path to ground (tackled in [2]) and loops involved inductors and/or voltage source (tackled in [19]). In [2] and [19] these problems were tackled by using R-support that made almost any randomly generated circuit valid.

In this work where we do not use the reactive elements (L and C), the task of unconstraining the evolutionary search becomes easier, because both kinds of elements that we use have internal resistances and both of them being anywhere inside the circuit have DC paths to the ground.

However, in most of the works listed in Table 1, rules prohibiting some transistor connections, such like emitter-to-collector connection, were applied to reduce the solution space. These connections do not certainly produce non-convergence errors. Emitter-collector connection leads only to temporarily tripping of that particular transistor from the whole circuit, until the further mutation process suddenly activates it. In this work we do not prohibit such kind of connections removing these last constrains. Thus, since we do not have any other kinds of constrains that have to be released, Our technique can be called as an absolutely unconstrained evolution for “QR” circuits.

Experiment settings. The embryo circuit is the element or the number of elements (including the voltage source), that can be predetermined for the particular circuit to ease the further circuit growth. We define the embryo circuit correspondent to the most popular case where the circuit is driven by a DC voltage source, source resistance Rsource=1kΩ and the load resistance Rload=1kΩ. These three elements compose the embryonic circuit and absolutely identical to that ones in most of the works (Fig. 2). The output voltage is measured on the pins of Rload.

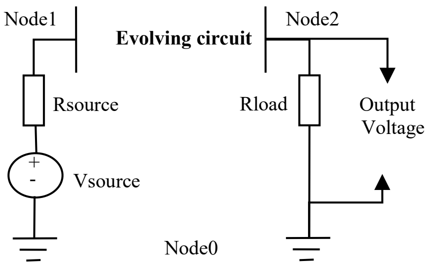

Fig. 2. Embryo Circuit

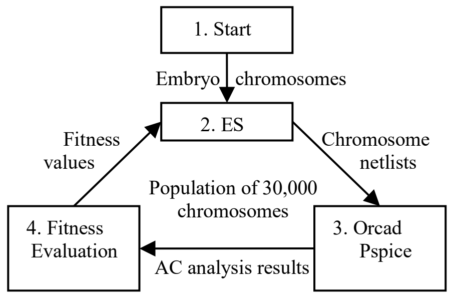

Fig. 3. The flowchart of the experiment

Fig.3 generally shows the algorithm of the experiment. It consists of 4 main blocks. The PC program written in C programming language describes all 4 parts and unites them in one code.

The Start-block provides population of chromosomes in form of PSPICE netlists. This block includes all the data necessary for embryo circuit production: initial parameters, element description and analysis options. Being delivered to ES block, every chromosome at this stage is grown up from the embryo to the individuals with the same number of genes (elements). We set this number at three.

ES part sets the particular parameters of ES, such as: mutation rate, population size, selection criteria and termination terms. It modifies the genotype and produces the population of chromosomes in form of cir-batch-file towards OrCAD PSPICE. Last one utilized in non-interactive batch simulation mode.

Block 3 downloads cir-file to the PSPICE, receives the result in form of out-file and passes it for evaluation to Block 4.

Block 4 defines the best chromosome, selects 10% fittest individuals and sends them to the Block 1. At this stage Start block, depending on the best fitness, decides which chromosome to send to which mutation.

As can be seen on Fig.5, the whole process consists of three kinds of operations over the each chromosome. Depending on correlations among lengths and fitnesses of the best and current chromosomes, each chromosome it put under the particular mutation. Add_new_element_mutation (ANEM) is a procedure, when one randomly generated gene is added to each chromosome except the chromosome with the best fitness value. We set the maximum difference in length between the shortest and the longest chromosome up to 5 genes. ANEM is applied in cases when the current individual is longer than the best one.

To restrain the difference between the shortest and the longest chromosomes the Delete_element_mutation (DEM) is devoted, that deletes one gene if the difference exceeds the limit. Thus, the evolution can focus on processing chromosomes of five different neighboring sizes.

The “circuit structure mutation” (CSM) performs mutation over any of four loci of randomly chosen gene (Fig.1). If the mutation comes to a pin connection, the whole structure of a circuit is changed. Despite the total amount of elements stays unchangeable, the number of nodes of a circuit could be reduced or increased. CSM also performs the parameter optimization procedure.

