

Automated Design of a Second Order Sigma Delta Modulator using Success History Based Adaptive Differential Evolution Algorithm

Shravan Kudikala¹ and Samrat L. Sabat²

*Centre for Advanced Studies in Engineering Science and Technology, University of Hyderabad,
Hyderabad, Telangana, India.*

¹Orcid: 0000-0001-6687-0154, ²Orcid: 0000-0002-1197-2943

Abstract

The optimization based automated analog design approach requires a global optimization technique that should satisfy both high success rate and high convergence rate for minimizing multiple design iterations. The traditional population-based optimization algorithms meet both the requirements. However, their performance depends on the values of the control parameters. Tuning the control parameters for a particular design is a tedious task. In this paper, a relatively new variant of Differential Evolution (DE) algorithm namely Success History based Adaptive Differential Evolution (SHADE) algorithm is applied to automate circuit design using simulator-in-loop optimization approach. The SHADE algorithm adapts its control parameters during the optimization run. The performance of the SHADE algorithm is compared with the DE on the success rate and convergence rate. The SHADE is applied for automatic design of a second order Sigma-Delta Modulator (SDM) to maximize its Signal to Noise Ratio (SNR) in a two-stage optimization process. In the first stage, a system level optimization is performed to obtain the integrator's specifications and in the second stage, the (two stage and folded cascode) OPAMPs are designed for the required specification posed by the first stage. TSMC's 180 CMOS process technology devices are considered for the SDM design with the supply voltage of 3.3 volts. The DC, AC, and transient simulation results confirm the effectiveness of the synthesized design.

Keywords: Analog Synthesis; Differential Evolution; CMOS; Success History based Adaptive Differential Evolution; Global optimization; Sigma-Delta Modulator; OPAMP Non-idealities; System level synthesis

INTRODUCTION

Speed, power, and area are the key factors that drive the digital technology to the advanced process nodes. Design migration, redesign or custom cell design is required for a new process node. The digital designs are usually migrated, and it is a heavily automated flow. Analog designs require manual intervention and it is complex, designer intensive and time-

consuming process [1]. In the analog design flow, transistor level circuit sizing is the most important stage where the key constraints, i.e., area, power are optimized. Further, analog circuits must be optimized for other key performances such as DC gain, slew rate, bandwidth, phase-margin, supply rejection, etc. besides area and power. Thus, analog designers face more complex tradeoffs compared to the digital counterpart. In addition, the degrees of freedom for an analog design are larger and usually interdependent. Therefore, the optimization task for the designers becomes increasingly difficult for newer process nodes. The complexity further scales up with the reduced supply voltages and increased short-channel effects.

Analog circuits can be synthesized using either manually or through an automated design process. Traditionally, manual design approach makes use of topology, device specific design equations to obtain the coarse component values such as transistor's width and length, resistance, capacitance, inductance, bias voltages, and currents. These values are further fine-tuned by the designer's intuition with SPICE accurate verifications. The efficiency of the design depends on the skill and expertise of the designer. In contrast, automatic design of analog circuits has the capability to handle the complexity in a short time with least manual intervention. In an automated design/sizing method, the circuit design problem is formulated as a mathematical optimization problem. The goal of the optimization is to find the component sizes that satisfy the design specification. The component values are chosen as the optimization variables and the performances of the corresponding set of variables are obtained either through design equations [2], [3] or the SPICE verification itself [4], [5]. A detailed description of various architectures for circuit synthesis was surveyed in [6]. The advantage of using SPICE simulator inside the optimization loop is that it is accurate and requires less setup time. However, the disadvantage is its run time. For each specification, it requires specific test bench and analysis to be evaluated. The entire optimization process including SPICE simulator in the loop can last from hours to days depending on the complexity and specifications of the design.

In the automatic synthesis approach, local or global search techniques can be used depending on the time requirement. Local search techniques are computationally fast, but the solution is not guaranteed as they are prone to be trapped in a local minimum, and the final solutions depend on the initial solution. Global search techniques like genetic algorithm [7], differential evolution, particle swarm optimization [8] have been used in the analog circuit synthesis. Although, these techniques take a longer time, they give the final global best solution. Among the existing population-based search techniques, DE is proved to be efficient in providing quality solution [9].

