Open-Ended Evolution to Discover Analogue Circuits for Beyond Conventional Applications

Yerbol A. Sapargaliyev, Tatiana G. Kalganova

Abstract Analogue circuits synthesized by means of open-ended evolutionary algorithms (EA) often produce unconventional designs. However, often such circuits are highly compact and the general nature of the evolutionary search methodology allows such designs to be widely applied. Previous work on evolutionary design of analogue circuits has focused on circuits that are lie well within analogue circuit design. In contrast, our paper considers the evolution of analogue circuits for designs that are usually synthesized in digital logic. We have evolved four computational circuits (CC), two voltage distributor circuits (VDCs) and a time interval meter circuit (TIMC). The approach, despite its simplicity, succeeds over the design tasks owing to the employment of substructure reuse and incremental evolution. Our findings expand the range of applications that could be considered suitable for evolutionary electronics.

Index Terms—**analogue, circuit, synthesis, CAD, SPICE, simulation, evolutionary algorithms, system-on-a-chip.**

I. Introduction

Evolvable Hardware (EHW) refers to the use of evolutionary algorithms for designing hardware. When the hardware in question is an electronic circuit the research subfield is often referred to as *Evolutionary Electronics* [1]. The evolutionary design of electronic circuits contrasts strongly with conventional design. Typically, little conventional knowledge is used and EE uses a generate-and-test methodology.

EA are guided by numerical evaluations of circuit performance known as fitness. New candidate designs are generated through selection, recombination and random alteration (mutation). This method of design is often less dependent on the personal knowledge of designers and partly as a consequence, produces unconventional designs. A variety of evolutionary algorithms and approaches have been used in Evolutionary Electronics. While genetic algorithms (GAs) remain popular, forms of genetic programming (GP) and evolutionary strategy (ES) have produced some outstanding results. ES was first introduced in [2]. The ES can be a very simple search algorithm and use a population of two (as in 1+1-ES) and only mutation to generate new candidates. The second column of Table 1 gives information on types of EA used in other published work.

One of the main aims of this paper is to investigate the potential of open-ended evolutionary circuit synthesis for the design of analogue circuits in applications that conventionally belong to the digital circuit domain. The functions we set as targets for evolution are typically performed by digital circuits that deal with analogue signals at their inputs and outputs, and usually incorporate analog-to-digital converters and digital-to-analogue converters in their structures. The high level of complexity of the targeted functions makes it reasonable to expect the resulting circuits to be large in scale. The last argument sets an additional requirement on the capacity of the methodology to tackle the scalability problem. For instance, Koza *et al* evolved the analogue circuits that perform digital functions: the NAND circuit and a *two-instruction arithmetic logic unit circuit* [44]. The evolved circuits were modest in size (6 and 26 components after pruning) with average functioning accuracy 4% and 10%, but took enormous computing efforts (2,2mln and 43,6mln evaluations respectively) and required such the methods to tackle the scalability problem as *substructure reuse* and parallel computing.

The first four targets are CCs whose primary tasks are conversion of incoming voltages into one of four computational functions: cube root, cube, square root and square. Analogue CCs can suggest a limited number of mathematical functions instead of the redundant repertoire of digital logic based CC. CCs were previously designed in [3]-[6].

The fifth and sixth targets are 4-output and 8-output VDCs that have one input and multiple outputs. The essence of VDC becomes easier to understand if we look at a single-source divergent neuron that has one dendrite and many axons with similar functionality [7]. The work of a single-source divergent neuron includes not only transporting the same signal from a single source to different locations, but also disintegrating the incoming signal and distributing the result among the outputs[1]. Analyzing the analogue choice for VDCs before the digital one, we must mention that in natural neural network, all (graded and impulse) signals are essentially analogue [8]. Moreover, most of the up-to-date industrial

Manuscript received December 23, 2010.

Yerbol Sapargaliyev is with the School of Engineering and Design, Brunel University, Uxbridge, Middlesex, UB83PH, UK (+7 705 6112211; yerbol.sapar@brunel.ac.uk)

Tatiana Kalganova is with the School of Engineering and Design, Brunel University, Uxbridge, Middlesex, UB83PH, UK (tatiana.kalganova @brunel.ac.uk).

[1] The procedure is in common with a well-known convergent neuron that integrates different signals from different locations into one [7], but has backward direction in case of signals' distribution.

sensors receive stimuli and transduce them into electric potentials purely in analogue form.

Finally, the seventh target for evolution is TIMC. TIMC is targeted to function in a single analogue mode instead of a number of digital operations inside the up-to-date laser rangefinder DAQ-2 [38], where the time interval metering function is performed by the several digital circuits when the rangefinder uses a laser beam to determine the distance to an object. The targeted analogue TIMC belongs to a class of devices that are known as *time-to-amplitude converters*. Time-to-amplitude converter generates a rectangular output pulse whose peak amplitude is linearly proportional to the time interval between a START and STOP input pulse pair" [42]. To our knowledge, this is the first attempt towards automatic synthesis of a time-to-amplitude converter circuit.

The work in this paper proposes and investigates an open-ended ES-based analogue circuit synthesis tool to: a) design unconventional nonlinear, multi-output and time-dependent functioning analogue circuits, b) design analogue circuits that are potentially able to replace digital circuits in applications that are conventionally adopted by digital logic, c) find much more efficient designs than was done by previous techniques.

The next section overviews the previous work in the area. *Section III* introduces the seven problems in more detail. *Section IV* describes the whole evolutionary technique. *Section V* presents the experimental results. The last section concludes the paper.

II. Previous Work

Open-ended methods of evolutionary analogue circuit synthesis have been challenged with an important question [9], "Are such methods able to create solutions that, when realized in silicon, are valid and trustworthy enough? In [10], the set of experiments have proven that the open-ended techniques enable the design of low/high-pass filters with topology-based robustness. In [11], the frequency-discriminator robust to wide temperature range was evolved with an open-ended GA, intrinsically in FPGA. [12] describes experiments that allowed adaptive in-situ circuit reconfiguration in extreme temperature and radiation environments. In [13], the unconstrained evolution successfully created analogue variability-tolerant CMOS circuits performing XOR and XNOR functions. The literature review on that subject allows distinguishing two approaches. The first traditional one follows the paradigm wherein the evolution is initially set to discover the unconventional design, and the circuit is later tuned to improve the robustness ([1], [11], [12], [14] and [15]). Another approach suggests the evolutionary system that is originally targeted for the robust designs ([9], [10] and [13]). In the current study, we adopt the first approach focusing on the exploration of the technique's capabilities to create novel designs, leaving the evolution of robustness for the next stage.

In the past, low-pass filters [1], [10], [14]-[20], high-pass filters [1], [10], [15], [17], [21], [22], amplifiers [11], [14]-[16] and CCs [3]-[6] were successfully synthesized with the help of EA. In [23], unconstrained evolution, both spatially and temporally was applied towards a digital reconfigurable hardware - FPGA. By releasing the full repertoire of behavior that FPGA can manifest in, namely, allowing any connections among modules and allowing the evolution of granularity and synchronization, evolution had been able to find a highly efficient electronic structure, which required 1-2 orders less silicon area to achieve the same performance as the conventional design. Natural behavior of analogue components started to be exploited inside a digital circuit. As an analogy to this approach, the unconstrained evolution in our previous endeavor to sharpen our technique was applied in [20] towards the original analogue circuits, and excellent results were obtained (low-pass filters).

Studies [1] and [17] gave the comparison between GP and GA. The first study was an analogy to a biology concept with a comparison of different types of variable length genotype strategies, whereas the second one was an intrinsic evolution of a real hardware for robustness purposes. According to [1], the "genotype length varying strategies" refer to the way in which the chromosome's lengths are sampled by the EA at each generation. It is easy to follow this idea if one looks at the sizes of the best circuits throughout generations. If the size at each following generation does not decrease, then it is the increasing length genotype (ILG), otherwise it is the oscillating length genotypes (OLG). The OLG strategy is a kind of ILG in which the genotypes are also allowed to decrease in size. The main purpose of OLG is to create pathways from larger to smaller genotypes with improved fitness values. The fifth column of Table 1 summarizes the length genotype strategies utilized previously, including fixed, OLG and ILG.

Table 1 summarizes the foremost related literature in analog Evolutionary Electronics. Most of the studies in the table focus on such circuits like filters and amplifiers (column 2) which, we think, is not an adequate enough challenge for the probation of up-to-date evolutionary techniques. Therefore, in this paper, we decided to target much more complex analogue circuits, some of which have never been designed before. The choice of target for the synthesis is based on the open-ended nature of our technique because we assume that the methods with constraints [24]-[28] and with pre-specified substructures [9], [22], [26], [27], [29], [30] are likely to be limited and after synthesis result only in conventional applications of analog circuits.

As can be seen from Table 1, the largest circuit evolved after pruning was made by Koza et al. [24] in 1997. One of the reasons why the 14-year old work is still on top is the powerful computing support, multi-cluster system that enabled them to operate the largest population (1,320,000 [44]) and the highest number of individual evaluations (6,700,000,000 [98]) in the EHW domain.

On the other hand, based on the overall aim of the paper, our interest covers those analogue circuits that are of unconventional applications. This limits the search to the most relevant works of Thompson [23] and Koza [44] (the last column of Table 1). Other works have neglected to evolve this category of circuits.

