An FPAA Approach to Adaptive Filter Design with Evolutionary Software-Driven Reconfiguration

P. Farago[1], G. Csipkes[1], D. Csipkes[1], C. Farago[1], S. Hintea[1]
[1]*Department of Bases of Electronics, Technical University of Cluj-Napoca, G. Baritiu St. 26-28, 400027, Cluj-Napoca, Romania*
paul.farago@bel.utcluj.ro

***Abstract*—The ever increasing requirement of highly complex processing functions integrated in silicon attracts new paradigms for reconfigurable circuit design. The target is the development of fully adaptive mixed-signal systems on a chip, with a conclusive example found in the field of radio communications. This paper proposes an auto-adaptive filter developed for analog filtering in next-generation intelligent multi-mode wireless receivers. The adaptive system is built around a field programmable analog array. A simultaneous reconfiguration algorithm performs both the design and synthesis of the analog filter providing partial solutions along the reconfiguration process. Topology synthesis is implemented with a look up table and parametric design is performed with an evolutionary design scheme. A hierarchical self-repair mechanism then operates towards invariance vs. environmental changes and post-silicon failure. The proposed auto-adaptive filter concept is validated with extensive transistor-level simulation and specific design illustrations of GSM and WCDMA intermediate frequency analog filters.**

***Index Terms*—Adaptive filters, evolutionary design, field programmable analog array, phase locked loop, self-repair, software-driven reconfiguration.**

I. INTRODUCTION

Latest trends in mixed-signal integrated circuits (IC) target the extensive integration of highly sophisticated processing functions on the same silicon dye, in the shape of mixed-signal systems on a chip (SoC). Considering the ever increasing demand for processing capabilities in both digital and analog form, it becomes ever more demanding for the circuit designer to keep up with the design challenges. New design paradigms need to be formulated, which mainly materialize in the shape of reconfigurable hardware, software tools for design automation and integrated system adaptivity.

A solution for system adaptivity is found in the joint deployment of reconfigurable circuitry with software-controlled auto-reconfiguration mechanisms. The reconfigurable circuit provides the key features of topological reconfiguration and parameter programming for the implementation of a variety of processing functions. Tools for design automation then implement a software control of the reconfiguration process. We have validated the premises of software controlled reconfiguration in [1], where we have interfaced a field programmable analog array (FPAA) with an evolutionary reconfiguration algorithm.

Recently, the requirements for intelligent systems which adapt to their operating environment have shown an ever increasing tendency influencing every aspect of modern life [2], [3]. A conclusive example can be found in the field of multi-standard radio communication systems. Facing the challenges of connectivity “anytime, anywhere”, an intelligent multi-standard radio receiver needs to cope with the coexistence of a variety of radio access interfaces, ranging from cellular communications and local and personal area networks to satellite connectivity. Under the current scenario of coexisting wireless access interfaces, with the continuous development of novel long term evolution (LTE) networks, next generation communication devices will target connectivity to up to 14 frequency bands, including 2G roaming, 3G voice and data transfer, 4G data, dual-band Wi-Fi, etc., while maintaining unconditional receive diversity for simultaneous communication [4].

Designed to cope with the coexistence of a variety of wireless interfaces, the multi-standard radio receiver adapts to the particular requirements of the targeted standard specifications. Adaptivity must be implemented all along the processing chain, ranging from the analog front-end, i.e. radio frequency (RF) and intermediate frequency (IF) processing modules, up to the baseband digital processor.

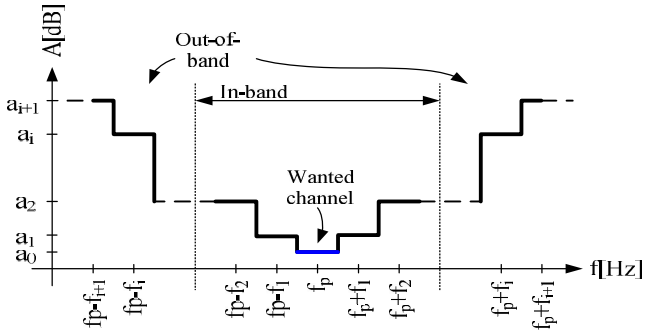

Fig. 1. A generic blocker profile defined by the wireless communication standard.

In this paper we have focused towards adaptive IF filtering in multi-standard radio receivers. The wireless communication standard IF-filtering requirements are

Manuscript received October 29, 2013; accepted March 5, 2014.

expressed in terms of a blocker profile depicted in Fig. 1. The blocker profile, as an illustration of the filtering specifications, characterizes the wireless communication link and imposes the standard-specific performance requirements of the receiver module to correctly receive the wanted signal even in the presence of in-band and out-of-band interferers.

In this paper we propose an auto-adaptive analog filter as solution to the problem of adaptive IF filtering in intelligent multi-standard radio receiver front-ends. The block diagram of the proposed auto-adaptive filter is illustrated in Fig. 2.

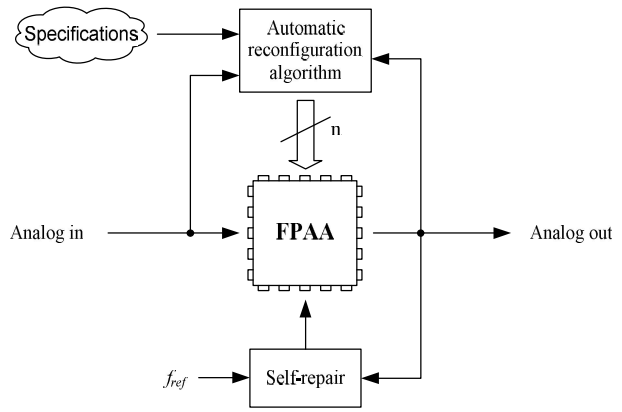

Fig. 2. Block diagram of the proposed auto-adaptive analog system.