Fitness Function and Termination Criteria. The target for evolutionary search is to evolve an analog circuit which output voltage is the cube root of its input voltage. To enable ourselves to make the estimation the final results from experiment we have set the same fitness terms as in [2]. That is, we made the PSPICE simulator to perform a DC sweep analysis at 21 equidistant voltages between –250 mV and +250 mV for the cube root. Fitness value is set to the sum, over these 21 fitness cases of the absolute weighted deviation between the target value and the actual output value voltage produced by the circuit. The smaller the fitness value, the closer the circuit to the target. We set for fitness to penalize the output voltage by 10 if it is not within 1% of the target voltage value. The error circuits, that mostly are non-convergent, are not analyzed by simulator and assigned to the worst fitness value that never could be reached by other circuits.

We set as a termination criteria reaching either of the following conditions: the fitness value does not improve over 20 generations (600 000 individuals), or the best circuit reaches more than 100 elements, or the best fitness value reaches 0.5. According to experience, the most frequent termination way was the stop in fitness increase.

4 Experimental Results

The result presented is the best out of 20 runs at 10 different PCs with different seed for the random number generator. The Evolutionary Strategy with linear representation and oscillating length genotype is utilized. The total population was 30 000 individuals, mutation rate 5%.

At generation 3 the best individual (No 24999) with 3 genes (in addition to embryo elements) shown the fitness 65.57. The circuit that this chromosome describes is presented on Fig. 4 and has 2 transistors and one resistor.

The next notable result appeared at generation 15 (No 23882) with 14 genes (in addition to embryo elements), which describes a circuit with 7 transistors, 1 diode (transistor whose collector is connected to the base) and 6 resistors. This circuit, pictured on Fig.5, has the fitness 5.53.

And, finally, the circuit that evolution reached after 133 generations (No 24318), and which evolution was not able to improve during the next 20 generations, appeared with totally 38 elements (in addition to embryo elements): 24 transistors, 12 resistors and 2 diodes (Fig. 6). The **fitness of this circuit achieved 2.27.**

The average error-circuits per population was 4-5%, mostly they are non-convergent.

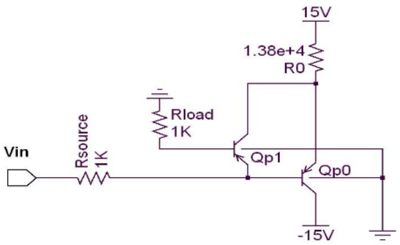

Fig. 4. Best circuit from generation 3

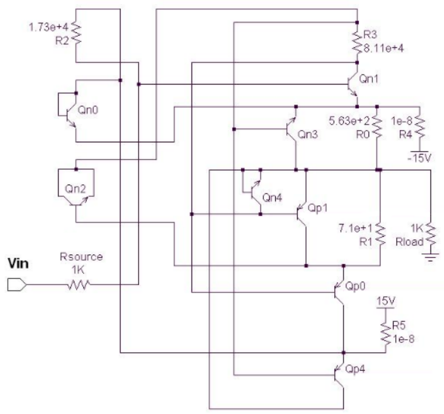

Fig. 5. Best circuit from generation 15

Table 2. Comparison of evolved cube root computational circuits

	1-st circuit		2-nd circuit		3-rd circuit	
	Element number_1	Fitness value_1	Element number_2	Fitness value_2	Element number_3	Fitness value_3
Evolved filter, Koza, 1997 [2]	3	77.7	18	26.7	50	1.68
Evolved filter in this work	3	65.6	14	5.53	38	2.27
Gain	0	15.57%	22.22%	79.29%	24.00%	-35.12%

The cube root computational circuits designed either conventionally or using evolutionary process is *very rare* to find in literature. Up to date we were unable to identify the work that will design non-linear computational circuits such as cube root, etc. conventionally [2]. To decide on efficiency of the proposed evolutionary technique we have only one work directly to address to [2]. Since we set the same fitness function, we can directly compare the circuits and their corresponding fitness values. The result of comparison is presented in Table 2.

