Manuscript received Jun. 11, 2007; revised Sep. 15, 2007

GP-based Design and Optimization of a Floating Voltage Source for Low-Power and Highly Tunable *OTA* Applications

MARYAM SHOJAEI BAGHINI[*], RAJENDRA KANPHADE[**], D. G. WAKAD*E* [**]
PRITI GAWANDE[**], MANISHA CHHANGANI[**], MANISH PATIL[**]
[*]EE Department of IIT Bombay, INDIA
[**]VLSI & Embedded System Design Center, SSGMCE, Shegaon, INDIA
mshojaei@ee.iitb.ac.in, rdkanphade@gmail.com, priti.gawande@gmail.com,
manisha.chhangani@gmail.com, manispatil1@gmail.com
[*]www.ee.iitb.ac.in, [**] http://www.ssgmce.org

Abstract—Reuse of analog building blocks is a time consuming process as CMOS technology scales down. Therefore automatic sizing while taking care of second order effects is of great importance. In this paper a method for automatic sizing and optimization of a floating voltage source (FVS) used in a CMOS Operational Transconductance Amplifier (OTA) is presented. The optimization determines the optimal component values and transistor dimensions for FVS in order to minimize the dissipated power and output impedance. The presented methodology uses geometric programming (GP) and simulation-based optimization in a time-efficient manner. The CMOS FVS is sized initially using convex optimization. Then the design is further optimized by a simulation-based circuit optimizer to include second order effects. Since the initial design uses GP method a globally optimum solution is obtained. The presented approach uses MATLAB version 7.1.0.246 and Cadence Analog Circuit Optimizer. The results are verified by detailed analog simulation using Cadence Analog Design Environment (ADE from IC 5.0.33) in 0.35um mixed-mode CMOS process.

Key-Words: - Operational Transconductance Amplifier, Convex Optimization, Geometric Programming, Simulation-based Optimization.

1 Introduction

Fast and optimum redesign of analog building blocks in deep sub-micron CMOS technologies is crucial in the IC industry when migrating from a technology to another technology. Besides standardization of analog specifications is not practical because the analog circuit needs to be redesigned as desired specifications change [1]. Therefore, the design effort of a given block for a different technology requires the work of an expert analog designer to provide, all the equations containing the knowledge of the adopted topologies. On the other hand, the analog expert should also provide the design criteria for the optimization of these blocks.

In circuit design optimization, a circuit and its performance specifications are given and the goal is to automatically determine the device sizes in order to meet the given performance specifications while minimizing a cost function, such as a weighted sum of the active area or power dissipation.

Existing approaches of automatic circuit sizing are broadly classified into four main categories [1-3]. These approaches are:

- Classical optimization Methods.
- Knowledge-Based Methods.
- Global Optimization Methods.
- Geometric Programming Methods.
- Designer-driven Stochastic Multi-objective Optimization Method.

The main disadvantage of the *classical optimization methods* is they only find locally optimal designs. Therefore small variations of any of the design parameters results in a worse (or infeasible) design.

The *knowledge – based methods* find a locally optimal design (or, even just a “good” or “reasonable” design) instead of a globally optimal design. The final design depends on the initial design chosen and the algorithm parameters.

The *global optimization methods* are useful when there is no proper modeling of the integrated circuit components. However they are slow if the entire design space needs to be searched. Evolutionary approaches like genetic algorithms improve the speed. However in practice they cannot guarantee a globally optimal solution.

Geometric Programming (GP) methods can solve large problems, with thousands of variables and tens of thousands of constraints, very efficiently (in minutes on a small workstation). The other main advantage is that the methods are truly global, i.e. the global solution is always found, regardless of the starting point (which, indeed, need not be feasible) and infeasibility is unambiguously detected [3]. GP needs expert designer knowledge to introduce the constraints in a special form.

During preparation of this paper we found a recently reported method called "Designer-driven Stochastic Multi-objective" optimization method. This method uses designer knowledge and requires approximate equations [4].