The synthesis approach using global search techniques, may not be successful in the very first attempt. Numerous runs have to be carried out to adjust the parameters of the synthesis process. These parameters can be classified into two categories: 1) Circuit related and 2) Optimizer related. For the population based optimization algorithms, the convergence is strongly dependent on the control parameters specific to the algorithm. Thus, there might be situations where the synthesis failures are due to the improper algorithmic settings thereby increasing the overall design time. Automatic selection (adaptation) of the control parameters is required to make the designer independent of algorithmic settings. It will help the designer to concentrate on the circuit related parameters and other advanced issues. Thus in this work, we propose to use a relatively new adaptive version of Differential Evolution (DE) algorithm namely SHADE (Success History based Adaptive Differential Evolution) for automatic analog circuit design. The performance of the SHADE algorithm is verified by sizing a folded-cascode amplifier and its rate of convergence and consistency is compared with the standard DE algorithm.

Further, the SHADE algorithm is used to demonstrate the hierarchical design of a second order Sigma Delta Modulator (SDM). The SDM is an indispensable block in modern high resolution low to medium speed analog to digital converters. Over sampling and noise shaping are the two important features that distinguishes SDM from conventional Nyquist rate data converters. Despite the advantages of SDMs, they are sensitive to non-idealities of the analog building blocks. These non-idealities include finite gain, bandwidth, slew-rate and output range of amplifier blocks among others. These non-idealities do not affect the performance of the SDM if they are sufficiently high. Minimizing the overall power and area requirements of the SDM certainly poses an optimization problem with sufficient "idealness" of the amplifier blocks as constraint. A flat optimization method of SDM is extremely time consuming especially with global optimizers. Thus, we have considered a hierarchical two stage optimization process where in the first stage the SDM is optimized with macro-models of integrator blocks modeled with non-idealities. In this stage the requirements (DC gain, bandwidth, etc.) of amplifiers are minimized implying minimization of power, area and non-idealities. In the second stage circuit level

optimization is carried out with the specifications obtained from the first stage.

The remainder of the paper is organized as follows: In section II, we have presented basics and mathematical description of the circuit synthesis problem. In Section III, we have presented the steps involved in DE and the adaptation of control parameters of SHADE algorithms. Further, we have compared both the algorithms using multiple run result of the synthesis of a folded cascode amplifier. In Section IV, basics of the sigma-delta modulator and the design parameters, realistic amplifier macro-models, optimization approach and the formulation are presented. In Section V, we have presented the results of system and circuit level synthesis of the SDM and its sub-circuits. Section VI concludes the paper.

CIRCUIT SYNTHESIS PROBLEM

Analog circuits are usually part of more complex subsystems such as Phase locked loops or data converters. A hierarchical design approach of the subsystem is sought for simplicity where macro-models of the analog circuits are used instead of the transistor level descriptions. The functionality of the subsystems is verified and optimized to yield circuit level specifications. A proper circuit topology is chosen from already known structures (e.g. Differential folded-cascode, two stage amplifier) based on the specifications. Circuit *synthesis* is a process of generating component values that satisfy user-specified performances. It is the converse operation of *analysis*, where performances are obtained for a given set of component values. In this work, we have followed the circuit synthesis model, mostly used in the literature as shown in Fig. 1. The candidate circuit designs are generated by a trial-and-error routine defined in the *optimization engine* and are evaluated iteratively in the performance *evaluation engine*. The optimization engine improves the candidate solutions to meet the desired specifications with the progress in iterations.