To tackle the scalability problem, and according to [24],

TABLE 1
DEVELOPERS IN EXTRINSIC EVOLUTION OF MOST COMPLEX ANALOGUE CIRCUITS EVOLVED

Researcher	Circuit name	EA type	Extrinsic/ Intrinsic	GLVS	Substruc-ture reuse	Scal. Method	Pruning	Input/ out-put No	Circuit size	Attempts to evolve circuits that perform digital functions
Koza *et al* [6], [44]	Square root	GP	Extrinsic	ILG	Yes	D&C	After pruning	3/3	64	NAND, 2-Instruction Arithmetic Unit
Thompson [23]	Tone discriminator	GA	Intrinsic	ILG	No	No	After pruning	1/1	20 FPGA cells	Tone discriminator
Mattiussi *et al* [24]	Temperature sensing	GA	Extrinsic	OLG	Yes	Dev.	After pruning	2/1	55	No
McConaghy *et al* [9]	Amplifier	GP	Extrinsic	ILG	Yes	M-O	After pruning	2/1	48	No
Sripramong *et al* [26]	Amplifier	GP	Extrinsic	Fixed	Yes	Represent.	After pruning	1/1	41	No
Shibata *et al* [27]	Absolute function	GA	Intrinsic	Fixed	Yes	Represent.	Before pruning	1/1	36	No
Trefzer *et.al* [45]	Amplifier	ES	Intrinsic	Fixed	No	M-O	After pruning	2/1	34	No
Layzel [49]	Oscillator	GA	Intrinsic	ILG	No	No	After pruning	1/1	33	No
He *et al* [50]	Amplifier	DE	Intrinsic	ILG	No	Represent.	After pruning	1/1	28	No
Hu *et al* [19]	Low-pass filter	GP	Extrinsic	ILG	Yes	Dev., M-O	After pruning	1/1	26	No
Lohn *et al* [15]	Low-pass filter	GA	Extrinsic	ILG	No	Represent.	After pruning	1/1	23	No
Ando *et al* [17]	Low-pass filter	GP,GA	Extrinsic	n/a	No	No	Before pruning	1/1	22	No
Kruiscamp *et al* [30]	Amplifier	GA	Extrinsic	Fixed	Yes	No	After pruning	1/1	22	No
Zebulum *et al* [1]	Amplifier	GP,GA	Extrinsic	3M	No	Represent.	After pruning	2/2	19	No
Xia *et al* [28]	Low-pass filter	GA	Extrinsic	Fixed	No	Adaptation	After pruning	1/1	18	No
Dastidar *et al* [29]	Amplifier	GA	Extrinsic	OLG	Yes	No	After pruning	2/1	18	No
Chang *et al* [22]	Low-pass filter	GP	Extrinsic	UDIP	Yes	Represent.	After pruning	1/1	17	No
Das *et al* [32]	Low-pass filter	GA	Extrinsic	UDIP	No	Represent.	After pruning	1/1	15	No
Ohe *et al* [51]	Amplifier	GP	Extrinsic	ILG	No	M-O	After pruning	1/1	15	No
Langeheine, *et al* [41]	Amplifier	ES	Intrinsic	Fixed	No	M-O	After pruning	2/1	15	No
Yuan *et al* [46]	Amplifier	DE	Extrinsic	ILG	No	Represent.	After pruning	1/1	14	No
Conca *et al* [47]	Low-pass filter	GA	Extrinsic	ILG	No	Dev.	After pruning	1/1	14	No
Kim *et al* [10]	Low-pass filter	ES	Extrinsic	ILG	No	Co-evolution	After pruning	1/1	14	No
Wang *et al* [25]	Amplifier	GP	Extrinsic	ILG	Yes	D&C	After pruning	1/1	13	No
Goh *et al* [16]	Amplifier	GA	Extrinsic	ILG	No	No	After pruning	1/1	12	No
Grimbleby [18]	Low-pass filter	GA	Extrinsic	ILG	No	No	After pruning	1/1	10	No
Sabat *et al* [48]	Amplifier	DE	Extrinsic	ILG	No	Represent.	After pruning	2/1	10	No
Gan *et al* [21]	Low-pass filter	GP	Extrinsic	OLG	No	Represent.	After pruning	1/1	7	No
This paper 1	**Square root**	**ES**	**Extrinsic**	**OLG**	**Yes**	**No**	**After pruning**	**1/1**	**22**	**No**
This paper 2	**Squaring**	**ES**	**Extrinsic**	**OLG**	**Yes**	**No**	**After pruning**	**1/1**	**35**	**No**
This paper 3	**Cube root**	**ES**	**Extrinsic**	**OLG**	**Yes**	**No**	**After pruning**	**1/1**	**39**	**No**
This paper 4	**Cubing**	**ES**	**Extrinsic**	**OLG**	**Yes**	**No**	**After pruning**	**1/1**	**44**	**No**
This paper 5	**4-out VDC**	**ES**	**Extrinsic**	**OLG**	**Yes**	**No**	**After pruning**	**1/4**	**51**	**4-output VDC**
This paper 6	**8-out VDC**	**ES**	**Extrinsic**	**OLG**	**Yes**	**D&C**	**After pruning**	**1/8**	**138**	**8-output VDC**
This paper 7	**TIMC**	**ES**	**Extrinsic**	**OLG**	**Yes**	**D&C**	**After pruning**	**2/1**	**87**	**TIMC**

GLVS is for the genotype length varying strategies; UDIP is for Uniformly Distributed Initial Population method; 3M is for 3 variable length genotype methods: ILG/OLG/UDIP; Dev is for Development; Scal. Method is for method applied to solve the scalability problem; D&C is for "divide and conquer" method; DE is for Differential Evolution; M-O is for multi-objective evolution.

"designers have introduced various approaches that can be divided into three classes: functional level evolution, incremental evolution (divide and conquer) and development." In this work we utilize the functional level and incremental evolution.

Improving the system at the EA level by exploiting the novel techniques, such as substructure reuse, novel representations, multi-objectiveness, co-evolution and adaptation, is most frequently used one (columns 6 and 7 of Table 1).

"Divide and conquer" approach is regarded as one of the most effective techniques against the scalability problem. Since the basic idea was first introduced by Torresen in [139], many approaches have been developed in the digital domain (i.e. [33], [52], [53]). However, the approach where the task is decomposed into subtasks and then the subtasks are evolved independently and then joined together, doubtfully can help in evolution of analogue circuits. Therefore, for the second option, few works have distinctly utilized this approach in the analogue domain [6], [25] (column 7). This comes from the physical nature of the electronic components that interact with each other by means of potentials and currents. This situation

differs from that of digital circuits, where the rules of Boolean algebra and the complex task could be decomposed by Shannon's expansion theorem or output decomposition [33]. As a fact, analog sub-solutions cannot be easily connected to get the proper functioning solution. That is, two perfectly working analog circuits, when connected to a common input, are not guaranteed to perform in the same way, and it is more likely that each circuit will disturb the functioning of its neighbour. In this regard, the incremental (or *staged evolution* [54]*)* is found to be most suitable, where the evolution of the current subtarget should be done together with all the subtargets evolved previously. That is, if one has the already evolved a subsolution, when evolving the second one the first subsolution must participate in that evolution, being encoded in the chromosome together.

The first four targets in current research are devoted to CCs, which are one of the most provoking issues for any automatic circuit synthesis system. It should be mentioned that for a decade, the largest analogue circuit evolved in the area of Evolutionary Electronics has been a square root circuit with 64 components in [6]. In [3], [4] and [6], they used Koza's GP circuit-constructing program trees with four kinds of functions. They also used automatically defined functions, and potentially enabled certain substructures to be reused. The paper [6] suggests an attractive opportunity to judge the effectiveness of the evolutionary tool. Targeting the same arithmetic functions, and utilizing an identical evaluation procedure (fitness function), one can directly compare the fitness values (average error), circuit size (economy), and PC time spent. In this paper, we took advantage of this opportunity. In [3], two CCs were developed by a similar evolutionary technique as in [6]; however, they used time-continuous signals in time-domain simulations. The transient analysis of a circuit in contrast to DC-analysis provides more robust circuits despite the higher time-consumption to complete the analysis. The patent in [5] presents the conventionally designed cubing CC, that was improved in [4] by the iterative refinement method. Both are compared in Section V.

Based on a literature survey, the previously reported designs were found to be analogue circuits with maximum four outputs (column 9, Table 1). In contrast, with an evolutionary design of digital circuits, we could not find a similar example in the analogue domain, where the scalability problem caused by numerous inputs and outputs is tackled ([33]-[35]). One reason is that the analogue circuit with multiple outputs is often considered to perform complicated signal processing, thereby providing a package of high-ordered output signals (in time) to the outputs. The size of such a circuit may include hundreds of components thus leading to a vastly expanded search space. Thus, as challenges for the proposed method, we took VDCs and TIMC in target cases 5, 6 and 7, and utilized incremental evolution[2].

[2] We regard incremental evolution as a case of more general "divide-and-conquer" approach, the essence of which lies in decomposing the target into subtargets.

III. INTRODUCTION TO THE PROBLEMS

A. *Computational circuits*

Computational circuits (CC) apply a computational function to an incoming voltage. In this paper, we try to evolve CCs that perform the following computational functions: cube root, cubing, square root and squaring. Fig.1 gives an idea about how an analogue CC can replace a set of digital logic.

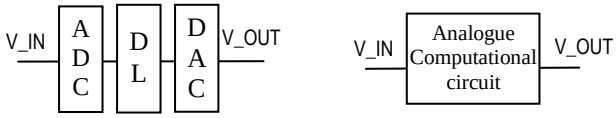

Fig. 1. A digital (left) and an analogue representation of a computational circuit.

B. *Voltage distributor circuits*

The conventional method of circuit design could easily model a neuron by utilizing the up-to-date digital signal processing units, such as controllers supplemented by analog-digital and digital-analogue converters.

However, in comparison with digital circuits, analogue circuits are faster. This is because asynchronous circuits are not constrained by an arbitrary clock. Instead, they are only limited by physical and electro-magnetic interactions. Furthermore, when compared with digital circuits, analogue circuits are economical both in power and silicon footprint [37].

This issue becomes especially important if the difference in components between competing circuits reaches multiples of a hundred, such as can be the case in implementations of neural networks where the number of units (neurons) tends to be very large. Fig. 2 gives a general view of a neuron model consisting of three digital circuit blocks. Our aim is to replace all three units by a single analogue circuit.

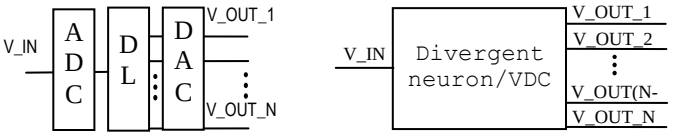

Fig. 2. A digital (left) and an analogue representation of a one-input multi-output voltage distributor/divergent neuron circuit.

Knowledge about neurons mostly concerns those convergent ones that integrate multiple signals from dendrites into a single signal to an axon. Divergent neurons are not as widespread in natural neural systems. This last fact is due to the convergent nature of neural network, which is mostly caused by a vast diversity of receptors that sense stimuli at a molecular level. That is, any stimuli comes into a natural neural network at such a fine-grained level that the networks are left only with converging the mosaic into the pictures, thus solving higher-level intelligent tasks like cognition. On the other hand, most of the up-to-date industrial sensors do not possess such a feature. Thus, practically, it is reasonable to target a circuit that simulates a divergent neuron that has the ability to disintegrate the incoming voltages from sensors and distribute them among multiple outputs. Hence, the circuit is called the voltage distributor (VDC).

The disintegration task for each output of VDC involves working in filter-like mode that passes the input signal located within a particular voltage band, without any change in the form of the signal. For 4-out VDC, the band-pass for each output equals *5V/4=1.25V*: the first output passes the voltages from 0 to 1.25V, the second from 1.25-2.5V, the third from 2.5-3.75V and for the fourth, the band-pass is 3.75-5V. Fig. 3a demonstrates separately the transient analysis of every pin of the targeted 4-out VDC. A similar situation exists with an 8-out VDC, where the band-pass for each of the outputs equals 0. 625V; that is, the band-passes for all 8 out-pins are: first (0-0.625V), second (0.625-1.25V), third (1.25-1.875V), fourth (1.875-2.5V), fifth (2.5-3.125V), sixth (3.125-3.75V), seventh (3.75-4.375V) and finally, eighth (4.375-5V). The united transient analysis at input and eight outputs of the targeted 8-out VDC is presented in Fig. 3b.

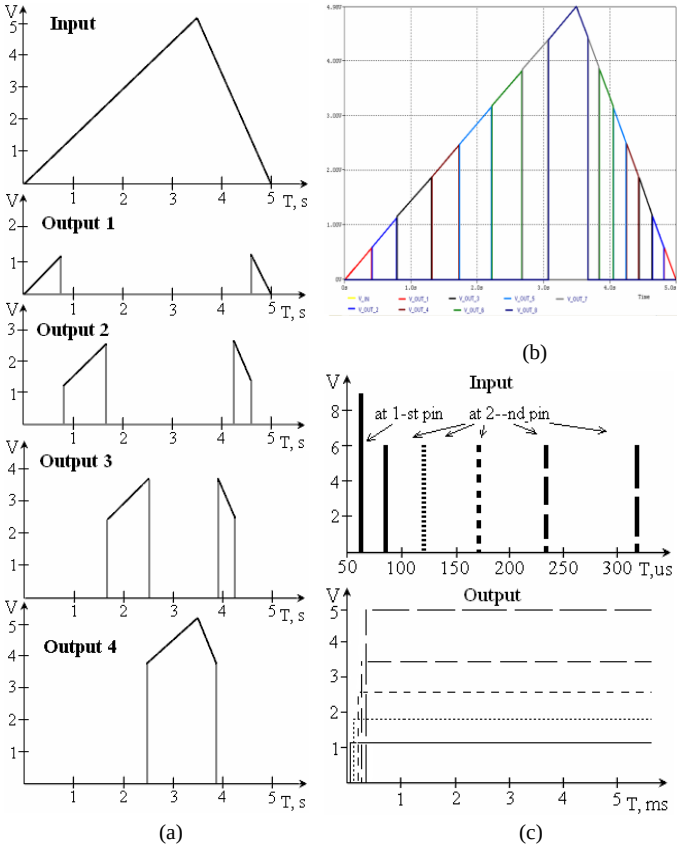

Fig. 3. (a) Transient analysis of potentials at input and four outputs of the targeted 4-out VDC. (b) The united transient analysis of potential at input and eight output pins of the targeted 8-out VDC; (c) The top graph shows 2 pulses at 2 inputs of the TIMC (both are of 50ns width): 1-st is 9V at 60us, for the 2-nd we took 5 arbitrary pulses at 85, 120, 170, 230 and 333,4ns. These coupled signals correspond to distances 25, 36, 51, 69 and 100km. The bottom graph shows 5 transient replies at the output pin of an ideal TIMC.