The auto-adaptive analog filter is built around an FPAA, which is interfaced to an automatic reconfiguration algorithm to implement software-driven reconfiguration of the analog filter via topological synthesis and parametric design. Additionally, proper operation of the analog system is ensured regardless of environmental changes, process, voltage and temperature (PVT) variations, fabrication tolerance, post-silicon failure, etc., with a built-in self-repair mechanism.

We have reused the FPAA, and the automatic topological and parametric design schemes we have formerly reported in [1]. The novelty of this paper is an improved implementation of the automatic reconfiguration algorithm which provides increased generality for filter synthesis, exhibiting a strong parallelism in the reconfiguration process. Also, a complex hierarchical self-repair mechanism of the integrated analog filter is provided.

This paper is organized as follows. Section II presents the proposed FPAA developed for auto-adaptive IF analog filters, describing the FPAA topology, the FPAA building blocks and the on-chip wideband self-calibration circuit built with a phase locked loop (PLL). Section III presents the automatic design algorithm to achieve software-driven FPAA reconfiguration. Section IV deals with the proposed hierarchical self-repair mechanism. Finally, Section V validates the proposed adaptive analog filter via extensive simulation, with the illustration of the fully automatic synthesis of GSM and WCDMA low-IF filters.

II. THE PROPOSED FPAA FILTER ARCHITECTURE

In this work, we are targeting the specific application of adaptive IF filtering in intelligent multi-mode multi-standard radio receivers. The FPAA, as a reconfigurable platform for adaptive system development, must exhibit two key aspects: hardware reconfigurability and parameter programmability. The block diagram of the proposed FPAA is illustrated in Fig. 3 and resembles a cascade of configurable analog blocks (CAB).

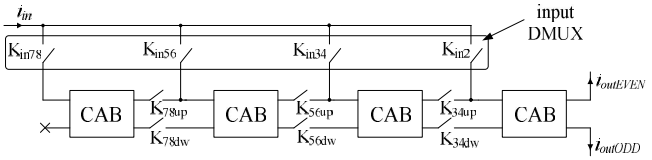

Fig. 3. Block diagram of the proposed FPAA [5].

Our FPAA design approach was to sacrifice generality and favor performance in order to comply with the operation requirements imposed by the radio communication standards. We have basically replaced the classical crossbar interconnection network with signal bus architecture [5], implementing two degrees of freedom: programmable filter order and programmable approximation function. Filter order programmability is achieved with an input de-multiplexer, implemented with analog switches K_{in**}, which routes the input signal to the desired network node, and consequently defines the signal path. Programmability of the filter approximation function is then internal to the CABs.

The proposed FPAA assembles the individual building blocks we have formerly reported in [1], [5] and [6]. The schematic of a CAB is illustrated in Fig. 4.

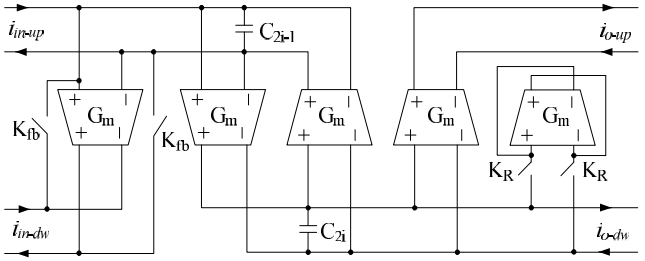

Fig. 4. Schematic of the CAB internal structure [6].

The proposed CAB may be configured to realize first and second order transfer functions in two classical analog filter topologies: cascade of Tow-Thomas (TT) biquads and state-variable leap-frog (LF) filters. Gm-C lossless integrators are built around integrating capacitors C_{2i-1} and C_{2i}. Analog switch K_{fb} creates a local negative feedback to implement an active resistor needed either for a lossy integrator or a LF input, as well as a negative feedback gain for a TT biquad. Analog switch K_R implements the termination resistance of a LF output [5]. The CAB configuration corresponding to different filter types and orders is listed in Table. I.

TABLE I.CAB SWITCH CONFIGURATION FOR VARIOUS FILTER TOPOLOGIES.

	K_{fb}	K_R	Input port	Output port
Lossy integrator	ON	off	i_{in_up}	i_{o_dw}
TT biquad	ON	off	i_{in_up}	i_{o_up}
LF input	ON	off	i_{in_up}	$i_{o\_up} + i_{o\_dw}$
Leap frog (LF)	off	off	$i_{in\_up} + i_{in\_dw}$	$i_{o\_up} + i_{o\_dw}$
LF output even	off	off	$i_{in\_up} + i_{in\_dw}$	i_{o_up}
LF output odd	off	ON	$i_{in\_up} + i_{in\_dw}$	i_{o_dw}

Programmability of the FPAA parameters is achieved via variation of the Gm-C filter transconductance and the

capacitance values respectively.

The transconductance value is programmed with the binary weighted transconductor array illustrated in Fig. 5, which yields an equivalent transconductance value equal to

$$G_m = G_{mu}\left(1+\sum_{i=0}^{2} b_i \cdot 2^i\right), \quad (1)$$

where G_{mu} is the unit transconductance. The unit OTA cell was implemented with a differential pair loaded with a gain-boosted folded-cascode output stage, Fig. 5, creating a very large output resistance to reduce the phase error at the lossless integrator unity gain frequency [7]. Dynamic degeneration was applied to the input differential stage to implement linearization via feedback [8].