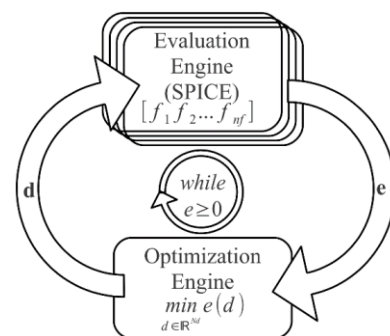


Figure 1: Optimization Flow of Analog Circuit Synthesis

Mathematically, the performances of the circuit under consideration for sizing can be characterized by a vector **f** of

dimension N_f , where $\mathbf{f} = (f_1, f_2, \dots, f_{N_f})$. f_i for $i = 1 \dots N_f$ are the performances of interest (DC gain, unity gain frequency, phase margin, etc). These performances are dependent on a vector of designable parameters \mathbf{d} of dimension N_d , where $\mathbf{d} = (d_1, d_2, \dots, d_{N_d})$. d_i for $i = 1 \dots N_d$ are the physical dimensions (Width, Length) of the devices and also can be electrical quantities like bias voltages and currents of the circuit. In general, the analog circuit synthesis problem is a multi-objective, multi constraint optimization problem, solved by minimizing the error function $e_i(\mathbf{d})$ defined as

$$e_i(\mathbf{d}) = \begin{cases} \frac{f_i(\mathbf{d})}{S_i^{ub} + \varepsilon} - 1 & \text{for upper bound specs} \\ 1 - \frac{f_i(\mathbf{d})}{S_i^{lb} + \varepsilon} & \text{for lower bound specs} \\ \frac{|f_i(\mathbf{d}) - S_i^{eq}|}{\varepsilon} - 1 & \text{for equality specs} \end{cases} \quad (1)$$

where e_i , S_i^{ub} and S_i^{lb} corresponds to the error function, upper and lower bound specifications of i^{th} performance indicator f_i . ε is a limiting factor. The circuit sizing can be formulated as a multi-objective constrained problem as:

$$\begin{aligned} \min_{\mathbf{d}} \{ \dots e_i(\mathbf{d}) \dots \}, \quad i = 1, 2, \dots, N_f \\ \text{s.t. } g_j(\mathbf{d}) \leq 0, \quad j = 1, 2, \dots, l \end{aligned} \quad (2)$$

where, $g_j(\mathbf{d})$ is the j^{th} constraint (e.g. Power, Area). It can be solved using single objective optimization algorithm by converting (2) to a scalar quantity as

$$e(\mathbf{d}) = \sum_{i=1}^k w_i \cdot e_i(\mathbf{d}) + \sum_{j=1}^l w_j \cdot g_j(\mathbf{d}) \quad (3)$$

where w_i and w_j are constant weights assigned to corresponding error functions and constraints respectively. Further, \mathbf{f} also depends on non designable parameters such as process specific variable vector \mathbf{z} (i.e., model parameters of the device) and environment variable vector θ (i.e., operating temperature and supply voltages), thus $\mathbf{f}(\mathbf{d}) \equiv \mathbf{f}(\mathbf{d}, \mathbf{z}, \theta)$.

DE AND SHADE ALGORITHMS

Differential Evolution has been successfully applied for the circuit synthesis problem and proved to be an efficient alternative to other existing evolutionary algorithms [10], [11]. The major steps for implementing the DE algorithm are (i) initialization of the population (ii) mutation (iii) crossover and (iv) selection process.

The DE algorithm starts with a random initial population in

the search space as (4);

$$d_{j,i} = d_j^{min} + rand_j[0,1] \cdot (d_j^{max} - d_j^{min}) \quad (4)$$

where $i = 1, \dots, NP$ (NP = population size) and $j = 1, \dots, n_d$ (n_d = number of circuit design parameters), d_j^{min} and d_j^{max} are the lower and upper bounds of the j^{th} design parameter respectively. Once every vector of the population has been initialized, its corresponding fitness value is calculated and stored in the memory. The mutation, crossover and selection operations are performed to create the population for the next iteration/generation $P^{(G+1)}$ using the current population $P^{(G)}$. In an iteration/generation, each member of the population is a target vector $\mathbf{d}_i^{(G)}$, and it is compared with a mutant vector. The mutation operator generate mutant vectors $V_i^{(G)}$ according to (5) by perturbing a randomly selected vector \mathbf{d}_{r_1} with the difference of two other randomly selected vectors \mathbf{d}_{r_2} and \mathbf{d}_{r_3} .