5 Conclusion

In this paper we applied the unconstrained evolution towards the analogue circuit design on the example of “QR” computational circuit performing the cube root function.

The method utilized here is much easier than that one applied in [2]. While last approach, with help of reusable sub-constructions, successfully evolved circuits with large amount of elements, our method, as it could be seen from Table 2, succeeds in small and middle sized circuits.

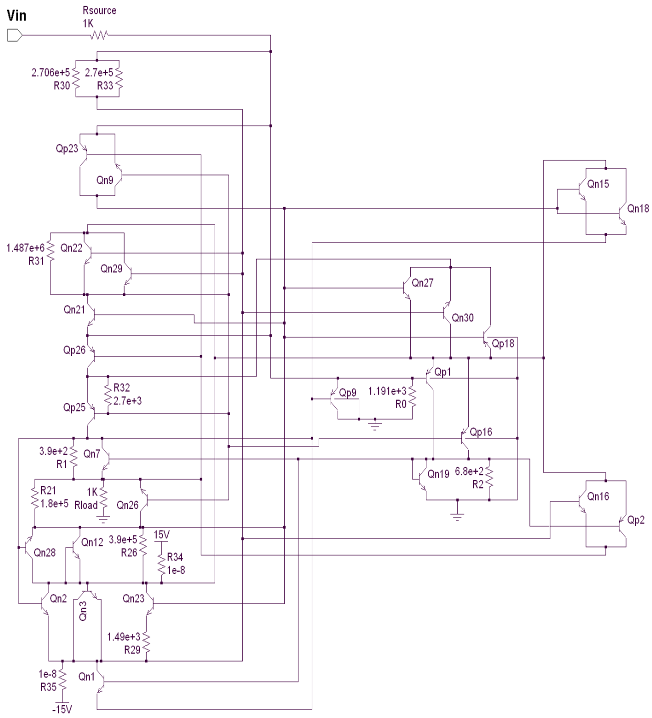

Fig. 6. Best circuit from generation 113

As it could be seen from Table 3, the computer resources, and thus potentially time, in our attempt much lower, reaching in average 90%.

The proposed method has shown its potential for further improvements by getting the fitness that close to one in [2]. If compare the final results, the *shortage* in fitness value is almost the same (**35%**) as the gain in element economy (**24%**).

Table 3. Comparison of computational resources

	1-st circuit	2-nd circuit	3-rd circuit
Evolved filter, Koza, 1997 [2]. 640,000 individuals per population	Gen. #0, 0-640,000	Gen. #17, 10,240,000-10,880,000	Gen.#60 37,760,000-38,000,000
Evolved filter in this work. 30,000 individuals per population	Generation #3 90,000	Generation #15 443,882	Generation #133 3,984,318
Gain	Up to 86%	Up to 96%	Up to 89%

This paper has shown one of the first successful attempts of application of Evolutionary Strategy towards the analogue circuit design.

The oscillating length genotype sweeping strategy with capability of evolution to focus on the limited genotype length dispersion proved its high surviving character.

Despite our attempt to succeed the works [2], last one, accomplished by the leading research group in evolvable hardware, exemplifies for us the perfect result ever made by evolutionary tool on the analogue circuit design.

Note that the GA convergence and averaged result over 20 or 100 runs have not been completed since there is not existing work these parameters can be compared with. This is subject for future work.

The analysis of the evolved circuit structures demonstrated in this work as well as in [2], shows that in order to provide the circuit structures stable to variations in temperature, supply voltage, noise, etc. it is necessary to complete the second evolutionary process that will concentrate on evolution of stable wiring. This will make the first step towards evolution of real-world analogue circuits, stable to changes in the environment, and, consequently, the evolvable hardware design approach can reach the competiveness stage with conventional design. This is also the subject for future work.

Acknowledgement
This work has been carried out with support from Concept Engineering Company who made a donation for this research.