The method, which is presented in this paper, takes advantage of GP-based and simulation-based optimization to balance speed-accuracy tradeoff. The method can be applied to a wide variety of analog functional modules. Here the presented method is applied to a cross-coupled CMOS OTA. OTA is an analog element which is used for wireless and video signal processing functions in SOC (System On Chip) and FPAA (Field Programmable Analog Array) applications with one important feature, i.e. external tunability/programmability, which is not available in operational amplifiers.

2 Geometric Programming (GP)

To formulate the analog design problem in geometric programming each constraint has to be converted in the form of monomial or posynomial.

2.1 Monomials and Posynomials

Let x_1, x_2, … x_n denote n real positive variables, and $x = (x_1, x_2 \ldots x_n)$ a vector with components x_i. A real valued function f of "x", with the form

$$f(x) = cx_1^{a_1} x_2^{a_2} \ldots\ldots\ldots x_n^{a_n} \tag{1}$$

where c > 0 and $a_i \in R$ is called a *monomial function*, or more informally, a monomial (of the variables x_1, x_2… x_n). The constant "c" is referred as the coefficient of the monomial, and the constants "a_i" are referred as the exponents of the monomial. As an example, $9.5x_1x_2^{-0.1}$ is a monomial of the variables x1 and x2, with coefficient 9.5 and x_2-exponent -0.1. Monomials are closed under multiplication and division: if f and g are both monomials then so are fg and f/g. (This includes scaling by any positive constant.) A monomial raised to any power is also a monomial [5].

The term "monomial", as used in the context of geometric programming is similar to, but differs from the standard definition of "monomial" used in algebra. In algebra, a monomial has the form (1), but the exponent "a_i" must be nonnegative integers, and the coefficient "c" is one.

A sum of one or more monomials, i.e., a function of the form

$$f(x) = \sum_{k=1}^{K} c_k x_1^{a_{1k}} x_2^{a_{2k}} \ldots\ldots\ldots x_n^{a_{nk}} \tag{2}$$

where $c_k > 0$, is called a *posynomial function* or, more simply, a posynomial (with k terms having the variables x_1… x_n). The function (or expression) "$5.8x^2yz^{-1}$" is a monomial (hence, also a posynomial). The function *"$(xz+yz)^2$" is* a posynomial but not monomial [6].

2.2 Standard Form Geometric Program

A geometric program (GP) is an optimization problem of the form

$$\text{Minimize} \quad f_0(x)$$
$$\text{Subject to} \begin{cases} f_i(x) \le 1, i = 1, \ldots\ldots, m, \\ g_i(x) = 1, i = 1, \ldots\ldots, p, \\ x_i > 0, i = 1, \ldots\ldots, n. \end{cases} \tag{3}$$

where f_i are posynomial functions, g_i are monomials, and "x_i" are the optimization variables. (There is an implicit constraint that the variables are positive, i.e., $x_i > 0$.) We refer to the problem (3) as a geometric program in standard form. In a standard form GP, the objective must be posynomial (and it must be minimized); the equality constraints can only have the form of a monomial equal to one, and the inequality constraints can only have the form of a posynomial less than or equal to one [5].

2.3 Geometric Programming in Convex Form

The main trick to solving a GP efficiently is to convert it to a nonlinear but convex optimization problem, i.e., a problem with convex objective and inequality constraint functions, and linear equality constraints. Efficient solution methods for general convex optimization problems are well developed. This results in the problem

$$\text{minimize} \ \log f_0(e^y)$$
$$\text{subject to} \ \log f_i(e^y) \le 0, i = 1, \ldots\ldots\ldots, m,$$
$$\log g_i(e^y) = 0, i = 1, \ldots\ldots\ldots, p \tag{4}$$

with variable "y", where $y_i = \log x_i$. Here we use notation e^y, where y is a vector, to mean component wise exponentiation: $(e^y)_i = e^{yi}$. The transformed version, given in relation (4), is the so-called *convex form* of the geometric program (3). Unlike the original GP, this convex form can be solved very efficiently [5].