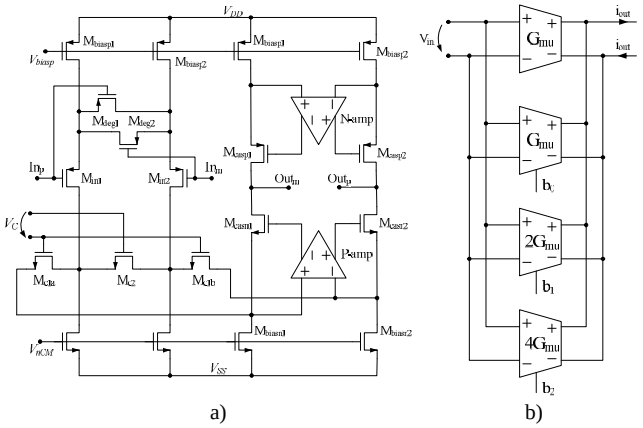

Fig. 5. The binary weighted transconductor array: (a) OTA internal schematic, and (b) schematic of the array [1].

Programmability of the capacitance value was implemented using the capacitance scaling circuit illustrated in Fig. 6, which operates towards a high-resolution control of the analog filter cutoff frequency and the magnitude and phase responses respectively [1], [6].

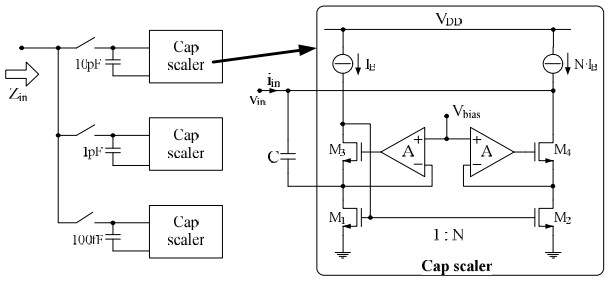

Fig. 6. The virtual capacitance scaling circuit [1], [5].

Capacitance scaling is achieved with the programmable gain-boosted cascode current mirrors [6], [9], implementing a decade-based switching template to emulate capacitance values in the 100 fF – 99.9 pF range according to equation

$$C_{eq} = 10pF \cdot (1+N_1) + 1pF \cdot (1+N_2) + 100fF \cdot (1+N_3), \quad (2)$$

where N_1, N_2 and N_3 are the gains of the three current mirrors respectively.

The binary weighted transconductor array implements a coarse tuning of the analog filter, and is employed for the division of the targeted frequency range into sub-bands. The capacitance scaling circuit on the other hand operates towards a high-resolution control of the analog filter, and is employed for in-band fine-tuning of the analog filter cutoff frequency, magnitude and phase responses respectively.

In order to deal with the inherent frequency errors, most practical filters feature a built in self calibration mechanism that fits the parameters of the filter within a tolerated error range around the nominal value. The main challenge in implementing FPAA self tuning is finding the filter parameters that can be adjusted within a sufficiently wide range of values that permit the coverage of all the supported programmable frequencies, while also ensuring independence on the chosen topology, order and corner frequency.

Conventional self calibration techniques assume that the filter circuitry has been designed for a fixed frequency and is therefore not variable. In these cases it is relatively easy to find the components whose parameters are proportional to the time constants to be calibrated. However, in programmable filters the adjustable components have already been used to implement the programmable transfer function. It results that the filter requires an additional degree of freedom that allows a practical combination between the reconfigurable architecture and the necessary wide band error compensation scheme.

In the proposed reconfigurable filter the additional degree of freedom is provided by the programmable transconductance G_{mu} of the unit OTA cell. A resistive divider implemented with triode transistors M_{c1a}, M_{c1b} and M_{c2}, as illustrated in Fig. 5, provides the required degree of freedom for fine-tuning the transconductance value via a control voltage V_C, according to equation

$$G_{mu}^{*} = G_m\left(\frac{1}{2} \pm \frac{V_{CD}}{4(V_{CCM} - V_{od-C})}\right), \quad (3)$$

where V_{CD} and V_{CCM} are the differential and the common mode components of the control voltage respectively, and $V_{od\text{-}C}$ is the overdrive voltage of the transistors building the resistive divider [10].

The principles of the wide band self calibration approach, similar with classical methods, are illustrated in Fig. 7 [8], [11]. In this configuration the slave filter features the reconfigurable architecture whose parameters need to be calibrated. The master filter, implemented as a PLL loop, is only used to measure the relative frequency error and compute the required adjustment. The VCO within the PLL has been constructed around the same fundamental OTA cell and the same capacitor types used in the reconfigurable filter itself. It results that the VCO output frequency f_{VCO} will be proportional to the real corner frequency of the filter and the relative errors, caused by process corners and environmental effects, will be identical. Consequently, when the PLL achieves a locked state, it synchronizes the VCO output frequency f_{VCO} with the reference f_{ref} and determines the control voltage V_C of the OTA cells required to eliminate the frequency errors. The same V_C voltage can be then used to adjust the transconductances in the slave filter. It is of particular importance that the self tuning mechanism uses a

relative error rather than absolute values to compute the necessary compensation. Therefore, the self calibration works even when the VCO output frequency is different from the filter cut off frequency. It results that the wide band error compensation is not influenced by the filter parameters designed by evolutionary computation. Furthermore, since the approximation of the filter response is determined by capacitance ratios, changing the unit transconductance G_{mu} conserves the shape of the filter response, regardless of the error and the chosen frequency parameters.

The schematic of the VCO implemented with linearized OTA cells is shown in Fig. 8 [8].

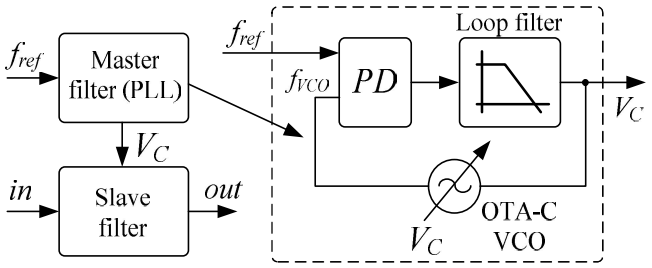

Fig. 7. Principle of classical PLL tuning of cut off frequency in low pass filters [1].