For both VDCs, we were unable to trace any existing device or published work that described an analogue or digital circuit performing a similar task. The last fact gave us an alluring opportunity to challenge the potential of the evolutionary technique.

C. TIMC for the laser rangefinder

A laser rangefinder is a device that uses a laser beam to determine the distance to an object. The most common laser rangefinder operates on the *time of flight* principle by sending a laser pulse in a narrow beam towards the object, and measuring the time taken by the pulse to be reflected off the target and returned to the sender [38]. The distance is given by:

$$S = \frac{cT}{2}, \qquad (1)$$

where ***c*** is the speed of light and ***T*** is the amount of time for the round-trip between the device and the target. A typical laser rangefinder has two main parts: optical and electrical. The optical block sends the laser beam and receives the reflection, providing the electrical block with two voltage pulses, based on which the electrical block calculates the distance.

As a prototype, we took the artillery quantum rangefinder DAQ-2 [38] with the following data:

- working range is 0.2÷100km,
- measurement accuracy is 6÷30m,
- width of both pulses is 50ns; fall/rise time of the pulse is up to 5ns; first pulse has 9V amplitude; reflected pulse has 6V.
- power supply required is 29V.

The main part of the electrical block of a laser rangefinder is a time-interval meter sub-block. The working principle of a conventional time interval meter sub-block consists of three functional stages.

1) At the first stage, two electrical pulses received from an optical block are reshaped into the voltage gate pulse, where the first incoming pulse is caused by the laser beam sent towards a target, and the second one is caused by the beam reflected off the target. The gate pulse is a pulse of some constant potential that should have the same time-width as the interval between the two narrow pulses caused by a laser beam.
2) At the second stage, the gate pulse (i.e., the time interval of the gate pulse) is filled up by clock signals from the crystal oscillator. According to (1), the gate pulse width is varying from about 0.667us for the minimum measured distance 0.1km to 0.667ms for the maximum measured distance 100km.
3) And finally, the number of pulses contained in the packet is counted. The result of counting in binary code should be sent to a decoder for further conversion into decimal code.

Fig. 4 is a general schematic of the time-interval meter sub-block of the up-to-date laser rangefinder. Based on the description available to public, we set a goal to synthesize the analogue circuit that is able to unite stages 1), 2) and 3) described above by receiving two pulses from an optical block and producing the particular constant voltage. The linear correlation between time gap and the voltage produced is set, ranging between the maximum 5V (against the maximum 100km) and 0V (for a distance 0km). The targeted time-interval meter sub-block based on an analogue circuit is shown

in Fig. 5. The targeted TIMC replies to arbitrary incoming pulses are shown in Fig. 3c.

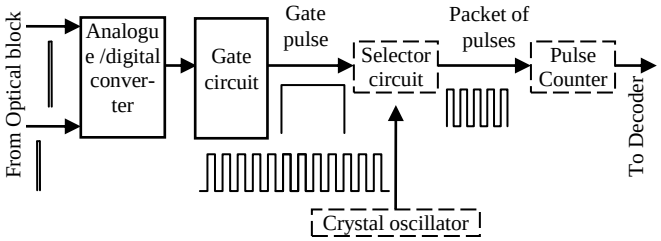

Fig. 4. The time interval meter sub-block of an up-to-date laser rangefinder made of digital logic. The shapes of the signals are shown under each pin. From left to right: there are two pulses coming in from an optical block, 9V and of 6V, separated by a time taken for the beam to be reflected and returned; they are converted to a digital form by ADC they are transformed to a gate pulse by gate circuit; a selector circuit fills up the gate with clock pulses from a crystal oscillator; a pulse counter circuit gets the packet of pulses from a selector circuit and counts the clock pulses; a decoder converts that count to a decimal form.

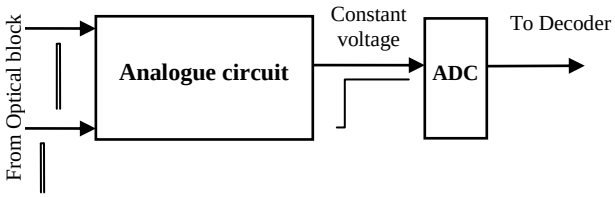

Fig. 5. The proposed time interval meter sub-block with the targeted analogue circuit. The shapes of the signals are shown under each pin. From left to right: two pulses are converted into constant voltage; the voltage level is in linear proportion to the time interval between two pulses; the ADC converts the voltage into the binary code for further decoding. Due to the resolution of the circuit is preferred to be at least 50uV (corresponds to 1 meter), that is totally 1e+5 discrete values, the 18-bit ADC with 262144 quantization levels will meet the requirement.

Thus, the first four targets are interesting since they provide the chance to compare with previously evolved designs, and judge the potential of the approach. The fifth and sixth targets are interesting since they represent the multi-output class of analogue circuits, which is always a challenge for any automatic synthesis tool. And, finally, the last target is chosen as it represents the real world problem, where the targeted solution is customized for a particular application.

IV. The Evolutionary Technique

A. *Encoding and embryo circuits*

We use three types of components for the synthesis of computational circuits: Qn – the n-p-n bipolar transistor, Qp – the p-n-p bipolar transistor, and R – resistor. We use an additional component capacitor (C) for TIMC, and we use two additional components for two VDCs: inductor (L) and capacitor (C). The linear (direct) circuit representation is proposed for use, similar to the one exploited in [1], where every component of a circuit is represented as a particular gene, and each gene consists of exactly four loci corresponding to the component's features: name, node number and parameter (except Q).

Thus, a targeted circuit is represented by a column of genes called the "chromosome" of that particular circuit. The genes in Fig. 6 look exactly the same as component lines in the PSPICE netlist; so, there is no necessity to convert a genotype into a netlist. This type of coding simplifies the terminology, for example, we mean "circuit" when we mention "chromosome", "component" when we mention "gene", "population" when we mention "netlist", and vice versa.

(a) (b)

Fig. 6. A gene coding a resistor (a) and a bipolar transistor (b): Rx-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 resistor's parameter.

The embryo circuit is a group of components (including a source(s) of input signal(s)) that is predetermined for a particular circuit. Mostly, these components are located at the circuit's inputs and outputs.

We defined the embryo circuit for all CCs in the same manner: a pulse voltage source, a source resistor Rsource=1kΩ and a load resistor Rload=1kΩ. These three components in Fig. 7a compose the embryonic circuit and are absolutely identical to the ones in most of the works in Table 1. The embryo also has two sources of direct voltage suggesting the evolution to choose between (or use both) +15V and -15V, so that the initial node number is five.

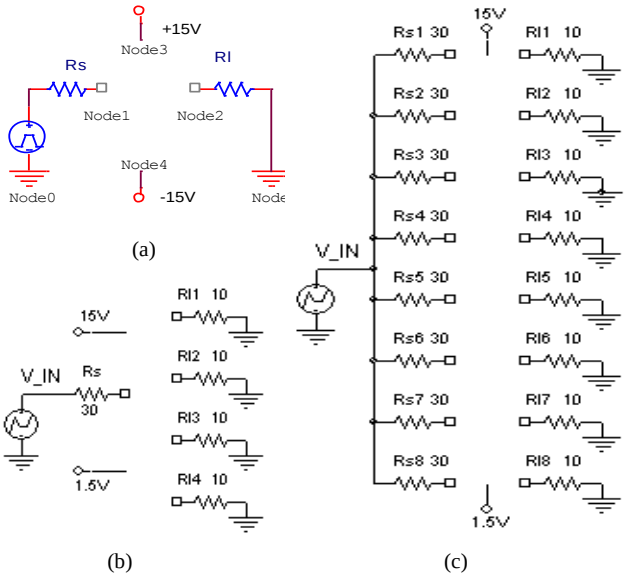

Fig. 7. (a) Embryo for the CCs; (b) Embryo for the 4-out VDC; (c) Embryo for the 8-out VDC.

In Fig. 7b and 7c, there is an embryo for 4-out VDC and 8-out VDC. The first embryo consists of a source of piecewise input signal (V_IN), source resistor (Rs) and four load resistors (Rl1…Rl4). The second embryo consists of a source of piecewise input signal (V_IN), eight source resistors (Rs1…Rs8) and eight load resistors (Rl1…Rl8). Both embryos can also have two sources of direct voltage suggesting the evolution to choose between (or use both) 15V

and 1.5V. As can be seen from the figures, the embryo for 8-out VDC has 8 parallel source resistors, while that for the 4-output circuit has only one. This difference is caused by the different techniques applied to the synthesis of the circuits.

TIMC of the laser rangefinder consists of two inputs with source resistors and one output with a load resistor (Fig.8a). The evolution is enabled to choose whether to use the direct voltage source +15V or not.

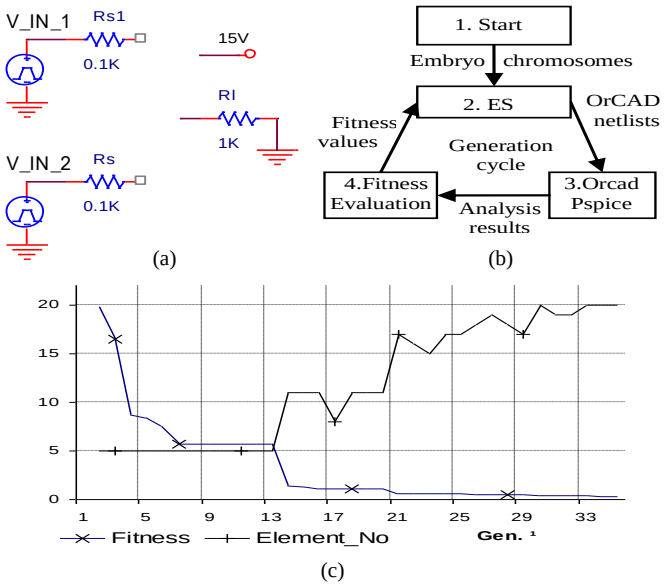

Fig. 8. (a) Embryo for the TIMC; (b) Flowchart of the experiment. (c) A chart fragments of the fitness value and the size of the best circuit vs. generation.

B. Unconstraining the evolution

The ability of unconstrained evolution to synthesize unconventional designs is well described in [11]. To reach our targets, we neither provided any pre-scribed substructures nor set any constraints for the whole process.

In [39], we introduced "the absolutely unconstrained evolution for analogue LCRQQ circuits", where no circuit-structure-checking rules were applied. We also did not prohibit the formation of *loops* of components during circuit growth. A *loop* is a component or a group of components aside from the main circuit that either does not connect to the main circuit or connects to it only via single node. *Loops* may not influence the functionality of the main circuit. However, they participate in carrying the neutral mutations [23].

To maximize the portion of circuits accepted by the simulation software, Transient analysis is applied to perform the evaluation instead of DC-analysis. This hint considerably decreases the number of invalid chromosomes and allows the multitude of individuals that could potentially be carrying the right structures, to pass on to the next generation. For more on unconstrained evolution, see the previous works [20],[39].