Advantages of convex optimization are:

- They can solve problems with thousands of variables and constraints, very efficiently.
- The global solution is *always* found in these methods, regardless of the starting point. Infeasibility is also detected, i.e., if the methods do not produce a feasible point they produce a certificate that proves the problem is infeasible. Also, the stopping criteria are completely nonheuristic, i.e. at each iteration a lower bound on the achievable performance is obtained.

3 Operational Transconductance Amplifier (OTA)

OTA is an excellent current mode module due to its inherent wide band capability. One of the features of OTA's is that its transconductance can be programmed or tuned, for example either by varying the analog bias voltage/current or by changing the gain of the current mirror used inside OTA.

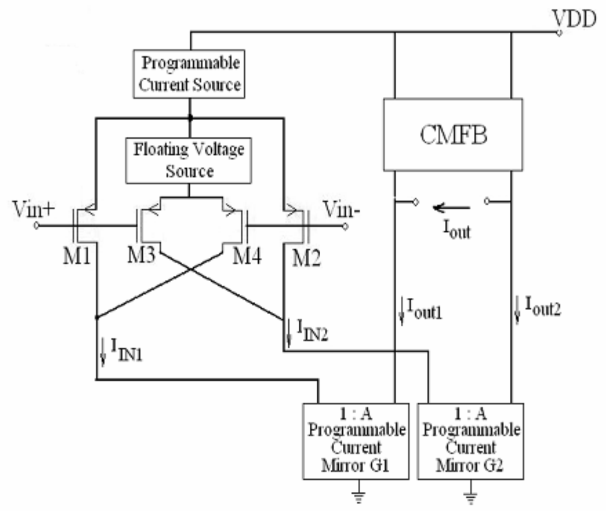
VDD
Programmable Current Source
Floating Voltage Source
CMFB
Vin+
Vin-
M1
M3
M4
M2
I_{out}
I_{IN1}
I_{IN2}
I_{out1}
I_{out2}
1 : A Programmable Current Mirror G1
1 : A Programmable Current Mirror G2

Fig. 1. Complete OTA Architecture

The programming feature of OTA helps the designers to realize the analog functional circuits of which the performance specifications can be configured as per the application requirements [6].

The OTA module, considered in this paper, is based on two cross-coupled differential MOS pairs, of which the complete schematic diagram is shown in Fig. 1. Cross-coupled topology of the transconductor circuit does not introduce additional internal nodes, resulting in improved linearity without high-frequency performance degradation.

The simplified schematic diagram of the cross-coupled differential MOS pairs is shown in Fig 2. Two cross-coupled differential pairs with MOS devices M1–M4 are operating in saturation. Both pairs are biased by a dc current 'iss' in combination with an adjustable floating voltage Vb with low output resistance. It can be shown that differential transconductance of this OTA, Gm, is expressed as

$$Gm=K \times Vb \qquad (5)$$

where K=0.5μC_{ox}W/L is the transconductance parameter of the transistors M1-M4 and Vb is the voltage of the floating dc voltage source [6]. All undefined parameters have their usual meanings.

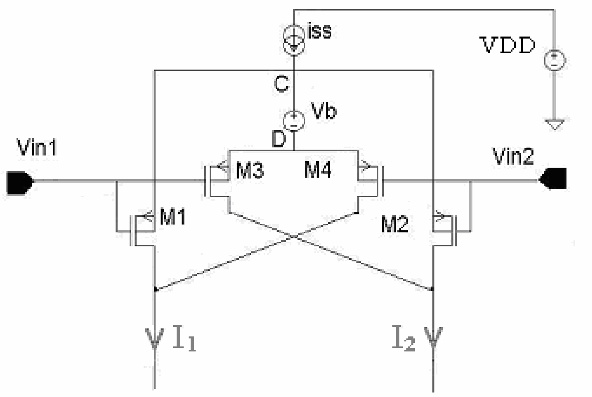

Fig. 2. Cross-coupled OTA

As per relation (5) value of Gm is changed by changing value of Vb. Therefore Vb should be a variable FVS with low output impedance in the entire range of Vb values.