The selected vector indices are in the range of 1 to NP . The scaling factor F is used for amplifying the difference vectors and typically chosen within the range of 0 to 1.

$$V_i^{(G)} = \mathbf{d}_{r_1}^{(G)} + F \cdot (\mathbf{d}_{r_2}^{(G)} - \mathbf{d}_{r_3}^{(G)}) \quad (5)$$

After generating the mutant vector, crossover operation is performed to enhance the population diversity. The mutant vector exchanges its components with the target vector $\mathbf{d}_i^{(G)}$ to generate a trial vector $U_i^{(G)}$ as (6).

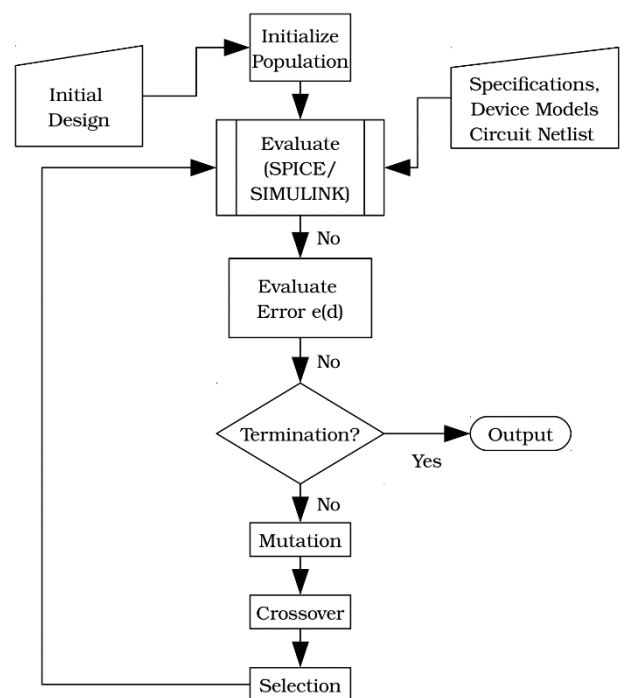


Figure 2. Flow diagram of the Differential Evolution algorithm based circuit synthesis

$$U_i^{(G)} = u_{j,i}^{(G)} = \begin{cases} v_{j,i}^{(G)} & \text{if } \text{rand}_j(0,1) \leq CR \text{ or } j = j_{\text{rand}} \\ d_{j,i}^{(G)} & \text{otherwise} \end{cases} \quad (6)$$

In the next step, the algorithm uses selection operator to keep the population size constant over subsequent generations as (7). This step decides whether the target or trial vector survives to next generation.

$$d_i^{(G+1)} = \begin{cases} U_i^{(G)} & \text{if } e(U_i^{(G)}) \leq e(d_i^{(G)}) \\ d_i^{(G)} & \text{otherwise} \end{cases} \quad (7)$$

where $e(\mathbf{d})$ is the objective function to be minimized. If the new trial vector yields an equal or lower value of the objective function, it replaces the corresponding target vector in the next generation; otherwise, the target vector is retained in the population. These steps are repeated till one of the termination criteria, i.e.: a) maximum number of generations or b) $e(\mathbf{d}) \leq 0$ or c) change in the value of the objective function for certain iteration is less than a specified tolerance value, is satisfied.

Despite DE's performance, as with other evolutionary algorithms, the control parameters of DE have to be determined. DE has three main control parameters: Population Size(NP), Scaling Factor(F) and Crossover Rate (CR). Tuning of these parameters is a time taking process. Since each circuit topology and its performance space is unique, it requires problem specific tuning of parameters for faster convergence [11]. Automatic parameter tuning or parameter adaptation is essential for the circuit synthesis problem to minimize overall design time. It has been included in a recent variant of DE, i.e., the Success-History based parameter adaptation for Differential Evolution (SHADE) [12]. Application of such variant to the circuit synthesis and its comparison with the basic version of DE [13] provides the circuit designer with a choice and fewer iterations for a successful circuit synthesis setup.