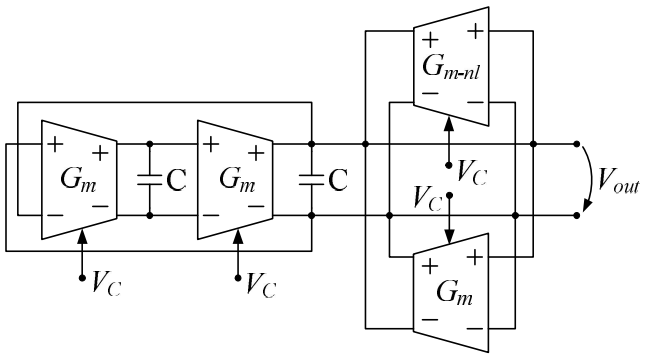

Fig. 8. OTA-C implementation of the VCO [1].

The oscillation frequency may be calculated according to

$$f_{osc} = \frac{G_m}{2®C}. \quad (4)$$

The peak voltage generated by the VCO must fall within the linear input range of the OTA-s. The amplitude of the output sine wave is controlled by the transconductance of the non-linear OTA cell $G_{m\text{-}nl}$, according to the principles described in [8]. The non-linear voltage to current characteristic of the OTA causes the poles of the VCO to move back and forth around the imaginary axes, thereby stabilizing the output signal amplitude. The non-linear OTA has been obtained by eliminating the dynamic degeneration resistors from the cell used in the reconfigurable filter and by adjusting the bias current for a *700 mV* peak to peak differential output amplitude.

In the proposed implementation of the self calibration circuitry the phase detector is a linearized analog multiplier realized as a folded Gilbert cell with an additional cascode output stage [12]. The loop filter is a simple RC passive network with one pole and one zero allowing the PLL to track frequency changes caused by process corners and temperature variations.

III. AUTOMATIC SOFTWARE DRIVEN RECONFIGURATION

In this paper, we propose a novel reconfiguration algorithm which synthesizes the FPAA filter starting from the filter blocking profile supplied as design specifications. The block diagram of the proposed reconfiguration algorithm is illustrated in Fig. 9 and exhibits two stages for FPAA reconfiguration: design of the filter transfer function (TF) and generation of the FPAA reconfiguration bitstring.

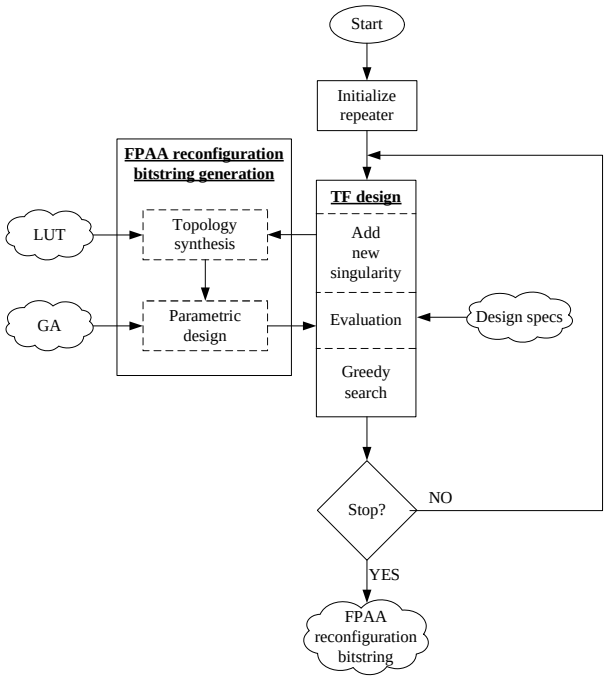

Fig. 9. Block diagram of the proposed FPAA reconfiguration algorithm.

We have built the proposed reconfiguration algorithm around the filter design algorithm based on Greedy search and Genetic Algorithms we have formerly reported in [13], where the transfer function of an analog filter was iteratively designed by adding singularities to a repeater. That filter design algorithm defines four basic operations for the addition of filter singularities, namely simple and complex-conjugated poles and zeros respectively. Placement of the singularity in the complex plane is performed using Genetic Algorithms, and the choice of a specific singularity in each iteration is drawn using Greedy search.

Since the proposed FPAA only provides reconfiguration capabilities for all-pole filters, we have restricted the singularity addition in this work to simple poles and complex conjugated pole pairs. A generic k^{th} iteration of the proposed FPAA reconfiguration algorithm is illustrated in the directed graph from Fig. 10.

To achieve simultaneity with the actual reconfiguration stage, we have extended the Greedy search and GA-based filter design algorithm reported in [13] by adding the topological synthesis and the parametric design stages from our sequential reconfiguration algorithm [1]. The aim of the topological synthesis level is to generate the FPAA reconfiguration bitstring in order to synthesize the analog filter on the reconfigurable array. The aim of the parametric design level is to determine the filter parameter set in order to comply with the imposed design specifications. The novelty of the proposed reconfiguration algorithm is that it

exploits the bi-univocal link between a TF singularity and the circuit section which implements that particular singularity. Thus, in this work we are able to introduce simultaneity in the treatment of the FPAA reconfiguration stages and generate partial solutions even if the reconfiguration process is not yet completed.