The process specifications of 0.35um CMOS technology, used in this paper, are given in table 1.

In the concerned FPAA, floating voltage source Vb was required to change from 29mV to 460mV with Iss=19uA and minimum output impedance [7]. "Iss" is the current source used for the biasing of OTA as shown in Fig.2.

4 Floating Voltage Source (FVS)

In order to obtain a wide range of application frequencies in OTA-C filter design, it is necessary for the transconductance of the OTA to be adjustable. This is achieved using tunable FVS.

To preserve the high linearity of the transconductance, voltage source Vb needs to have low output impedance. Fig. 3 shows the FVS architecture which exhibits low output impedance with low power dissipation. This CMOS FVS consists of a differential pair with a shunt-series feedback. For simplicity biasing path from Vdd to the source of transistor Mvb5 is not shown. In Fig. 3 "V_{out+} - V_{out-}" is the amount of voltage shift, i.e. Vb.

Table 1. Process Specifications for 0.35um CMOS Technology

Parameter	Value
V_{DD}	3.3V
V_{SS}	0V
V_{Tn}	0.45V
V_{Tp}	0.65V
µnCox	158uA/V^2
µpCox	66.7uA/V^2

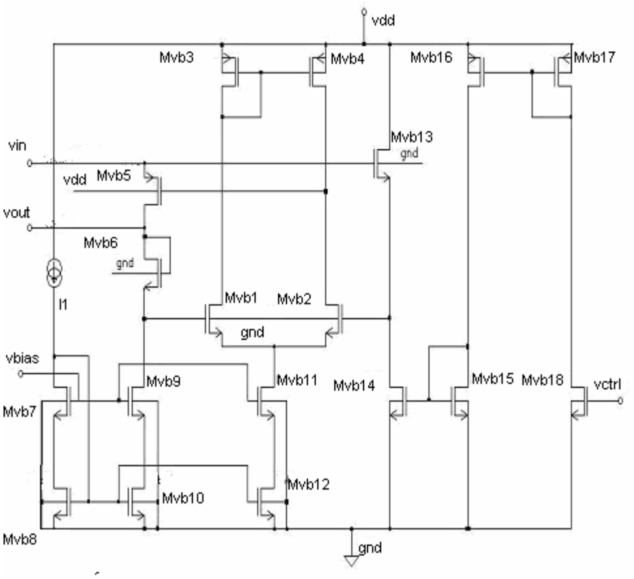

Fig. 3 Schematic Diagram of CMOS FVS

The dc output voltage of FVS depends on the gate-source voltage of transistors

$$V_b = V_{gs13} + V_{gs2} - V_{gs1} - V_{sg6} \tag{6}$$

AC analysis shows that the output resistance at the output terminals mainly depends on the value of output resistance of transistors Mvb5 and Mvb6.

$$R_{out} = \frac{R_{ob6} \| r_{o5}}{1+T} \tag{7}$$

where R_{ob6}=output resistance of cascade current source and *T* is Loop gain of FVS given by

$$T = A_V \times f \tag{8}$$

In relation (8) A_V is forward gain and *f* is feedback factor given by

$$A_V = Gm_2 (r'_{o2} \| r_{o4}) \tag{9}$$

$$f = gm_5 (r_{o5} \| R_{ob6}) \tag{10}$$

In relation (9) Gm_2 and r'_{o2} are transconductance and output impedance of differential pair Mvb1-Mvb2, respectively. From (6) to (10) it is clear that transistors Mvb1, Mvb2, Mvb5, Mvb6 and Mvb13 are the main design elements of FVS. The only condition applied to the FVS is that Vgs2 > Vgs1.