A. Adaptation of F and CR

In SHADE, the adaptation of parameters is achieved by assigning individual CR_i and F_i for each member d_i and changing them according to the following equations:

$$CR_i = \text{randn}_i(M_{CR,r_i}, 0.1) \quad (8)$$

$$F_i = \text{randc}_i(M_F,r_i, 0.1) \quad (9)$$

where $\text{randn}_i(\mu, \sigma^2)$ and $\text{randc}_i(\mu, \sigma^2)$ are normal and Cauchy distributions, r_i is a randomly selected index ($1 \leq r_i \leq H$). Here M_{CR}, M_F are historical mean values of size H of the vectors S_{CR} and S_F . S_{CR} and S_F contains record entries

of the CR and F parameters that are pooled during each generation based on the success of mutant vector in the selection step (Eqn. (7)) over the parent individual. These historical entries are initialized to 0.5, and are evaluated using the equations:

$$M_{CR,k}^{(G+1)} = \begin{cases} \text{mean}_{WA}(S_{CR}) & \text{if } S_{CR} \neq \emptyset \\ M_{CR,k}^{(G)} & \text{otherwise} \end{cases} \quad (10)$$

$$M_{F,k}^{(G+1)} = \begin{cases} \text{mean}_{WL}(S_F) & \text{if } S_F \neq \emptyset \\ M_{F,k}^{(G)} & \text{otherwise} \end{cases} \quad (11)$$

where k ($1 \leq k \leq H$) is the position in the memory to be updated, $\text{mean}_{WA}(S_{CR})$ and $\text{mean}_{WL}(S_F)$ are weighted means computed as

$$\text{mean}_{WA}(S_{CR}) = \sum_{k=1}^{|S_{CR}|} w_k \cdot S_{CR,k} \quad (12)$$

$$\text{mean}_{WL}(S_F) = \frac{\sum_{k=1}^{|S_F|} w_k \cdot S_{F,k}^2}{\sum_{k=1}^{|S_F|} w_k \cdot S_{F,k}} \quad (13)$$

$$w_k = \frac{\Delta e_k}{\sum_{k=1}^{|S_{CR}|} \Delta e_k} \quad (14)$$

where $\Delta e_k = |e(V_k^{(G)}) - e(d_k^{(G)})|$.

B. Comparison of DE and SHADE

For the comparison purpose, both the DE and SHADE algorithms are applied for the synthesis of the circuit topology shown in Fig. 3. The specifications and constraints for the design are listed in the Table 1. Fig. 4 and Fig. 5 shows the convergence plots of ten successive runs of the DE and SHADE algorithms respectively.

Convergence patterns in Fig. 4 reveals that for one out of ten runs, DE fails to converge and also the convergence time is irregular or inconsistent. The SHADE algorithm being global in nature has managed to converge in all ten successive runs and more consistently converged compared to DE. Thus, it can be concluded that for circuit synthesis, SHADE is a better alternative to standard DE algorithm. In the upcoming section, we have carried out the hierarchical synthesis of a sigma-delta modulator using the SHADE algorithm.