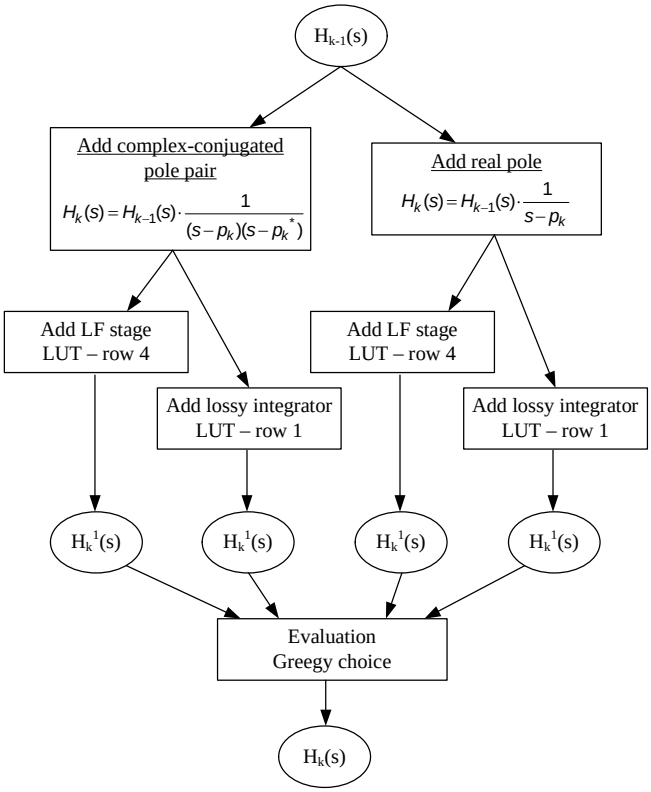

Fig. 10. Illustration of the k^{th} iteration of the proposed reconfiguration algorithm.

According to Fig. 10, a new singularity added to the transfer function implies reconfiguration of the analog filter structure. Addition of a simple pole is implemented with a lossy integrator or a LF section, and addition of a complex conjugated pole pair is implemented with a biquad or two LF sections. Next, the parameters of each of the four analog filter structures are determined, aiming to approximate the blocking profile. Finally, an evaluation engine evaluates the four filters, and the one which best approximates the blocking profile is chosen via Greedy search. After the Greedy iteration, the partial solution can further be improved by increasing the filter order and adding further singularities.

The stopping criterion for the FPAA reconfiguration algorithm is no improvement of the analog filter over two consecutive iterations. The maximum number of iterations is limited by the highest filter order implementable on the FPAA.

The most straightforward strategy for topological synthesis is a LUT, i.e. Table I, which stores the necessary FPAA topological reconfiguration data. LUT query is simultaneous and bi-univocal with the singularity addition operations. The choice between biquad cascade and LF topology is basically drawn considering the filter order. Accordingly, biquad cascade is chosen for small-order filters and LF topology for filters of 6^{th} order or higher. Thus, the topological synthesis stage implements an overall knowledge-based synthesis scheme.

The next stage is the determination of the filter parameters. Classical equation-based topology sizing schemes use symbolic analysis, e.g. Mason's rule, to express a set of circuit design equations. This often leads to a compatible undetermined system of equations. The solving of such systems is difficult and usually leads to conflicting solutions. It is sensible to express that the classical design approach is strongly topology dependent, computational hungry, and often requires an additional optimization stage for fine-tuning the filter parameters. The proposed FPAA however exhibits certain particularities which we exploit in order to reduce the computational burden in the FPAA reconfiguration problem.

The number of filter topologies is limited by the FPAA reconfiguration capabilities, thus it makes sense to have the implementable transfer functions stored in the LUT along with the reconfiguration bitstrings, thus saving considerable computation overhead. Next, for solving the system of design equations, evolutionary computation provides a feasible alternative to the classical equation-based design methods. In the evolutionary approach, the parametric design problem is defined as an optimization problem which optimizes the set of circuit parameters in order to minimize the error between the desired and the implemented circuit performance. The main benefit of the evolutionary approach is that the sizing algorithm is general, regardless of the topology under design, and thus the sizing process is transparent to the actual analog filter

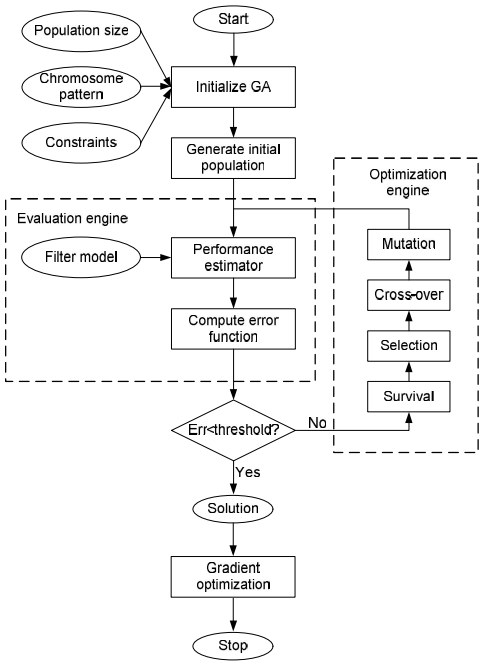

Fig. 11. Block diagram of the parametric level design scheme implemented with GAs [1].

Genetic algorithms (GA) are used in the present work to implement an evolutionary parametric design scheme. GA is an iterative search algorithm which operates on a population of solution candidates rather than on individual solutions, and aims towards the minimization of an objective function. The GA framework models the process of natural evolution. In similarity to the adaptation of natural organisms, the GA

population is iteratively evolved by application of genetic operators inspired from natural evolution, namely selection, cross-over, mutation and survival, in order to produce individuals better fitted to solve the optimization problem.

The block diagram of the evolutionary parametric design scheme we have implemented with GA to solve the filter sizing problem is illustrated in Fig. 11. The first step is to generate a random initial population, in accordance with the GA population size, chromosome pattern and evolutionary constraints. For the representation of an analog filter, the chromosome pattern is expressed in

$$chromosome = [C_k], \tag{5}$$

where $100\,fF \le C_k \le 99\,pF, k = \overline{1,n}$, n is the filter order, C_k is the value of the k^{th} capacitance in the filter topology, and is constrained to the 100 fF – 99 pF variation range, with a 100 fF variation step size. Each individual from the initial population, represented with the chromosome pattern from (5) is an analog filter instance of the filter topology generated in the topological reconfiguration stage.