References

[1] Thompson, A.: Hardware Evolution: Automatic Design of Electronic Circuits in Reconfigurable Hardware by Artificial Evolution. D.Phil. thesis, University of Sussex, Brighton, Sussex, England (1996)

[2] Koza J.R., Bennett III F.H., Forrest H, Lohn J, Dunlap F., Andre D., Keane M.A.: Automated synthesis of computational circuits using genetic programming. In: 1997 IEEE Conference on Evolutionary Computation, pp. 447–452 Piscataway, NJ: IEEE Press (1997)

[3] Lohn J.D., Colombano S.P.: Automated Analog Circuit Synthesis using a Linear Representation. In: The 2nd International Conferene on Evolvable Systems: From Biology to Hardware, pp.125- 133 Springer-Verlag, Berlin (1998)

[4] Zebulum R.S., Pacheco M.A., Vellasco M.: Comparison of different evolutionary methodologies Applied to electronic filter design. In: IEEE Conf. on Evolutionary Computation, pp. 434-439 Piscataway, NJ: IEEE Press (1998)
[5] Goh C., Li Y.: GA automated design and synthesis of analog circuits with practical constraints. In: The Congress on Evolutionary Computation, Vol. 1, pp. 170-177 (2001)
[6] Ando S., Iba H.: Analog Circuit Design with a Variable Length Chromosome. In: Congress on Evolutionary Computation, pp.994-100 IEEE Press (2000)
[7] Grimbleby J.B.: Hybrid genetic algorithms for analogue network synthesis. In: Congress on Evolutionary Computing (CEC99) pp. 1781-1787 Washington USA (1999)
[8] Fan Z., Hu J., Seo K., Goodman E., Rosenberg R., Zhang B.: Bond Graph Representation and GP for Automated Analog Filter Design. In: E. Goodman (ed.) 2001 Genetic and Evolutionary Computation Conference Late-Breaking Papers, pp.81-86 ISGEC Press, San Francisco (2001)
[9] Wang F., Li Y., Li L., Li K.: Automated analog circuit design using two-layer genetic programming. In: Int. J. on Applied Mathematics and Computation, Special Issue on Intelligent Computing Theory and Methodology Vol. 185, Issue 2, pp. 1087-1097 (2007)
[10] Hu J., Zhong X., Goodman E.: Open-ended robust design of analog filters using genetic programming. In: Genetic & Evolutionary Computation Conference (GECCO) pp.1619-1626 ACM New York, NY, USA (2005)
[11] Dastidar, T.R., Chakrabarti, P.P., Ray, P.: A synthesis system for analog circuits based on evolutionary search and topological reuse. In: IEEE Trans. on Evolutionary Computation, Vol. 9, Issue 2, pp. 211 – 224 (2005)
[12] Sripramong, T., Toumazou, C.: The invention of CMOS amplifiers using genetic programming and current-flow analysis. In: IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 21, Issue 11, pp.1237 – 1252 (2002)
[13] Zebulum R., Stoica A., Keymeulen D.: Experiments on the Evolution of Digital to Analog Converters. In: IEEE Aerospace Conference, Big Sky, Montana, USA Manhattan Beach, CA ISBN: 0-78-3-6600-X. (Published in CD) (2001)
[14] Hu J., Zhong X., Goodman E.: Open-ended Robust Design of Analog Filters Using Genetic Programming. In: Genetic & Evolutionary Computation Conference (GECCO) Vol. 2, pp.1619-1626 ACM Press, Washington, DC (2005)
[15] Kuo T. and Hwang S.-H.: Using disruptive selection to maintain diversity in genetic algorithms. In: Appl. Intel. 7, pp.257–267 (1997)
[16] Brameier, M.: On Linear Genetic Programming. In: PhD thesis, University of Dortmund, Dortmund, Germany (2004)
[17] Vesselin K., Miller J.: The advantages of landscape neutrality in digital circuit evolution. In: International Conference on Evolvable Systems (ICES), Lecture Notes in Computer Science, pp. 252-263 Springer (2000).
[18] Thompson A.: Artificial evolution in the physical world. In: Gomi (Ed.) Evolutionary Robotics, AAI Books (1997)
[19] Sapargaliyev Y., Kalganova T.G.: On Comparison of Constrained and Unconstrained Evolutions in Analogue Electronics on the Example of "LC" Low-Pass Filters. In: IEICE transactions on Electronics Vol.E89-C No.12 pp.1920-1927 (2006)