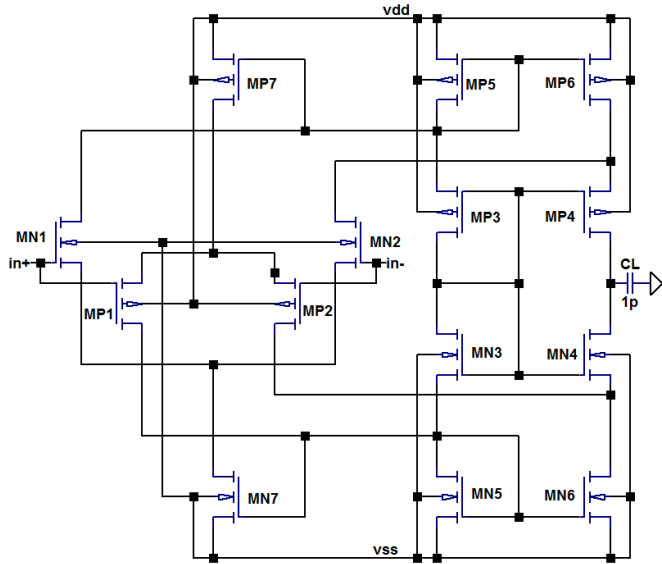


Figure 3: Schematic of self biased folded cascode amplifier

Table 1: Specifications, Constraints for the circuit in Fig. 3

Spec. / Constraint	Value
Power	5mW
Area	5000 μm^2
DC Gain	10000V/V (80dB)
UGF	100MHz
Phase margin	60 deg.
Offset	1e-6V
Output Swing	1V peak-peak
Slew Rate	100V/ $\mu\text{Sec.}$
L_{max}, L_{min}	[1 $\mu\text{m}, .36\mu\text{m}$]
W_{max}, W_{min}	[1000 $\mu\text{m}, 1\mu\text{m}$]
V_{dd}, V_{ss}	[1.65V, -1.65V]
Load Cap.	1pF
No. Des. Variables	16
Operating point constraints	32
Max. Generations	500

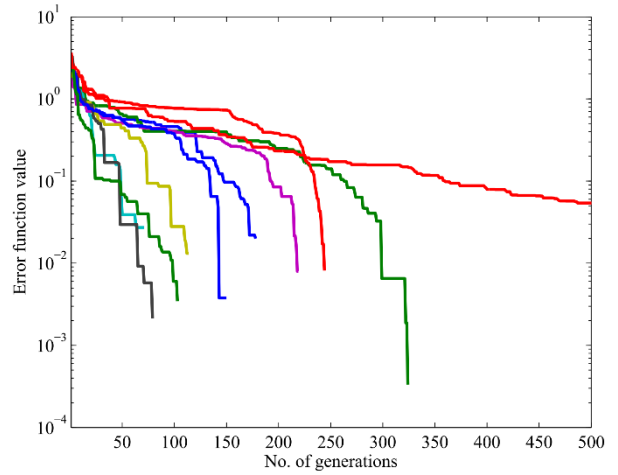


Figure 4: Convergence plots of the error function for 10 runs of Differential Evolution algorithm

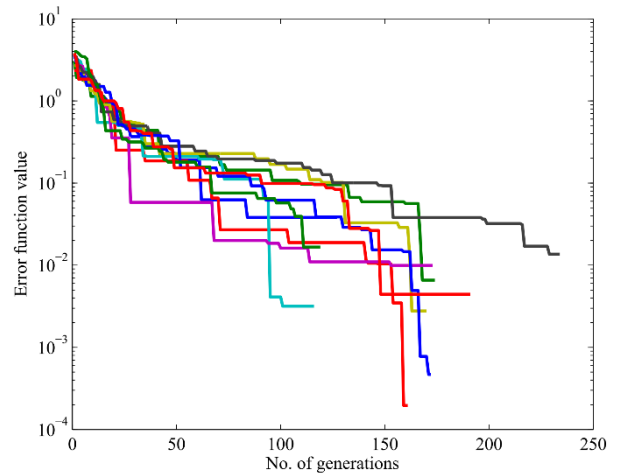


Figure 5: Convergence plots of the error function for 10 runs of SHADE algorithm

SIGMA-DELTA MODULATOR, SYSTEM LEVEL MODELING AND OPTIMIZATION

A. Basics of sigma delta modulator

Delta-sigma modulators have been adopted in many applications like audio, sensing, instrumentation and communication. The oversampling and noise-shaping properties of Delta-Sigma ADC helps in achieving superior resolutions up to 24-bit. The advantage of DSM compared to conventional Nyquist rate (e.g., Flash ADC) is that they are low cost, compatible with contemporary CMOS processes with low supply voltages and insensitive to process variations.

The block diagram of a second order sigma delta modulator is shown in Fig. 6.