The next step in the sizing process is the evaluation of the population. In the current work, which handles the analog filter design problem, the aim of the evolutionary algorithm is to minimize the error between the blocking profile, expressed with transfer function $H_{(spec)}(s)$, and the implemented filter characteristics, expressed with transfer function $H(s)$. The error function is expressed as [14]

$$\pounds(s) = H_{(spec)}(s) - H(s), \tag{6}$$

which can be customized to express the errors in the magnitude, phase and group delay characteristics respectively. To be noted is that (6) expresses the objective function of a curve fitting problem, which in the present parameter level design problem operates towards matching the implemented frequency characteristics against the blocking. The main advantage of the error function expressed in (6) is that is uses the filter output to directly compute the objective function.

The optimization goal is to reduce the error function below a certain threshold value, constituting the GA stopping criteria. Should the stopping criteria be fulfilled, the best individual from the current generation is selected and supplied to an additional deterministic gradient optimization stage, and is outputted as the optimization result. Otherwise, genetic operators are applied to the population of solution candidates in order to create a new generation, expectedly better fitted to solve the optimization task.

IV. FPAA Self-Repair Mechanism

Analog filters, implemented in their integrated form, typically encounter large variations of the frequency parameters compared to the mathematical model initially used in the design process. These variations can be as large as ±50 % due to process corners, temperature variations and the tolerances associated with the supply voltage. Following FPAA reconfiguration, the implemented analog filter must perform as specified even in the presence of such variations. We have proposed and implemented a hierarchical self-healing and self-repair mechanism, illustrated in Fig. 12.

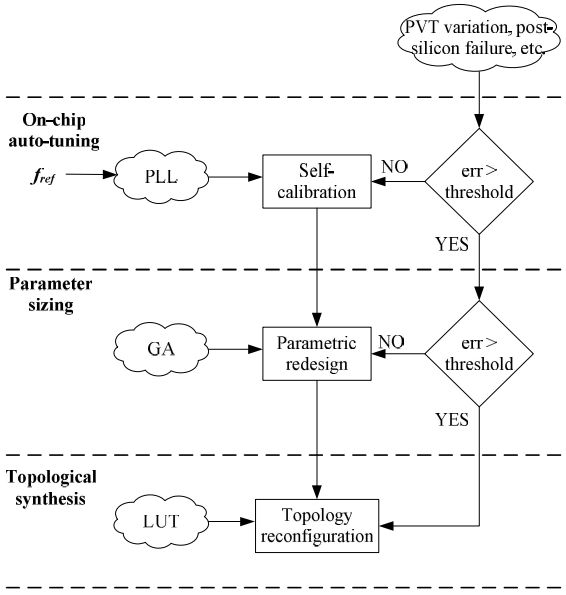

Fig. 12. The proposed FPAA self-repair hierarchy.

The self-repair hierarchy illustrated in Fig. 12 consists of three stages: built-in self calibration, parametric redesign and topological reconfiguration. The self-repair mechanism operates as follows. Slight performance deviations resulting from environmental changes, e.g. PVT variations, are easily compensated within the built-in self calibration module, i.e. the PLL, via intrinsic on-chip auto-tuning to adjust the unit transconductance of the binary weighted transconductor array. The adjustable transconductance does not influence the evolutionary algorithm used to compute the ideal component values. The structure of the unit OTA, particularly the resistive current divider at its output, permits a continuous adjustment of the transconductance by means of a control voltage V_C as described in the previous sections. Since the evolutionary algorithm designs the filter parameters by using a mathematical model, the low level changes of the transconductance G_{mu} will be transparently superimposed with the resulting parameters, allowing a wide band compensation of the frequency errors.

Should the magnitude of the deviations be too large for on-chip auto-tuning, a new set of filter parameters is computed with GA from Fig. 11, one level higher in hierarchy in the parametric redesign stage. For severe performance deviations, e.g. as a result of post-silicon failure, where parametric redesign is insufficient as well, a new topology needs to be synthesized in the topological reconfiguration stage.

V. Simulation Results

The Gm-C FPAA with wide-band self-calibration proposed in Section II was implemented in a 90 nm digital CMOS process and was simulated in the ICStation CAD environment, with Eldo provided by Mentor Graphics. The automatic software-driven reconfiguration algorithm proposed in Section III and the FPAA self-repair mechanism were implemented in Matlab using the Genetic Algorithm

and Direct Search toolbox and the Optimization toolbox. The most significant results are presented as follows.

In order to demonstrate the functionality of the proposed FPAA, we have implemented several 1 dB Chebyshev approximation filters on the reconfigurable array, for various filter orders and cut-off frequencies. The resulting frequency characteristics are plotted in Fig. 13.

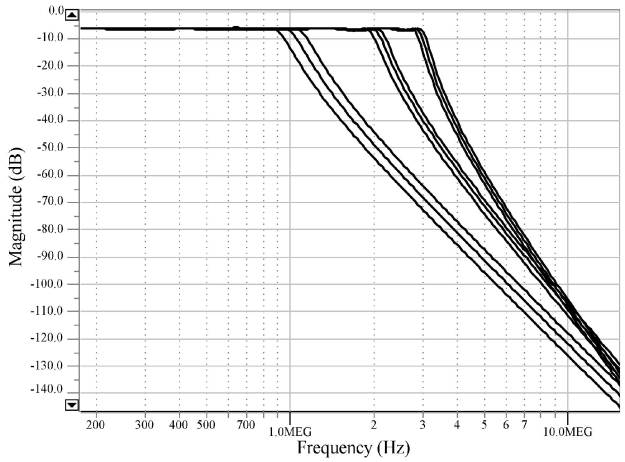

Fig. 13. Simulated magnitude response of the FPAA filter reconfigured for various filter orders and cutoff frequencies.