C. Program structure and mutation types

Fig. 8b shows the algorithm of the experiment. It consists of four main blocks that have been coded and united in one C++ program. The Start-block provides the population of embryo chromosomes in the form of PSPICE netlists to the ES block. At this stage, the embryo is cloned to the population number, and then, every clone is grown randomly with the help of starting components described in *Subsection A*. Due to the randomness of the process and the unconstrained rules according to which the growth takes place, only about half of the first population proves valid and can be simulated. However, the last chromosomes will be parents to an almost totally valid offspring (only 3-5% of invalid circuits per population during later generations).

The ES block contains and applies particular parameters of ES, such as: mutation rate, population size, selection criteria (fitness function) and termination terms. First of all, the ES block gets the whole population size *P*, with fitness value assigned to each chromosome from the previous block. According to the prescribed selection value, *S*% (usually from one chromosome to 50% of the total population), it chooses the best *S*% of the chromosomes as parents for the next generation. Then, the ES block clones each of the selected parents in the amount (*P**S/100) per individual that makes the population complete again.

After that, the mutation procedure is applied to every individual. There are totally six types of mutations, which are:

- Add_new_component_mutation (ANEM);
- Delete_component_mutation (DEM);
- Circuit_structure_mutation (CSM), that includes following three subtypes: component_name_mutation, component_pin_mutation, component_parameter_mutation;
- We also regarded the Substructure_reuse_mutation (SRM), in which the group of genes is modified during one procedure since it is just another way of chromosome modification.

Each mutation modifies *M*% of the total amount of a chromosome, where *M%* is the prescribed mutation rate (usually varying from 0 to 10%). Since each gene contains exactly four loci, the ANEM and DEM modify the chromosome by four loci, whereas CSM mutates per one locus. So, to mutate 5% of the chromosome with 20 genes (80 loci) means to modify only four loci, which is achievable by applying either one ANEM or one DEM procedure, or by applying CSM four times. ANEM and DEM also enable to regulate the development of the chromosome. For example, if the circuit is growing too fast without considerable improvement of its functionality (fitness value), then it is reasonable to launch the DEM; if the fitness growth has stuck, adding extra component to the circuit by ANEM will lead to the revival of the search. On the other hand, the CSM is more helpful in retaining the size of the growing chromosome. It is more devoted to search for a better solution within the given number of genes of the chromosome. It will mutate the circuit's structure, components' types and components' parameters generation by generating while the fitness stops improving.

And finally, the SRM comes into the scene when the multiple tries of ANEM and DEM do not show any effect on fitness improvement. SRM recalls fragments (substructures) of the best chromosomes from the past generations and joins them to the current circuit structures. To get this procedure working, the substructure database has to select, memorize, compare and replace substructures at each generation. Altogether, a CSM-ANEM-DEM-SRM combination enables

to keep the chromosomes' length in a population within a limited range, and enables the evolution to be focused on its search inside this range, while simultaneously allowing chromosomes to grow gradually. Fig. 8c represents the fragments of the circuit size and its fitness during evolution. The size of the circuit grows gradually from 5 to 20 components, while the fitness value of the best circuit falls down (improves). It can be seen that ANEM improved the fitness at generations 24,26,27,33; DEM worked out at 17,22,23,28,29,31; SRM improved the fitness at generations 14,21 and 30 with substructures consisting of 6, 6 and 2 components correspondingly. The rest of the evolution is ruled by CSM (1-13,15-16,19-20,25,32). In general, the behavior of the chromosome's length during evolution corresponds to "oscillating length genotype strategy" proposed in [1], where the chromosomes' length can grow as well as reduce.

Getting the cir-batch-file from Block 2 (ES block), the Block 3 starts PSPICE, simulates and saves the results in out-file. PSPICE is exploited in non-interactive batch simulation mode.

Block 4 contains the fitness function. It reads all chromosomes one by one from the out-file, evaluates them, assigns fitness, selects the best *S%*, ranges them over and sends the results to Block 2.

D. Substructure reuse mutation (SRM)

The substructure reuse method is well explained and widely instantiated in [36], where Koza uses direct encoding as well, but in a different representation (in the form of List software sub-programs). So the idea of sub-structures utilized in this study is in common with Koza's concept. In our case, as mentioned in Section II, the substructures could not be prescribed due to the nontrivial nature of the targets and, thus, were automatically created during the evolution. The effectiveness of SRM depends directly on the size of the substructures to be reused: the larger the substructure, the higher the mutation (number of loci) it brings to a chromosome. Since the junction points for a substructure inside a circuit are under the choice of a random process, the substructure with a large amount of components ***E1*** has more possible ways ($\boldsymbol{N}^{E1}$) to be connected to a circuit containing ***N*** nodes than the substructure with fewer amounts of components ***E2*** to be connected to the same circuit: $\boldsymbol{N}^{E1} > \boldsymbol{N}^{E2}$, where ***E1*** > ***E2***. Thus, larger the substructure, larger the population size it requires for containing enough diversity that two structures could bring by their junction. We used the limit for substructure size upto six components at maximum population size 30000 chromosomes. This followed from a series of experiments, which verified that at this point, substructures of larger size rarely help evolution.

As experiments reveal, this approach succeeded in all target cases except 8-output circuit and TIMC. In last cases, the problem was lack of computing power to evolve larger populations with longer chromosomes, which is known as "the scalability problem" [33]. As soon as the circuit reached 50-60 components in size, the fitness improvement stopped despite undertaking all types of mutation procedures. To tackle this issue, we utilized the incremental evolution.

E. Incremental evolution

As mentioned above, incremental evolution is introduced for the design of 8-output VDCs and TIMC. Each task is decomposed into subtasks, and evolved step by step in automatic mode. The design task and fitness function is incremented each time the current task is solved.

For an 8-out VDC, a total of eight subtasks corresponding to eight subcircuits are set. Each subcircuit is responsible to get an incoming signal and produce an output signal to its own output pin. If the first task is a design of the first subcircuit, the second task is a design of the first and second subcircuits, the third task is a design of the first, second and third subcircuits, and so on. Finally, the eighth task is a design of all eight subcircuits, that is, the whole VDC. The evolution starts from the first subcircuit and upon its completion, moves to the next one. The main advantage of such an approach is the possibility to start the evolution of the next subcircuit (i.e. the third) based on reuse of the previously evolved subcircuits (i.e. the first and second). Due to the similarity of functions that subcircuits perform independently, namely, to pass a particular voltage band and stop the rest, the evolution's task (except the first subcircuit) is just to reprocess the previously evolved subcircuits into a new subcircuit with its own pass band.

Thus, there are two types of substructure reuse that we imply in the frame of 8-out VDC. The first one mentioned in *Subsection D* is one of the mutations (SRM) to apply to chromosomes when they are stuck in a fitness growth. Another kind of reuse is applied during the incremental between the transition from one subcircuit to another. If the first type of substructure is limited to six components, the size of the second is unlimited; if the place for the first substructure is randomized, the place for the second one is definite: between the corresponding source and load resistors of embrio in Fig. 7c.

For TIMC, initially we tried to evolve the whole circuit at once without exploiting the task decomposition, but the evolution failed to converge towards an acceptable solution. Then the problem was divided into two subtasks. The first one was the evolution of two-input-one-output gate pulse producing subcircuit, and the next one was the evolution of a one-input-one-output subcircuit, which was in series with the first subcircuit.

Since both experiments ran non-stop throughout all the substages, the dynamic fitness function similar to "adaptive fitness schedule" was introduced from [15], that is, the fitness function was incremented "whenever the current fitness threshold is reached by at least one chromosome in a population".

F. Solving the generalization problem

The problem of generalization appears when the validity of the circuit function is limited only to a case of source signals used during evolution, and does not extend to arbitrary signals. In the proposed work, the problem of generalization has been solved by sampling the source signals.

For CCs, it is enough for the voltage source to form a pulse signal rising from –250 mV to +250 mV for the cube root, cubing and squaring, and from 0 mV to +500 mV for the square root. Thus, for the proper functioning of circuits it is sufficient to evolve them based on the simplest input.

For both VDCs as input signals, we take the piecewise asymmetrical form starting from 0V and going up to 5V for 3.5s, and down to 0V in the last 1.5s (the upper graph in Fig. 3a).

For TIMC, the situation is more complicated since instead of changing a signal form, we set a number of coupled incoming signals with different time intervals between them. At the first substage, for the gate pulse forming subcircuit, we succeeded with three coupled signals (corresponding to 0.5, 20 and 100km), which is the resulted subcircuit generalized for the rest of the cases. For the second subcircuit, initially, we tried the same three coupled signals, but the resulting circuit was unable to function in the case of an arbitrary distance. Then we tried five signals and failed again. Finally, we found out that the minimum number of input signals (pair of signals) should be six. The higher this number, the more precisely the circuit functioned for the rest of the cases. Thus, we used seven cases of coupled signals corresponding to distances 0.4, 2, 10, 30, 45, 65 and 95 km. This number means that every chromosome of a population at each generation is tested seven times for seven different incoming signals, and seven fitness cases are composed the final fitness value for that particular chromosome by a simple sum. Thereby, seven was chosen as a compromise value between precision and computing time.

G. Fitness Function

For all design cases, a fitness value is set to a sum over *p* fitness cases of the absolute weighted deviation between the target value and the actual output value voltage produced by the circuit:

$$F = \sum_{i=0}^{p} | V_{ideal}^{i} - V_{measured}^{i} |, \quad (1)$$

where V_{ideal}^{i} is the voltage obtained in the *i*th point for ideal response and $V_{measured}^{i}$ is the voltage obtained in the *i*th point for the evolved circuit. The p equals 21 time-points for CCs, 81 for VDCs and 11 for TIMC. The smaller the fitness value is, the closer the circuit is to the target. The fitness penalizes the output voltage by 10 if it is not within specified percent range of the target voltage value.

For TIMC, where the output from the circuit is supposed to be a constant voltage, all the eleven measured points are equidistant within a range of 1ms to 10ms, which is sufficient for the analog-to-digital converter (ADC) to catch up with the signal for further coding.

The dynamic fitness function is scheduled for subcircuits of VDCs and TIMC for each incremental substage as a simple sum of the fitness values of all subcircuits evolved at the time:

$F = \sum_{i=1}^{i=8} F^{i}$, where F^{i} is a fitness value of the subcircuit that can be calculated by (1) and where *i* equals 8 in a case of 8-out VDC and 2 in the case of TIMC.

We set the following condition as termination reaching criteria: the fitness value does not improve over 20 consecutive generations.

H. Experiment Implementation and ES parameters

The PSPICE default model and distribution parameters are used during all experiments for all kinds of components. For instance, the bipolar transistor parameters are: IS=0.1F XTI=3 VAF=∞ BF=100 NE=1.5 ISE=0 IKF=∞ XTB=0 BR=1 NC=2 ISC=0 IKR=∞ RC=0 CJC=0 MJC=.33 FC=.5 CJE=0 MJE=.33 TR=0 TF=0 ITF=.0 VTF=∞ XTF=0 RB=0, ABSTOL=1.0pA, RELTOL=0.001, VNTOL=1uV, ITL1=150, ITL2=20, etc. [43]. The OrCAD PSpice-10.3 is utilized as the simulation software in a non-interactive batchmode. All circuits are tested on output voltage (V_OUT) at load resistor Rl (Figs. 7 and 8a).

For inductors (from 1E-9H) and capacitors (from 1E-12F), there are 84 values, for resistors (from 1.8Ω) there are 96 values used by evolution. All values are of E-12 series, i.e. there are seven and eight decades corresponding to 12 parameters each.

The ES with linear representation and oscillating length genotype is utilized. We used different selection rates for different targets: 10% for CCs, 1% for VDCs and 0.2% for TIMC. The ES deserves the name 'simplest EA', because it does not require the crossover operation: all the offspring chromosomes are identical to a corresponding parent. A mutation rate of 5% is allowed to apply to each chromosome by one of the mutation instruments described in *Subsections C* and *D*.