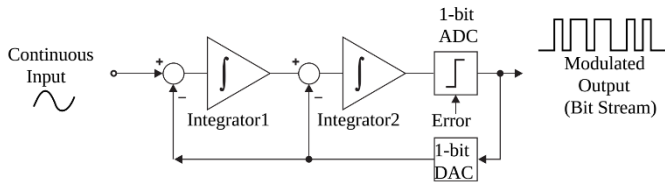


Figure 6: Block diagram of second order Sigma-Delta modulator

The Delta-Sigma modulator consists of three basic functional blocks: a) the loop filter, b) the coarse ADC, and c) the DAC in the feedback loop. The coarse ADC is usually implemented using a 1-bit quantizer, and hence the output is bit stream.

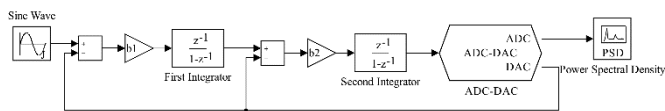


Figure 7: Block diagram of second order SDM implemented in SIMULINK

In a typical Nyquist rate A/D converter of N bit resolution, the number of quantization levels are 2^N . The fraction of voltage range q between two successive quantization levels or of the least-significant bit is given by

$$q = \frac{1}{2^N - 1} \quad (15)$$

During A/D conversion, a quantization error of $\pm q/2$ is expected with uniform noise power $q/\sqrt{12}$. f_b spread over the signal bandwidth f_b . Then the signal-to-noise (SNR) of an ideal N -bit Nyquist converter is given by

$$SNR_{ideal} = (6.02N + 1.76) \text{ dB} \quad (16)$$

In all practical situations, quantization noise is larger than theoretical value and hence the noise limited resolution, i.e. Effective Number of Bits (ENOB) is given as

$$ENOB = \frac{SNR - 1.76\text{dB}}{6.02\text{dB}} \quad (17)$$

When the sampling rate is much higher than Nyquist rate $f_{ov} = R \cdot 2f_b$, the A/D converters are called oversampling converters and the factor $R(100\text{ to }500)$ is called oversampling ratio (OSR). With oversampling converters we can achieve lower quantization error values and the ideal SNR is calculated as

$$SNR_{ideal} = (6.02N + 1.76 + 10\log R) \text{ dB} \quad (18)$$

The z -domain transfer function of an M^{th} order SDM is given by

$$Y(z) = X(z)z^{-1} + E(z)(1 - z^{-1})^M \\ = STF(z)X(z) + NTF(z)E(z) \quad (19)$$

The noise shaping phenomenon of the sigma-delta modulator pushes the quantization noise from signal band to out of the signal band and consequently, the improved SNR is given by [14]

$$SNR(\text{dB}) \approx 20\log A + 6.02N + 1.76 \\ + 10\log(2M + 1) - 9.94M + 3.01(2M + 1)r \quad (20)$$

Here, A is the magnitude of the input signal, and r is given by $R = 2^r$, $M = 2$ for the second order system shown in the Fig. 6. The sigma-delta modulator can be implemented either by using switched capacitor (SC) (Discrete-time) or an active RC (continuous-time) filters. SC based sigma delta ADCs are more compatible with the standard CMOS process and insensitive to mismatch and clock jitter. For this reason, SC-based sigma-delta ADCs are popular and considered in this work.

B. Non-Idealities of SC Integrators and their effects on SNR

Although SC-based Delta-sigma architectures are robust to the non-idealities encountered in electrical implementations of the integrator blocks, it is necessary to consider their impact when SNR is high in demand. The main sources of integrator non-idealities are finite DC gain, Slew rate, Bandwidth and Output swing of the opamp. It is well known that due to finiteness of the practical op-amp implementations, "Leaky integration" occurs and it has a big impact on Noise transfer function and thus on the SNR [15]. These non-idealities show little effect on the SNR (when the opamp in the integrator is practically close to the ideal opamp) at the cost of more resources in terms of area and power. Finding optimal opamp performances that ensure required SNR despite the non-idealities with least possible resources is of utmost importance. For this purpose, we have used the optimization approach to synthesize the SDM using system level simulations. The macro-model of the integrator with opamps having finite gain, bandwidth, slew rate and swing have been included in the simulations. The objective of the optimization procedure is to achieve 100dB SNR with optimum opamp performances. The Fig. 8 shows the block diagram of the realistic model of the integrator that has been used in the second-order SDM implementation shown in Fig. 7.