For the illustration of the software-driven automatic reconfiguration algorithm, we have implemented an evolution strategy which favors survival rather than reproduction. The reason was to have the genetic information of the promising individuals transferred from one generation to the next. To maintain population diversity over generations and avoid a premature biasing of the evolutionary process, we had to force a deviation of the search direction towards new areas in the search space, by implementing a rather high mutation rate. The GA was initialized with a population of 60 individuals. The genetic operators were set as follows: survival of 6 individuals, roulette wheel selection, scattered cross-over with a 60 % rate and Gaussian mutation with a 40 % rate. Our tests show that, for the all-pole classical approximation filters, an evolutionary design run converges in an average of 30–40 generations. The objective function value for an evolutionary design run is plotted against generations in Fig. 14, with solid line for the best and dotted line for the mean values respectively. The evolution of the best fitness value shows a rather rapid convergence of the GA, while the relatively constant value of the mean fitness shows that population diversity is maintained along the evolutionary process.

Further on, the design illustration of low-IF analog filters for multi-mode wireless receivers is presented. The design goal was to fit the analog filter characteristics against the blocker profile from Fig. 1, with an additional objective to minimize the implemented filter order. The software-driven automatic reconfiguration of the low-IF analog filter for the GSM cellular communication standard resulted in a 4th order filter implemented with a cascade of Tow-Thomas biquads. The corresponding frequency characteristics are plotted in Fig. 15.

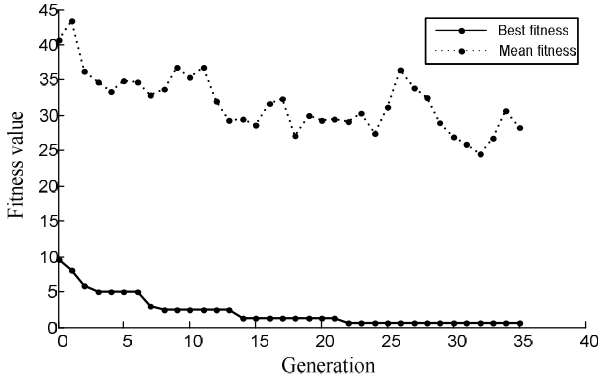

Fig. 14. Evolution of the objective function against the number of generations, for the sizing of an FPAA filter.

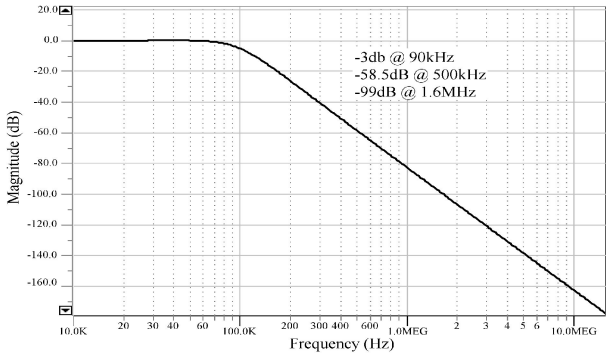

Fig. 15. Frequency characteristics of the GSM low-IF analog filter automatically synthesized on the proposed FPAA.

For low-IF filtering for the WCDMA cellular communication standard, the automatic reconfiguration algorithm generated a 6th order LF analog filter structure. The corresponding frequency characteristics are plotted in Fig. 16.

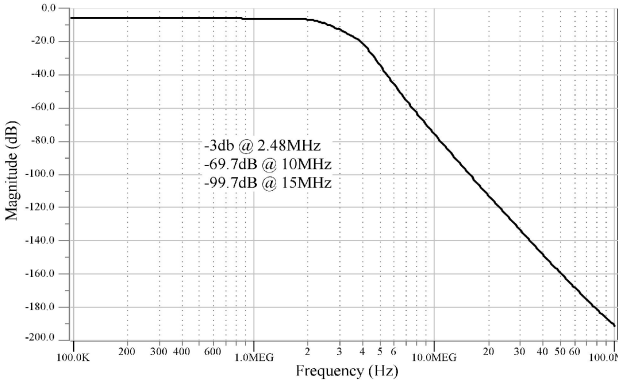

Fig. 16. Frequency characteristics of the WCDMA low-IF analog filter reconfigured on the proposed FPAA.

Figures 15 and Fig. 16 validate the proposed FPAA and software-driven reconfiguration algorithm for reconfigurable analog filtering in wireless communications. The evolution-oriented synthesis of the analog filters generated smaller filter orders than commonly used in practice. However, both figures illustrate a certain deviation in meeting the design specifications of the analog IF filters. This stems from the definition of the GA objective function in (6) which defines a very steep roll-off of the filter characteristics among adjacent sub-channels. One possible solution to compensate for this drawback is to define a

custom transition band between the blocking sub-channels.

To be noticed is that the automatically synthesized GSM and WCDMA analog IF filters are all-pole approximations of the blocker profile. The analog filter characteristics could further be improved by allowing arbitrary transmission zeros to the transfer function. This however assumes an extension to the proposed FPAA in order to allow the addition of the transmission zeros.

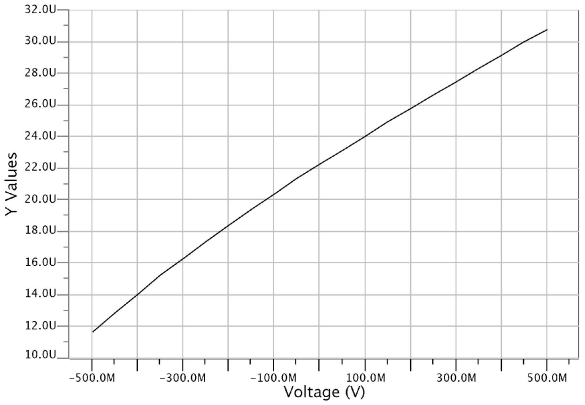

Fig. 17. Variation of the unit transconductance G_{mu} vs. the differential control voltage V_C.