A population size of 30,000 chromosomes is set. We use five PCs with Intel Core 2 Duo/2GHz processor running at the same time, independent of each other.

V. Experimental Results

A. The computational circuits

The results presented for CCs are the best out of five runs for each case with different seeds for the random number generator.

TABLE 2. STATISTICS FOR THE EVOLUTION OF THE 4 CCS

No	Fitness	Component No.	Generation No.
		Square Root	
1	0.283	43	119
2	**0.194**	**23**	**123**
3	0.443	50	208
4	0.798	38	97
5	0.255	50	200
		Squaring	
6	**0.0302**	**35**	**92**
7	0.0459	43	309
8	0.0563	48	143
9	0.0951	38	97
10	0.0776	50	135
		Cubing	
11	0.0095	50	195
12	0.0205	38	72
13	0.0079	49	109
14	**0.0061**	**44**	**78**
15	0.0101	37	98
		Cube Root	
16	0.764	44	115
17	1.060	49	179
18	**0.251**	**39**	**152**
19	0.268	50	201
20	0.643	40	294

The data for all 20 runs is presented by Table 2, where the best runs are marked in bold. The average time per run is 43 hours. The best-of-run circuit (Fig. 9) for the problem of designing a square root circuit has 23 components with fitness 0.194. The best-of-run circuit (Fig. 10) for the problem of designing a squiring circuit has 35 components with fitness 0.0302.

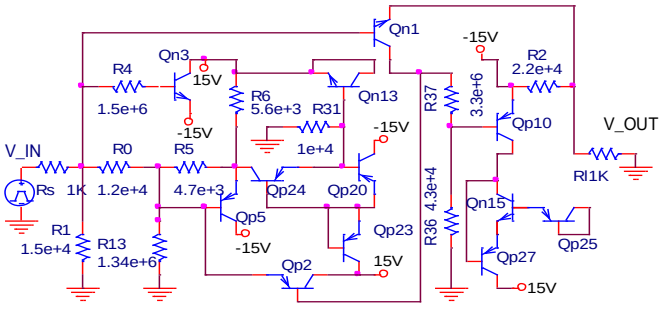

Figure 9. The Evolved Square Root Circuit

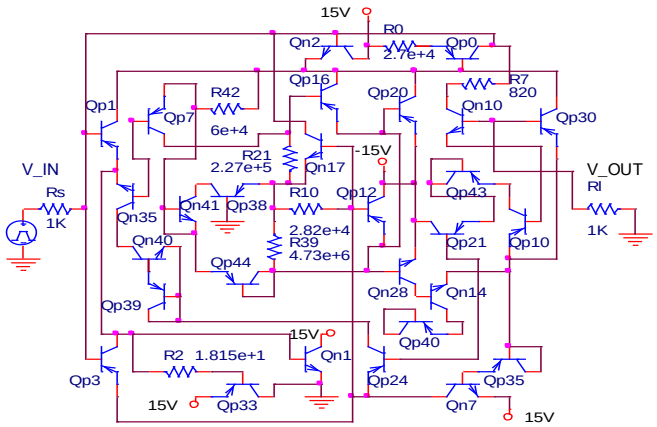

Figure 10. The Evolved Squaring Circuit

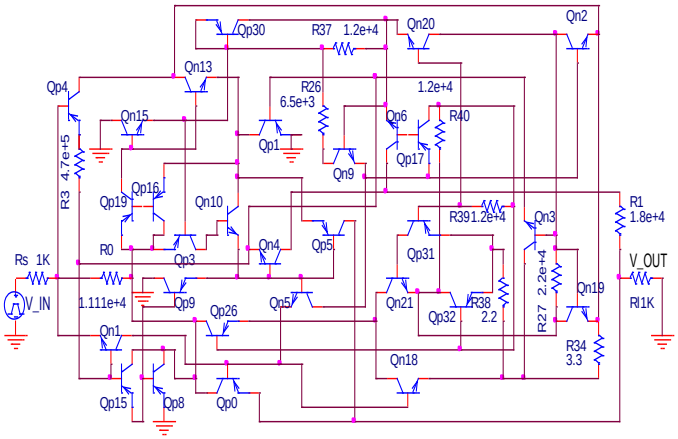

Figure 11. The Evolved Cube Root Circuit

The best-of-run circuit (Fig. 11) for the problem of designing a cube root circuit appeared at generation 152 and has 39 components with fitness 0.2508. The best-of-run circuit (Fig. 12) for the problem of designing a cubing circuit appeared at generation 78 and has 44 components with fitness 0.00614.

The schematics published in [3]-[6] enabled us to source-code them, analyze their netlists in PSPICE and get the fitness values that are appropriate for comparison. Both DC and transient analysis gave us identical results for each schematic, which, together with other published results, enabled us to aggregate all the data into Tables 3 and 4. The values in the column under the name "Improvement (times)" in Tables 3 and 4 are calculated by the formula: Improvement = min (value_from_work_[6], value_from_work_[3], etc.) / proposed_work_value. The values highlighted in bold are the best for each parameter among other studies.

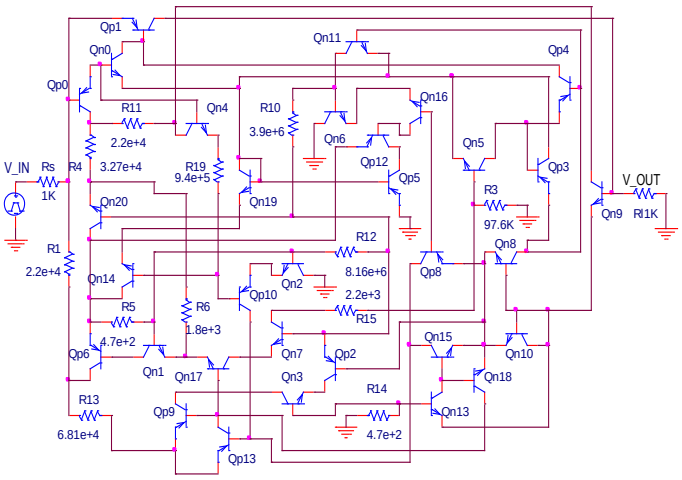

Figure 12. The Evolved Cubing Circuit

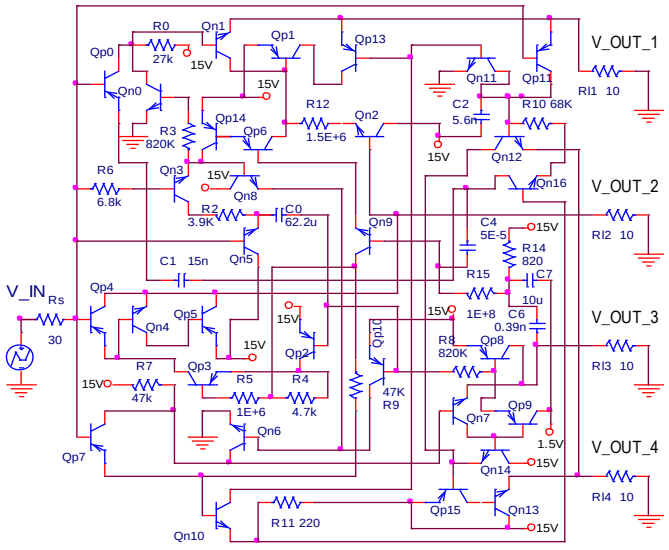

Fig. 13. The evolved 4-output Voltage Distributor.

We got exactly the same fitness values for some circuits from [6]. The last fact ensured that we chose proper transistor models (SPICE default models) and other simulation parameters. The extreme right column of the tables suggests the relative comparison between the value received in this paper and the best corresponding values from the past. As can be seen, for 15 out of 16 comparable positions, our results are considerably better. Notably, the best by size (12 components) conventionally designed cubing circuit from [5] has an average error (7.13mV) 25 times larger than that of the cubing circuit (44 components) evolved by us (0.29 mV).

TABLE 3. COMPARISON WITH CIRCUITS PUBLISHED BEFORE

Parameter \ Author	Koza *et al.* [6]	Mydlowec *et al.* [3]	Proposed work	Improve-ment (times)
Square root				
Average error, mV	183.57	**20.00**	9.23	2.2
Fitness value	**3.855**	70.403	0.194	19.9
Component No	64	**39**	22	1.8
Evaluation No	Data n/a	6,7E+9	3,7E+6	1800
Squaring				
Average error, mV	Data n/a	27.00	1.44	18.7
Fitness value	Not converged	4.812	0.0302	159.3
Component No	39	**37**	35	1.1
Evaluation No	Data n/a	1,1E+9	2,7E+6	407
Cube root				
Average error, mV	80.00	-	11.90	6.7
Fitness value	1.68	-	0.2508	6.7
Component No	50	-	39	1.3
Evaluation No	3.8E+7	-	4.5E+6	8.4

TABLE 4. COMPARISON WITH CUBING CIRCUITS

Parameter \ Author	Koza *et al.*[6]	Streeter *et al.*[4]	Cipriani *et al.*[5]	Propose d work	Improve-ment, times
Cubing					
Aver.error, mV	1.04	**0.99**	7.13	0.29	3.4
Fitness value	0.0219	Data n/a	Data n/a	0.0061	3.6
Component No	56	47	**12**	44	0.3
Evaluation No	Data n/a	2.94E+6	-	2.34E+6	1.3

B. VDC

Per one run has been made for each of the VDC. The evolution time of 4-out voltage distributor was 123 hours. The best-of-run circuit (Fig. 13) appeared at 120th generation and had 51 components (embryo excluded), among which were 14 resistors, 6 capacitors, 0 inductors, 16 NPN transistors and 15 PNP transistors, with fitness 0.38.

The aggregated transient response of the circuit, as can be seen from Fig. 14, almost exactly repeats the form of the incoming piecewise signal.

The evolution time of 8-out voltage distributor was 344 hours, which is about 43 hours per subcircuit.

The best-of-run circuit (Fig. 15) appeared at 629th generation and had 138 components (embryo excluded), among which were 38 resistors, 8 capacitors, 7 inductors, 46 NPN transistors and 39 PNP transistors, with the best overall fitness 1.757. Table 5 highlights the detailed information per incremental substage: the best fitness, the component number of the evolved subcircuit and the successful generation number.

The most ideal function with fitness 0.028 is produced by out-pin No.2 that is responsible for the band 0.625V-1.25V; the worst reply with fitness 0.797 is at the out-pin No.7 in a band 3.75V-4.375V. Fig. 16b shows the aggregated transient reply of the circuit for the incoming piecewise signal (Fig. 16a).

To verify that we overcome the problem of generalization, we applied different arbitrary signals to the resulted 8-output voltage distributor. Fig. 16c shows the piecewise signal that is

more complicated than the one applied during evolution, and Fig.16d shows the corresponding transient reply at the outputs. Fig. 16e and 16f show the arbitrary exponential signal and its transient response.

TABLE 5.
SUBCIRCUIT DATA FOR 8-OUT VOLTAGE DISTRIBUTOR

No	Fit-ness	Compo-nent No	Gen. No
1	0.095	10	76
2	0.028	22	132
3	0.174	16	110
4	0.323	23	37
5	0.049	14	26
6	0.200	23	107
7	0.797	22	104
8	0.089	8	37
Total	1.757	138	629

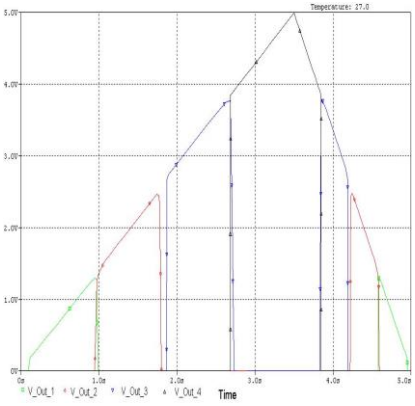

Fig. 14. The summary transient analysis at outputs of 4-out Voltage Distributor.