4.1 GP-based optimal design of FVS

GP-based design needs biasing conditions, dc voltage level and small signal specifications of FVS to be formulated. Importance here is that the constraints on the design variables are posynomial inequalities and hence, can be handled by GP.

Biasing conditions:

- **For NMOS:**

$$V_G - V_D \langle V_{thn} \qquad \text{where, } V_{thn} > 0$$

- **For PMOS:**

$$V_G - V_D \rangle V_{thp} \qquad \text{where, } V_{thp} < 0$$

- **Transistor Mvb1:**

$$V_{in} - V_{gs6} - (VDD - V_{sg3}) \langle V_{thn} \Rightarrow$$

$$\sqrt{\frac{I_{D3}}{\frac{1}{2}\mu_p C_{OX}\left(\frac{W}{L}\right)_3}} \langle V_{thn} - V_{out(max)} + V_{gs6} + V_{DD} - |V_{thp}|$$

- **Transistor Mvb2:**

$$V_{in(max)} - V_{gs13(min)} - \left(V_{in(max)} - V_{sg5}\right) \langle V_{thn} \Rightarrow$$

$$\sqrt{\frac{I_{D5}}{\frac{1}{2}\mu_p C_{ox}\left(\frac{W}{L}\right)_5}} \langle V_{thn} + V_{gs,13(min)} - |V_{thp}|$$

- **Transistor Mvb4:**

$$V_{DD} - V_{sg3} - \left(V_{in(max)} - V_{sg5}\right) \rangle V_{thp} \Rightarrow$$

$$\sqrt{\frac{I_{D3}}{\frac{1}{2}\mu_p C_{OX}\left(\frac{W}{L}\right)_3}} \langle -V_{thp} + V_{DD} - V_{in(max)} + V_{sg5}$$

- **Transistor Mvb5:**

$$V_{in} - V_{sg5} - (V_{in} - Vb_{min}) \rangle V_{thp}$$

$$\sqrt{\frac{I_{D5}}{\frac{1}{2}\mu_p C_{OX}\left(\frac{W}{L}\right)_5}} \langle -V_{thp} + V_{gs13(min)} - V_{gs6} - |V_{thp}|$$

Saturation conditions of Mvb1 and Mvb5 satisfy those of Mvb4 and Mvb2, respectively.

- **Output impedance:**

The output resistance is a posynomial (Rel (7)).

We developed a GP-based optimization file using posynomial constraints, derived above, to size the FVS transistors in MATLAB environment. Power dissipation minimization was chosen as the objective function. This optimization led to initial sizing of transistors of FVS, of which aspect ratios are given in Table 2. Sizing in MATLAB environment on a Pentium 4 PC took only a few seconds. This is because fast optimization speed is one of the features of convex optimization methods.

Table 2. Aspect Ratio of MOS Transistors of GP-Based Optimized FVS

Transistor	W/L (μm/ μm)	Transistor	W/L (μm/ μm)
Mvb1	19/1	Mvb10	1/1
Mvb2	19/1	Mvb11	3/1
Mvb3	1/1	Mvb12	3/1
Mvb4	1/1	Mvb13	1/1
Mvb5	300/1	Mvb14	1.3/1
Mvb6	0.8/1	Mvb15	2/1
Mvb7	3/1	Mvb16	1/1
Mvb8	3/1	Mvb17	3/1
Mvb9	1/1	Mvb18	1/1

4.2 Simulation Results of GP-Optimized FVS

The GP-based designed FVS was simulated using Cadence's Analog Design Environment in 0.35um CMOS technology with process specifications given in Table 1 and control voltage (V_{ctrl}) of 1V. Fig. 4 shows the simulation results of Matlab- optimized FVS. This FVS provides a voltage shift 27.6mV to 460.9mV (shown with label "VB" in Fig. 4) for control voltage varying between 1V to 2.3V ("V_{ctrl}" in Fig. 3) while the average output resistance is 158.7ohms. Power consumption changes from 30uW (supply current of 9uA) to 230uW (supply current of 69.7uA) in the entire range of VB.