The variation of the unit transconductance G_{mu} by means of the control voltage V_C is the fundamental adjustment imposed by the self-repair mechanism. Figure 17 shows the simulated variation of the unit transconductance against a 1 V change of the control voltage. The 1 V range of the control voltage leads to approximately ±50 % adjustment of the unit transconductance, VCO oscillation frequency and consequently filter cut off frequency. This variation range given by the continuous G_{mu} adjustment is sufficient for a wide band compensation of the typical frequency errors.

VI. CONCLUSIONS

This paper proposed an implementation for an auto-adaptive analog filter for next-generation intelligent multi-mode wireless radio front-ends. The adaptive filter was built around an FPAA developed to allow a variable filter order and filter approximation function, exhibiting four independent degrees of freedom for analog filter design: topological reconfiguration, filter band selection via a binary weighted transconductor array, continuous adjustment of the transconductance via a PLL-based self-calibration block, and a fine-grain variation of the capacitance values via a virtual capacitance scaling circuit.

The major contributions of this paper are an improved and more general software-driven reconfiguration algorithm, and a hierarchical self-repair mechanism. The novel reconfiguration algorithm performs transfer function design and FPAA filter synthesis simultaneously, thus being able to provide partial solutions along the reconfiguration process. Topological reconfiguration was achieved via a LUT. The shape of the filter characteristics is then determined via evolutionary computation which performs a curve-fitting operation. Finally, the hierarchical self-repair mechanism operates continuously to compensate post-silicon PVT variations and technological drifts.

The proposed implementation for auto-adaptive analog filter was validated with extensive simulation, illustrating its functionality on the specific design examples of GSM and WCDMA IF analog filters. The wireless communication analog filters however exhibit some slight deviations from the imposed specifications, stemming from a rather constrained definition of the objective function.

As a future work, we propose to define a customized objective function for the evolutionary analog filter design problem, in order to eliminate the slight deviations from the imposed design specifications, stemming from a rather constrained definition of the objective function. Also, we propose the extension of the FPAA as well as the reconfiguration algorithm to also implement arbitrary transmission zeroes.

REFERENCES

[1] P. Farago, G. Csipkes, D. Csipkes, S. Hintea, "An evolutionary approach to design and calibration of reconfigurable OTA-C filters", in *Proc. 35th Int. Conf. Telecommunications and Signal Processing*, Prague, Czech Republic, 2012, pp. 363–368.

[2] F. Sandu, V. Sandu, S. Nan, A. I. Dumitru, "Data acquisition and virtual instrumentation system for the study of Peltier and Seebeck effects", in *Proc. 12th Int. Conf. Power Electronics and Electrical Engineering (OPTIM 2010)*, 2010, pp. 903–908.

[3] A. Sirbu, G. Breaban, I. Cleju, I. Bogdan, "Improved genetic algorithm for the bandwidth maximization in TDMA-based mobile Ad Hoc networks", *Elektronika ir Elektrotechnika*, vol. 19, no. 7, pp. 104–109, 2013,. [Online]. Available: http://dx.doi.org/10.5755/j01.eee.19.7.1991

[4] *Dual-Band PADs for Next-Gen Mobile Devices*, TriQuint Semiconductor, Microwave Journal, 2012. [Online]. Available: http://www.microwavejournal.com/articles/17675-dual-band-pads-for-next-gen-mobile-devices.

[5] D. Csipkes, G. Csipkes, S. Hintea, H. Fernandez-Canque, "An analog array approach to variable topology filters for multi-mode receivers", *Elektronika ir Elektrotechnika*, no. 9, pp. 43–48, 2010.

[6] G. Csipkes, D. Csipkes, S. Hintea, P. Farago, "A 5th order current mode OTA-C low pass filter with programmable corner frequency", in *Proc. 2011 Int. Conf. Applied Electronics*, Pilsen, Czech Republic, 2011, pp. 71–74.

[7] M. M. Ahmadi, R. Lotfi, M. Sharif-Bakhtiar, "New architecture for Rail-to-Rail input constant-Gm CMOS operational transconductance amplifiers", *ISLPED*, 2003, pp. 353–358.

[8] J. Kardontchik, *Introduction to the design of transconductor-capacitor filters*. Kluwer, 1992. [Online]. Available: http://dx.doi.org/10.1007/978-1-4615-3630-7

[9] J. Aguado Ruiz, A. Lopez Martin, J Ramirez Angulo, "Three novel improved CMOS capacitance scaling schemes", in *IEEE Int. Symposium on Circuits and Systems*, 2010, pp.1304–1307.

[10] Z. Y. Chang, D. Haspeslagh, J. Verfaillie, "A highly linear CMOS - bandpass filter with on-chip frequency tuning", *IEEE Journal of Solid-State Circuits*, vol. 32, no. 3, pp. 388–397, 1997. [Online]. Available: http://dx.doi.org/10.1109/4.557637

[11] Y. Kim, H. Yu, "Automatic tuning circuit for Gm-C filters", in *12th IEEE Int. Conf. Electronics, Circuits and Systems*, 2005, pp.1–4.

[12] J. Ramirez-Angulo, S. Ming-Shen, "The folded Gilbert cell: a low voltage high performance CMOS multiplier", in *35th Midwest Symposium on Circuits and Systems*, 1992, vol. 1, pp. 20–23. [Online]. Available: http://dx.doi.org/10.1109/MWSCAS.1992.271344

[13] P. Farago, L. Festila, S. Hintea, P. Soser, G. Csipkes, D. Csipkes, C. Ciufudean, "A novel filter design method based on greedy search and genetic algorithms", in *Proc. 11th WSEAS Int. Conf. Signal Processing, Computational Geometry and Artificial Vision, Proc. 11th WSEAS Int. Conf. Systems Theory and Scientific Computation, Recent Advances in Signal Processing, Computational Geometry and Systems Theory*, Florence, Italy, 2011, pp. 152–157.

[14] J. Grimbleby, "Hybrid genetic algorithms for analogue network synthesis", in Proc. *IEE/IEEE Int. Congress on Evolutionary Computation*, 1999, vol. 3, pp. 1781–1787.