C. The time interval meter circuit

For the evolution of TIMC only one run is made. It took about a week to design the whole circuit, where 17% of the time was spent on the first subcircuit and the rest 83% on the second one. The first subcircuit with two inputs and one output, with the primary task to provide a gate pulse, consists of 31 components. The second subcircuit, with the task to accept a gate pulse and produce the required constant voltage, consists of 56 elements. The whole design consists of 87 components among which are: 29 resistors, 26 p-n-p transistors, 17 n-p-n transistors and 15 capacitors. The best fitness of the first subcircuit is 0.906, the final fitness of TIMC has even reached 1.137 at generation 64.

The PSpice's performance analysis enables us to measure the generalization ability of the circuit by tracing the dependence of circuit replies on a swept parameter. If we take the absolute average deviation from the ideal circuit response as a swept parameter, and apply it to a family of waveforms, we produce a trace that is a function of the variable that changed within the family. As can be noticed from Fig. 18b-c, which represents the absolute average deviation along 1000 equidistant circuit replies, the measurement accuracy of TIMC could be approximately split into three groups: 3m for distance range 0.1÷2.5km, 16m for 2.5÷15km and 54m for 15÷100km. In comparison with conventional digital TIMC, where the measurement accuracy varies within the range 6÷30m, it should be mentioned that for shorter distances, the analogue TIMC does much more accurate measurements and in general, looks quite competitive. An analysis of circuit replies (Fig. 18a) has showed that the most stable output voltages are produced between 2 and 4 ms. This fact sets the sampling rate for the ADC mentioned in Section III-C (Fig. 5) to 2 KHz.

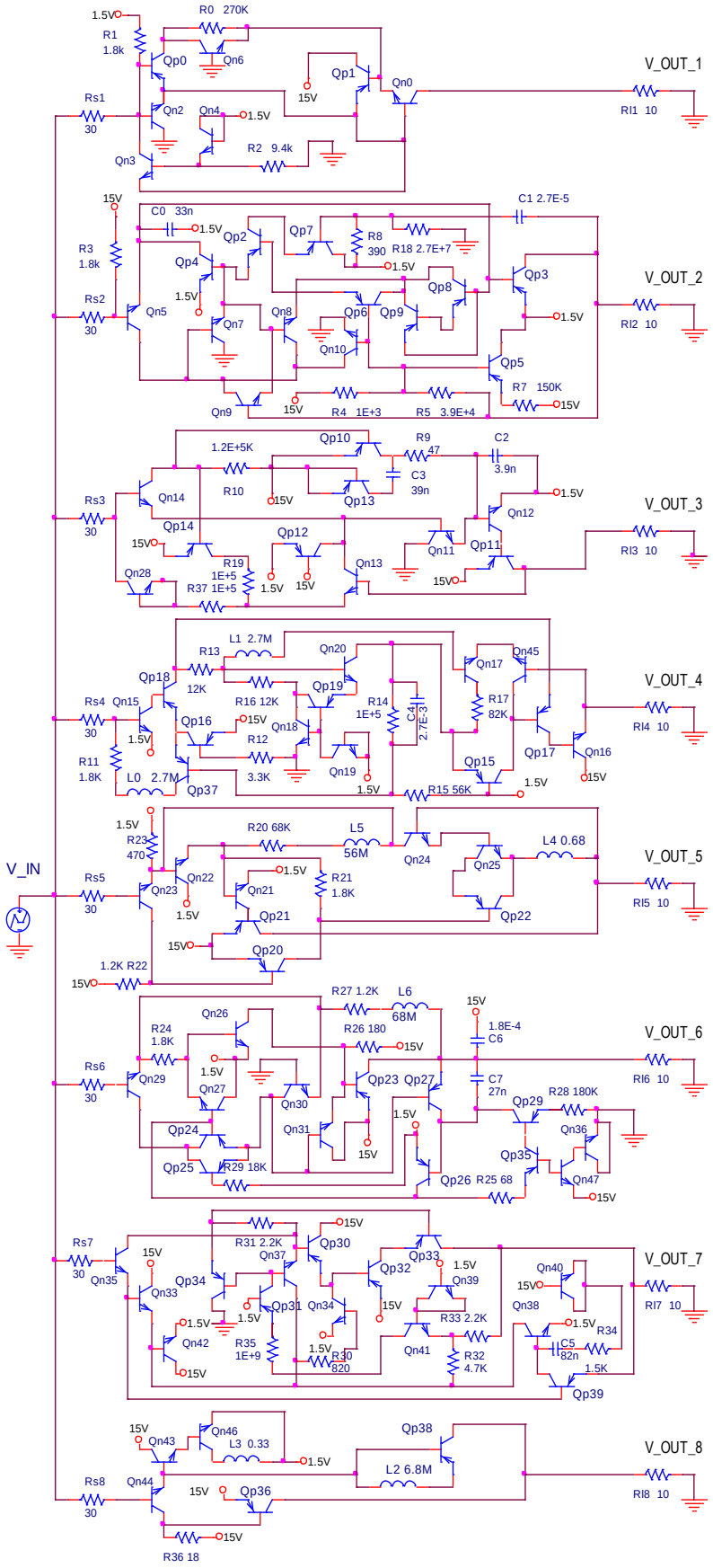

Fig. 15. The evolved 138-component 8-output Voltage Distributor.

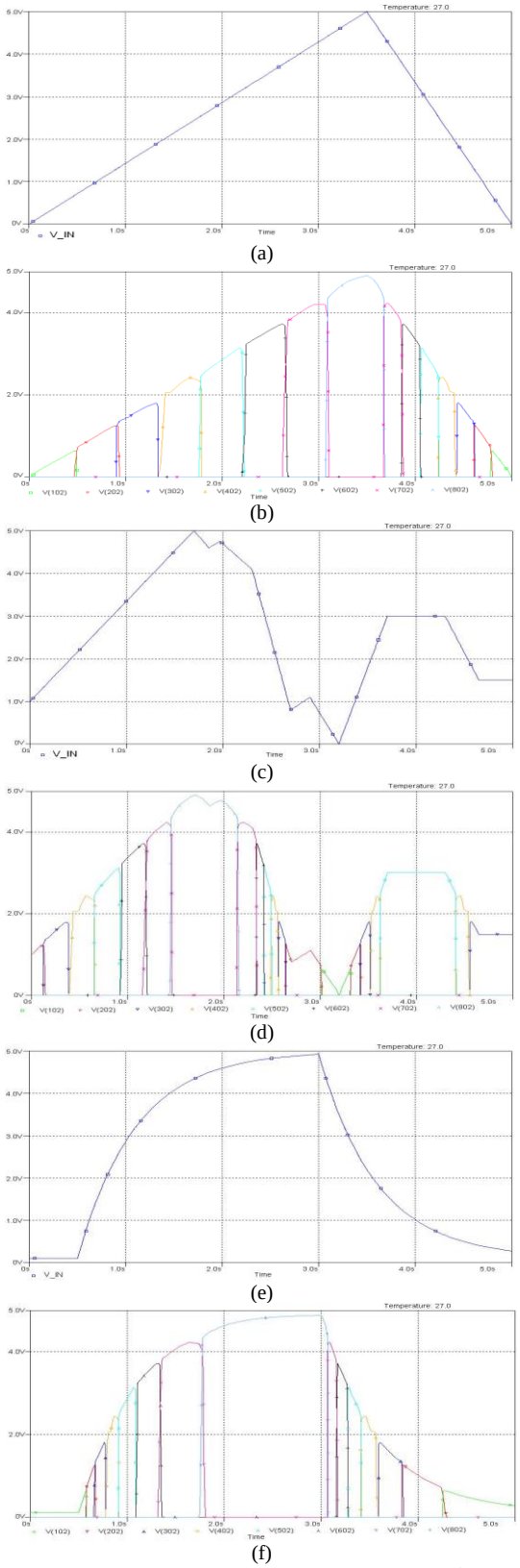

Fig. 16. The transient analysis at input and outputs of the 8-out Voltage Distributor. (a) A piecewise signal used during evolution. (b) The response of the best circuit to a piecewise signal. (c) The incoming arbitrary piecewise signal. (d) The response to an arbitrary piecewise signal. (e) The incoming arbitrary exponential signal. (f) The response to an arbitrary exponential signal.

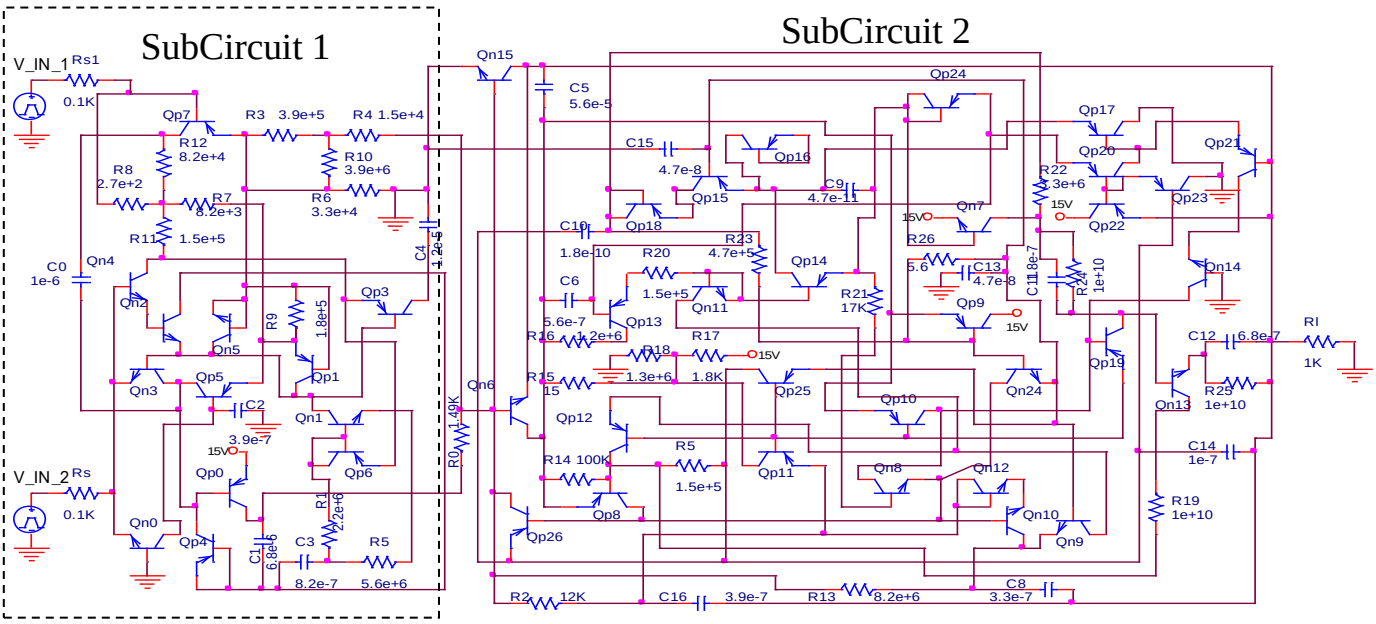

Fig.17. The evolved TIMC consisted of 2 subcircuits: the first subcircuit (dashed) passes the gate pulse to the second one.