4.3 Simulation-Based Optimization of FVS

Since GP modeling does not consider the second order effects simulation-based optimization is used at the next step. For this purpose Cadence Optimizer is used. The initial values of the design variables are taken from MATLAB optimization output. The Cadence optimizer first determines how the values of the goal expression vary as design variables change. Then the optimizer updates the design variables in a manner to move the values of the expression in the direction of goals. The optimizer simulates the circuit with updated values to check the outcome. If stopping criteria are not met, the optimizer iterates through the optimization process.

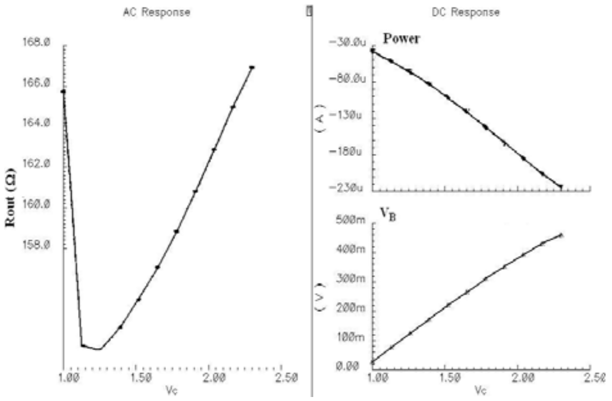

Fig. 4 Simulation Results of GP-Optimized FVS

In Cadence's Optimizer output impedance and DC power consumption minimization are set as optimization goals. W1, W3, W5, I_S (bias current of differential pair of FVS) and I_{D5} are considered as design variables. Fig. 5 shows the Cadence's optimizer waveform window at minimum control voltage (V_{ctrl}=1). The left side of the Fig. 5 shows how the value of goals, i.e. average current consumption and output impedance of the FVS, is changing with iterations. Right side of Fig. 5 shows how design variables change to achieve the desired goals. As Fig. 5 shows optimizer reduces output impedance iteratively while power dissipation is almost unchanged. This is based on the complete modeling of transistors given by technology model files in 0.35um CMOS process. Final aspect ratios of the transistors of FVS are given in Table 3.

4.4 Simulation Results of Final FVS

The final FVS was simulated using Cadence's Analog Design Environment in 0.35um CMOS technology. Fig. 6 shows the simulation results of FVS. This FVS provides a voltage shift 27.33mV to 460.6mV for control Voltage varying between 1V to 2.3V while the average output resistance has reduced to 131 ohms. Power consumption is almost as the same as that of GP-based design.

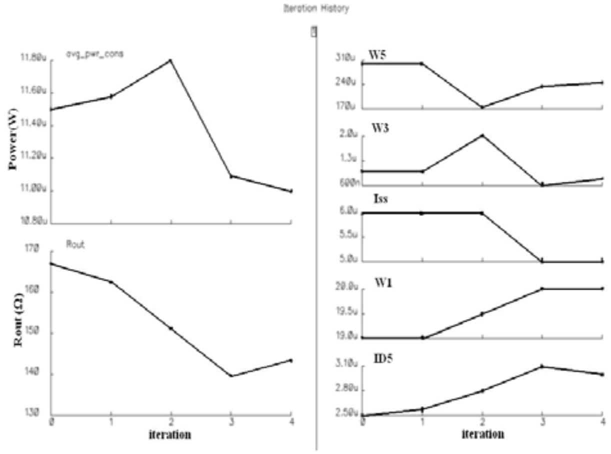

Fig. 5. Cadence Analog Circuit Optimizer's Waveform Window for FVS

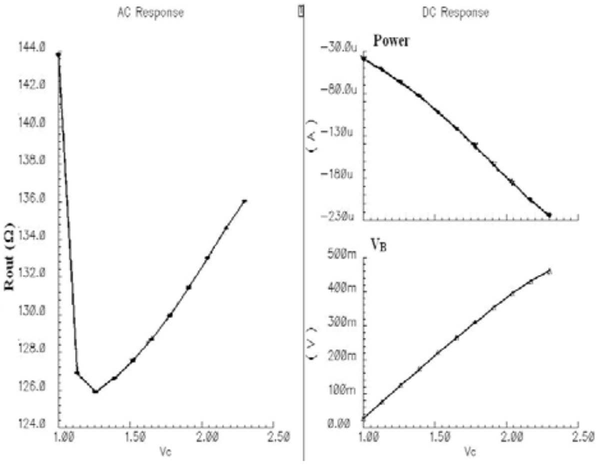

Fig. 6. Simulation Results of Optimized FVS

Table 3. Final Aspect Ratio of MOS Transistors of Cadence-Optimized FVS

Transistor	W/L	Transistor	W/L
Mvb1	20/1	Mvb10	1/1
Mvb2	20/1	Mvb11	3/1
Mvb3	0.8/1	Mvb12	3/1
Mvb4	1/1	Mvb13	1/1
Mvb5	245/1	Mvb14	1.3/1
Mvb6	0.8/1	Mvb15	2/1
Mvb7	3/1	Mvb16	1/1
Mvb8	3/1	Mvb17	3/1
Mvb9	1/1	Mvb18	1/1

5 Conclusions

In this paper automatic design and optimization of a floating voltage source for a highly tunable and programmable OTA application in 0.35um CMOS technology was presented. FVS was first designed using GP-based optimization in MATLAB environment to get a global optimal design with respect to power dissipation. The initial design was further optimized using a simulation-based environment (here Cadence Circuit Optimizer) to take into account the second order effects. In this way a fast, automatic and optimization-based design approach was achieved. Automated and globally-optimized low-power design is desirable as power requirements of integrated circuits become more stringent for portable and battery-operated devices.

6 Future Work

The method presented in this paper provides an efficient automatic analog circuit sizing. As an initial attempt an automatic sizing procedure can be implemented by automatically linking MATLAB to a circuit simulation-based optimizer. Automatic generation of GP model and its derivatives for nano scale devices will have an impact on analog design and reuse in future technologies.

References:

[1] L. Laberk, T.Tixer, Y. Fellah and N. Abouchi, "A New Approach for Automated Analog Design Optimization", *Proc. of WSEAS Intl. Conf. on Circuits*, 2007, pp 17-21.

[2] T. Chen, Y. Cheng, "Global Optimization Using Hybrid Approach", *Proc. of WSEAS Int. Conf. on Sim, Modeling and Optimization*, 2007, pp.309-313.

[3] M.M.Hershenson, S.P.Boyd and T.H. Lee "Optimal Design of a CMOS Op-Amp via Geometric Programming" in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 1, 2001, pp. 1-21.

[4] V. Aggarwal, U.M. O'Reilly, Multi-objective optimization theory and circuit optimization, http://web.mit.edu/varun_ag/www/sizing.html,2007.

[5] S. P. Boyd, S. J. Kimy, L. Vandenberghe and A. Hassibi, "A Tutorial on Geometric Programming", ISL, EE Dept., Stanford Univ, Stanford, CA, 02/05.

[6] Pankiewicz B., Wojcikowski M., Szczepanski S., and Sun Y., "A FPAA for CMOS continuous-time OTA-C filter applications," *IEEE J. of Solid-State Circuits*, Vol. 37, No. 2, 2002, pp. 125-136

[7] R. D. Kanphade, M. Shojaei Baghini, D. G. Wakade, M. Chhangani, M. V. Patil, S. M. Ranjan, J. R. Verma, N. K. Ingole and P. Gawande "Design of FPAA using custom IC and optimization-based design flow", *Proc. of CDNLive'06*, USA, 2006.