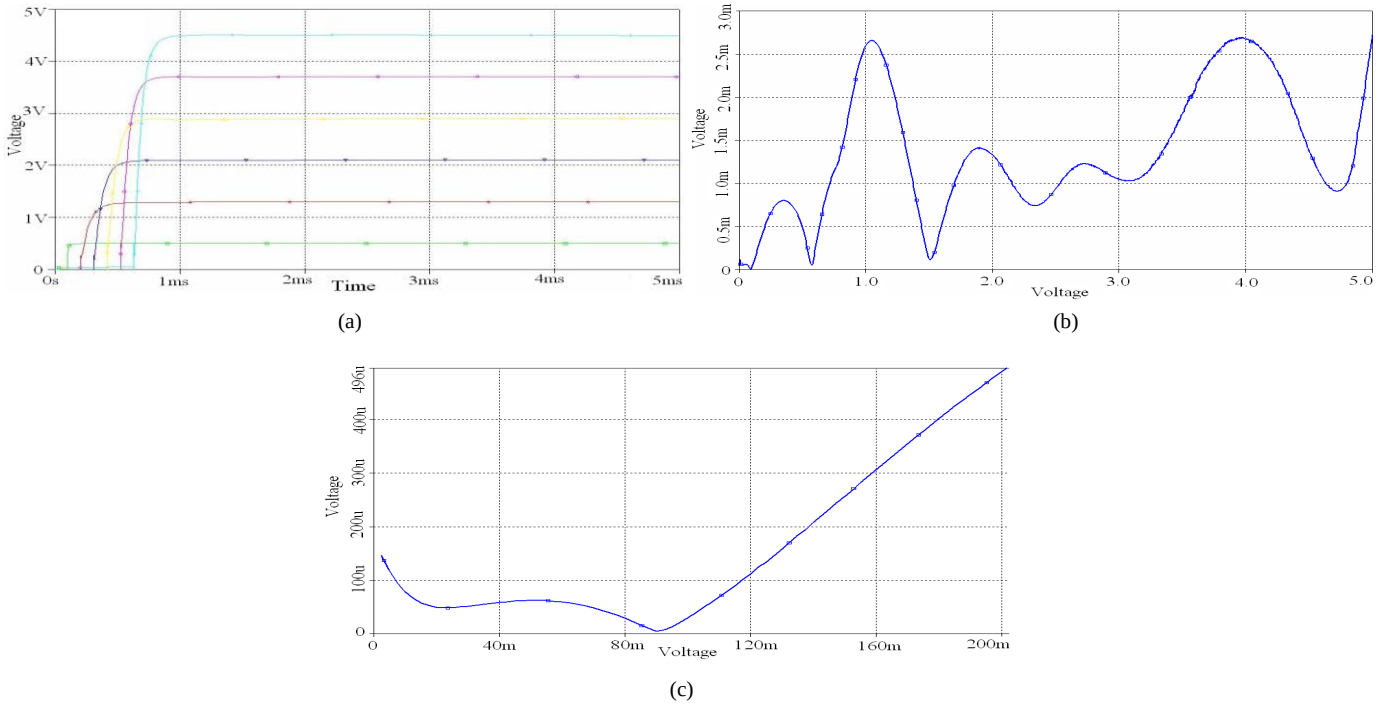

Fig. 18. (a) The voltage replies of the evolved TIMC to six arbitrary incoming signals corresponding to 10, 26, 42, 58 74 and 90km. (b) The function of the absolute average deviation from the ideal circuit response along 1000 equidistant circuit replies. (c) The same as in (b) but here is the fragment from 0 to 0.2V.

Moreover, as was mentioned in Section IV, while solving the generalization problem, we noticed the tendency in which the accuracy of measurements depends directly on the number of input cases during evolution. Thus, it is logical to conclude that reaching the same accuracy for longer distances (30m), and even exceeding it, is just a matter of computing time; however, the last assumption of extrapolation needs to be proven experimentally.

VI. CONCLUSION

In this paper, we described the application of ES to the synthesis of unconventional nonlinear, multi-output and time-dependent functioning analogue circuits. All evolved examples are complex analogue circuits that are able to replace digital circuits in their conventionally adopted applications. To succeed with the first target, we utilized the linear representation, oscillating length genotype strategy and six types of mutations, including the substructure reuse. To

succeed with the last two targets, we additionally applied the incremental evolution and dynamic fitness function that led us to the circuits with the largest component number in the area of analogue Evolutionary Electronics: 138 for 8-out VDC and 87 for TIMC.

The strength of the technique was also proven by direct comparison of the resulted CCs with CCs evolved previously.

One of the main targets of this paper is to show the new potential application area for the analogue circuits synthesized by evolutionary methods. As shown, the circuits designed may compete with digital ones for a number of features, such as economy in circuit components, lower voltage supply and faster signal processing. As mentioned in Section II, the methodology presented does not design robust and industrial-strength circuits what could be considered as the main drawback of the approach. The circuits evolved require further refinement. Therefore, the next work should be focused in that direction.

The human ability to design analogue circuits has some limits. This is supported by Aaserud's "the analogue dilemma" [55]: "Analogue circuit design… usually stretches over a significant period of time and is performed by designers with a large portfolio of skills. It is therefore considered by many to be a form of art rather than a science." In this sense, the targets for the evolution in this paper are selected from a prospective application domain that is problematic for conventional design.

Indeed, the best by size (12 components) conventionally designed cubing circuit [5] (Table 4) has an average error of 7.13mV which is 25 times larger than that (0.29 mV) of the cubing circuit (44 components) evolved in this work. Moreover, during evolution the intermediate result with a fitness of 7.27 was obtained at generation No.20, but with a component number of 11. The next generation of the cubing circuit with 13 components gave a fitness of 6.64.

The human designer with substantial practical experience in the design of analogue and digital circuits has been attempting to design 4/8-output VDCs and TIMC. Considering VDC, the designer draws the conclusion that it is possible to design this circuit purely with analogue components (shown on Fig. 19), but it may take an unduly significant amount of time and effort. The voltage controlled oscillator (VCO) modulates the incoming voltage signal through frequency. The modulated signal comes in the bandpass filters (BPF), each of which is tuned to its own pass band. Particular signals which passed through the BPFs then are demodulated by analogue demodulators. The drawback of such a purely analogue circuit is that each path starting from the BPF input up until the circuit output is independent of the other. This makes the signals at all *N* outputs asynchronous. This last fact may bring problems if someone further utilizes signals from outputs, for example in trying to recreate the original signal. Therefore, synchronization is required at the circuit outputs, which could be set as digital or as analogue. While the first one requires the introduction of additional digital devices and comes at the cost of increased complexity in timing analysis, the last one requires cumbersome transformers. In any case, even without synchronization, the both VDC circuits require a much higher number of components than obtained by evolution.

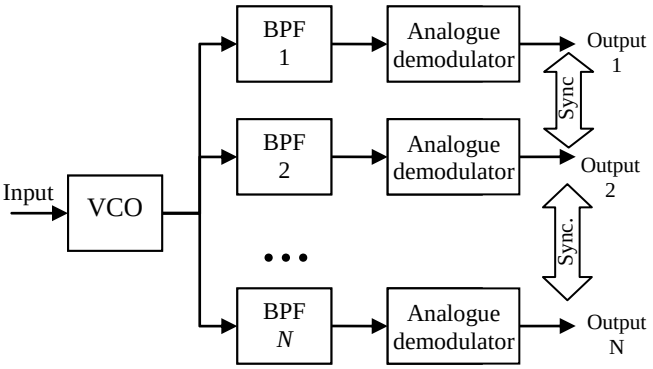

Fig. 19. The human designed N-output VDC with synchronization.

Considering TIMC, the designer concludes that the problem is not in the design of TIMC itself, but rather in reaching such performance features as "usable distance/time range" and "measurement accuracy." Meanwhile, is using purely analogue components there is no visible methodology for making it, while the digital approach to this task has been well-known for many years [38].

The main perspective of the approach suggested from the point of view of future applications is the system's *ability to design the analogue system-on-a-chip* (SOC), where all parts of the proposed electronic system are integrated into a single integrated circuit (IC) chip. Conventionally, SOC "may contain digital, analogue, mixed-signal, and often radio-frequency functions – all on a single chip substrate" [56]. The difference with the current concept of SOC is - while conventional SOCs are supposed to contain digital and mixed analogue-digital signals - the proposed SOC is suggested to contain only analogue components that process purely analogue signals. The advantages that this technology may bring to potential users in comparison with conventional SOC are:

1. The compactness of such the systems being located in one crystal. This advantage is based on economizing on synchronization circuits, and other redundant circuits that are required to support digital logic inside ICs. Moreover, as is presented in this paper, EHW suggests considerable economy in terms of components in comparison with human-designed analogue circuits.
2. Decreased power consumption. This feature is caused by prior advantage as well as by the inherent nature of analogue electronics. Furthermore, the evolutionary approach is able to enable a designer to set the preferable power supply as one of an evolution's objective. This feature is especially pertinent given trends in global energy efficiency.

In this sense, CCs, 4/8-output VDCs and TIMC - may be regarded as analogue SOCs. The functions that these systems perform are quite simple from the point of view of the digital designer, but they are hard issues for the specialist in analogue. The last two targets do not exist in analogue circuitry, but in digital circuitry they comprise the bulk of digital circuits. There are a lot of applications that may benefit from the proposed technique. Some of them are those that utilize sensors and require tiny sizes and low power consumption, such as, *wearable electronics* [57] and

embedded systems [58].

REFERENCES

[1] R. Zebulum, M. S. Vellasco, M. A. Pacheco, "Variable Length Representation in Evolutionary Electronics", *Evolutionary Computation*, v.8 n.1, March 2000, pp.93-120.

[2] R. Ingo, Optimierung Technischer Systeme Nach Prinzipien der Biologischen Evolution, *PhD dissertation*, Technical University of Berlin, 1970.

[3] W. Mydlowec, J. Koza, Use of Time-domain Simulations in Automatic Synthesis of Computational Circuits Using Genetic Programming, in *Genetic and Evolutionary Computation Conference*, Late Breaking Papers, Las Vegas, Nevada, 2000, pp. 187 - 197.

[4] M. Streeter, M. Keane, J. Koza, Iterative Refinement of Computational Circuits Using Genetic Programming, *in Proc. of Genetic and Evolutionary Computation Conference*, San Francisco, CA: Morgan Kaufmann, 2002, pp 877-884.

[5] S. Cipriani, A. Takeshian, Compact Cubic Function Generator, U. S. patent 6,160,427. Filed September 4, 1998. Issued December 12, 2000.

[6] J. Koza, Automated synthesis of computational circuits using genetic programming, in 1997 *IEEE Conference on Evolutionary Computation*, 1997, pp. 447–452, Piscataway, NJ: IEEE Press

[7] F. Bloom, "Introduction to preclinical neuropsychopharmocology", *Psychopharmocology: The Forth Generation of Progress*, Edited by Bloom F.E., Kupfer D.J. New York, Raven, 1995, pp. 1-7

[8] D. Hubel, "Eye, Brain, and Vision", *Scientific American Library*, New York, 1988. Available: http://hubel.med.harvard.edu/book/bcontex.htm.

[9] T. McConaghy and G.G.E. Gielen, "Globally reliable variation-aware sizing of analogue integrated circuits via response surfaces and structural homotopy", *IEEE Transactions on Computer-Aided Design*, 28(11), Nov. 2009, pp.1627-1640.

[10] K. Kim, A. Wong, H. Lipson, "Automated Synthesis of Resilient and Tamper-Evident Analogue Circuits without a Single Point of Failure", *Genetic Programming and Evolvable Machines (online)*, 2009.

[11] A. Thompson, P. Layzell, Evolution of Robustness in an Electronics Design, in International Conference on Intelligent Engineering Systems, Springer, Berlin/Heidelberg, Germany, 2000, pp. 218–228.

[12] A. Stoica, D. Keymeulen, T. Arslan, V. Duong, R. Zebulum, I. Ferguson and X. Guo, Circuit Self-Recovery Experiments in Extreme Environments, *Proceedings of the 2004 NASA/DoD Conference on Evolvable Hardware*, pp. 142-145, Seattle,USA, June, 2004.

[13] J. Walker, J. Hilder and A. Tyrrell, Towards Evolving Industry-feasible Intrinsic Variability Tolerant CMOS Designs, *11th IEEE Congress on Evolutionary Computation*, 1591-1598, Trondheim, Norway, May, 2009.

[14] J. Koza, L. Jones, M. Keane, M. Streeter, S. Al-Sakran, "Toward automated design of industrial-strength analogue circuits by means of genetic programming", in *Genetic Programming Theory and Practice II*, Chapter 8, Boston: Kluwer Academic Publishers, 2004, pp. 121–142.

[15] J. D. Lohn, G. L. Haith, S. P. Colombano, D. Stassinopoulos, "A Comparison of Dynamic Fitness Schedules for Evolutionary Design of Amplifiers", *in Proc. of the NASA/DoD Workshop on Evolvable Hardware*, IEEE Computer Society, CA, U.S.A., 19-21 July, 1999, pp. 87-92.

[16] C. Goh and Y. Li, "GA automated design and synthesis of analogue circuits with practical constraints", *The Congress on Evolutionary Computation*, Vol. 1, 2001, pp.170-177.

[17] S. Ando, H. Iba, "Analogue Circuit Design with a Variable Length Chromosome", *Congress on Evolutionary Computation*, IEEE Press, 2000, pp.994-100.

[18] J.B. Grimbleby, "Hybrid genetic algorithms for analogue network synthesis", *Congress on Evolutionary Computing*, Washington, USA, 1999, pp. 1781-1787.

[19] J. Hu, X. Zhong, E. Goodman, "Open-ended Robust Design of Analogue Filters Using Genetic Programming", *Genetic & Evolutionary Computation Conference*, ACM Press, Vol.2, Washington, DC, 2005, pp.1619-1626.

[20] Y. Sapargaliyev and T. G. Kalganova, "On Comparison of Constrained and Unconstrained Evolutions in Analogue Electronics on the Example of "LC" Low-Pass Filters", *IEICE Transactions on Electronics*, Vol.E89-C No.12, 2006, pp.1920-1927.

[21] Z. Gan, Z. Yang, G. Li, M. Jiang, "Automatic Synthesis of Practical Passive Filters Using Clonal Selection Principle-Based Gene Expression Programming", *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, ISSN 0302-9743, Volume 4684, 2007, pp. 1611-3349 (Online).

[22] S. Chang, H. Hou, Y. Su, "Automated Passive Filter Synthesis Using a Novel Tree Representation and Genetic Programming", *IEEE Trans. on Evolutionary Computation*, vol. 10, 2006, pp. 93-100.

[23] A. Thompson, *Artificial evolution in the physical world*, in Gomi (Ed.) *Evolutionary Robotics*, AAI Books, 1997.

[24] C. Mattiussi, D. Floreano, "Analogue Genetic Encoding for the Evolution of Circuits and Networks," *IEEE Trans. on Evolutionary Computation*, vol. 11, 2007, pp. 596-607.

[25] F. Wang, Y. Li, K. Li, Z. Lin, "A New Circuit Representation Method for Analogue Circuit Design Automation", *IEEE World Congress on Computational Intelligence*, IEEE Press, Hong Kong, 2008.

[26] T. Sripramong and C. Toumazou, "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, 2002, pp.1237–1252.

[27] H. Shibata,; N. Fujii, "Automated design of analogue computational circuits using cell-based structure", *Circuits and Systems, ISCAS 2002. IEEE International Symposium*, Volume 2, 26-29 May 2002, pp.II-53 - II-56.

[28] X. Xia, Y. Li, W. Ying, L. Chen, "Automated design approach for analogue circuit using genetic algorithm", *Lect. Notes Comput. Sci.*, 4490, 2007, 1124–1130.

[29] T. Dastidar, P. Chakrabarti, P. Ray, "A synthesis system for analogue circuits based on evolutionary search and topological reuse", *IEEE Trans. on Evolutionary Computation*, Vol. 9, Issue 2, 2005, pp. 211 – 224.

[30] W. Kruiskamp and D. Leenaerts, "Darwin: CMOS opamp synthesis by means of a genetic algorithm", *Proc. 36th Design Autom. Conf.*, 1995.

~~[31] G. Aplaydin, S. Balkir, G. Dündar, "An Evolutionary Approach to Automatic Synthesis of High-Performance Analogue Integrated Circuits", *IEEE Transactions on Evolutionary Computation*, 2003, pp. 240--252.~~

[32] A. Das, R. Vemuri, "An Automated Passive Analogue Circuit Synthesis Framework using Genetic Algorithms", *IEEE Computer Society Annual Symposium on VLSI*, 2007, pp. 145-152.

[33] T. Kalganova, "Bidirectional incremental evolution in evolvable hardware," *Proc. 2nd NASA/DoD Workshop Evolvable Hardware*, Jul. 13–15, 2000, pp. 65–74.

[34] L. Sekanina, M. Bidlo, "Evolutionary Design of Arbitrarily Large Sorting Networks Using Development", *Genetic Programming and Evolvable Machines*, Vol. 6, No. 3, September 2005, pp. 319-347.

[35] G. He; N. Xiong, A. V. Vasilakos, Y. Li, Z. Shi, "Automated Design of Logic Circuits with a Increasable Evolution Approach", *Proc. of 11th IEEE International Conference on High Performance Computing and Communications*, IEEE Computer Society, Volume 00, Washington, DC, 2009, pp. 206-213

[36] J. R. Koza, *Genetic Programming II: Automatic Discovery of Reusable Programs*, Cambridge, MA: MIT Press., 1994.

[37] B. Vigoda, "Analogue Logic: Continuous-Time Analogue Circuits for Statistical Signal Processing" [Online] Sep. 2003, Massachusetts Institute of Technology, XP007904886; URL: http;//media.mit.edu/physics/publications/theses/03.07.vigoda.pdf.

[38] Feodosia state Optical Factory, The Artillery Quantum rangefinder (online), http://fkoz.feodosia.com.ua/main3.phtml?link=23

[39] Y. Sapargaliyev and T.G. Kalganova, "Unconstrained evolution of analogue computational 'QR' circuit with oscillating length representation," in G.S. Hornby et al., eds, *Proc. Intern. Conf. Evolv. Syst.*, 2008.

[40] T. Kuyucu, M. A. Trefzer, J. F. Miller and A. M. Tyrrell, "Task Decomposition and Evolvability in Intrinsic Evolvable Hardware", Proceedings of the IEEE Congress on Evolutionary Computation CEC 2009, Trondheim, Norway, May 2009.

[41] M. Trefzer, J. Langeheine, J. Schemmel, and K. Meier, "Operational Amplifiers: An Example for Multi-Objective Optimization on an Analogue Evolvable Hardware Platform", In J. M. Moreno, J. Madrenas, and J. Cosp, editors, *Evolvable Systems: From Biology to Hardware, Sixth International Conference, ICES 2005*, number 3637 in LNCS, pages 86-97, Sitges, Spain, September 2005. Springer-Verlag.

[42] "Model 2145 Time to Amplitude Converter/Single channel Analyzer", *Datasheet*, Canberra Industries, Inc., Available: www.canberra.com/pdf/Products/Model-2145-SS-0226.pdf.

[43] *PSpice A/D reference guide*. Version 10.0. San Jose: Cadence Design Systems, Inc., 2003.
[44] F. Bennett III, J. Koza, M. Keane, J. Yu, W. Mydlowec and O. Stiffelman, "Evolution by Means of Genetic Programming of Analogue Circuits That Perform Digital Functions," in *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference*, July 13–17, 1999, pp. 1477-1483.
[45] M. A. Trefzer, "Evolution of Transistor Circuits," *Dissertation for the degree of Doctor of Natural Sciences*, Ruperto-Carola-University of Heidelberg, Germany, December 2006.
[46] H. Yuan and J. He, "Evolutionary design of operational amplifier using variable-length differential evolution algorithm, Computer Application and System Modeling (ICCASM)," in *2010 International Conference* on, pp. V4-610 - V4-614, Oct. 2010.
[47] P. Conca, G. Nicosia, G. Stracquadanio and J. Timmis, "Nominal-Yield-Area Tradeoff in Automatic Synthesis of Analogue Circuits: A Genetic Programming Approach Using Immune-Inspired Operators," NASA/ESA Conference, Adaptive Hardware and Systems, AHS 2009, in Francisco, CA, July 29 2009-Aug. 1 2009, pp. 399-406.
[48] S. L. Sabat, K. S. Kumar and S. K. Udgata, "Differential Evolution and Swarm Intelligence Techniques for Analogue Circuit Synthesis," in *World Congress on Nature & Biologically Inspired Computing*, Dec. 9-11 2009, pp. 469-474.
[49] P. Layzell, "Hardware Evolution: On the Nature of Artificially Evolved Electronic Circuits," *PhD thesis*, School of Cognitive and Computing Science, University of Sussex, Sussex, 2001.
[50] J. He; M. Liu and Y. Chen, "A Novel Real-Coded Scheme for Evolutionary Analogue Circuit Synthesis, Ligent Systems and Applications," in *International Workshop*, May 23-24 2009, pp. 1-4.
[51] K. Ohe, M. Konishi and J. Imai, "Design support classifier of filter circuit structure," in *SICE Annual Conference*, Aug. 20-22 2008, pp. 2695-2699.
[52] J. Torresen, "A divide-and-conquer approach to evolvable hardware," in *Evolvable Systems: From Biology to Hardware. Second International Conference*, ICES 98, LNCS 1478, Springer-Verlag, 1998, pp. 57-65.
[53] J. Walker, K. Völk, S. Smith and J. F. Miller, "Parallel evolution using multi-chromosome Cartesian genetic programming," in *Genetic Programming and Evolvable Machines*, vol. 10, no. 4, 2009.
[54] I. Harvey, P. Husbands and D. Cliff, "Seeing the light: artificial evolution; real vision," in D. Cliff, P. Husbands, J. A. Meyer and S. Wilson, (eds.), *From Animals to Animals 3, Proceedings of the third international conference on Simulation of Adaptive Behavior*, MIT Press/Bradford Books, 1994.
[55] O. Aaserud, I. Nielsen and I. Ring, "Trends in current analogue design: A panel debate," in *Analogue Integrated Circuits and Signal Processing*, vol. 7, no. 1, 1995, pp. 5-9.
[56] B. Al-Hashimi (ed.), *System On Chip: Next Generation Electronics*, IEE Circuits Devices and Systems Book Series, 2006.
[57] A. Lymberis and D. De Rossi, *Wearable ehealth systems for personalised health management — state of the art and future challenges*, IOS Press, 2004.
[58] L. Sekanina and V. Drábek, "Theory and applications of evolvable embedded systems," in *Proc. of the 11th IEEE Int. Conference and Workshop on the Engineering of Computer-Based Systems*, IEEE CS Press, 2004, pp. 186-193.

Yerbol Sapargaliyev received MSc. degree in physics from Kazakh State University, Kazakhstan, in 1993 and MA in Economics from Kazakhstan Institute of Management Economics and Strategic Research, in 1995. From 1994 to 1999, he worked as a Marketing Manager in Arna-Sprint Data communications Ltd. From 1999 to 2005 he worked as a Marketing Manager in Huawei Technologies, Co., Ltd. Since 2005 he joined Bio-inspired intelligent system research group, School of Engineering and Design, Brunel University, UK, where he is currently a PhD Research Student.

Tatiana Kalganova received MSc degree from Belarusian State University of Informatics and Radioelectronics, Belarus in 1994 and PhD degree from Napier University, UK in 2000. In August 2000 she has joined Electronic and Computer Engineering Department, Brunel University. Her research interests are evolvable hardware, ant colony algorithms, scalability in AI systems. She was awarded a personal grant from the Education Ministry of the Republic of Belarus for distinctive achievements in the field of exact sciences in 1997, and a grant from the International Soros Science Education Program (ISSEP) for distinctive achievements in the field of exact sciences in 1996.