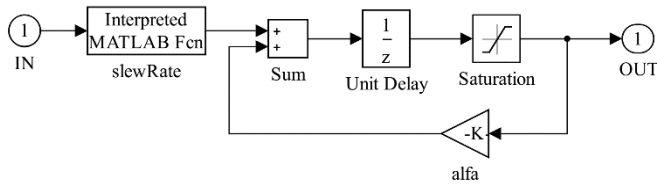


Figure 8: Block diagram of realistic integrator implemented in SIMULINK

• **DC gain:** An ideal integrator with unity gain has a transfer function given by

$$H(z) = \frac{z^{-1}}{1 - z^{-1}} \quad (21)$$

but due to finite DC gain, only a fraction α of the previous output is added to the next sample. The new transfer function with this “leakage” is given by

$$H(z) = \frac{z^{-1}}{1 - \alpha z^{-1}} \quad (22)$$

where the DC gain H_0 has the value

$$H_0 = H(0) = \frac{1}{1 - \alpha} \quad (23)$$

• **Bandwidth and Slew Rate:** Finite bandwidth and slew rate are correlated and can effect the transient response of the integrator producing inaccurate charge transfers within each clock cycles. The output during n^{th} integration period is given by

$$v_0(t) = v_0(nT - T) + \alpha V_s (1 - e^{-\frac{t}{\tau}}), nT - T/2 < t < nT \quad (24)$$

where $V_s = V_{in}(nT - T/2)$, α is the integrator leakage, $\tau = 1/(2\pi \cdot UGF)$ is the time constant and UGF is the unity gain frequency of the opamp used in the integrator. The maximum slope of the above equation when $t = 0$ is given by

$$\frac{d}{dt} v_0(t)|_{max} = \alpha \frac{V_s}{\tau} \quad (25)$$

If the opamp slew rate (SR) is greater than the specified value in the above equation, then the SNR of the SDM is not limited by SR. Else,

$$v_0(t) = v_0(nT - T) + SR \cdot t \quad \text{for } t \leq t_0 \quad (26)$$

and

$$v_0(t) = v_0(t_0) + (\alpha V_s - SR \cdot t_0) (1 - e^{-\frac{t-t_0}{\tau}}) \quad \text{for } t > t_0 \quad (27)$$

where t_0 is given by

$$t_0 = \frac{\alpha V_s}{SR} - \tau \quad (28)$$

• **Saturation (Output Swing):** The obvious effect of clipping can occur when the output saturation levels of the opamp are below the specified range of the SDM itself. If the clipping occurs, the output of the SDM fails to follow the ideal waveform thus leading to distortions. Therefore, the output swing of the opamp has to be taken into consideration during the design.

All the above-mentioned non-idealities of the opamp have been modeled and taken into account during the simulation.

C. System level optimization

We have three degrees of freedom, i.e., modulator order, Oversampling ratio, and Quantizer resolution to optimize the performance at the architecture level synthesis of the SDM. One can synthesize the required noise transfer function (NTF) Using the SDMtoolbox [16] and the resulted block diagram can be mapped to a behavior level circuit implementation. The behavioral level simulation quickly validates the system when subjected to realistic circuit component behaviors such as noise, clock jitter, OPAMP non-idealities, etc.

Table 2: Specifications of the Sigma-delta modulator shown in Fig. 7

Specification	Value
Input Signal Amplitude	1 V
Signal Bandwidth	22.05 KHz
Sampling Frequency	11.29 MHz
Over Sampling Ratio	256
Target SNR	100 dB
Number of Samples	66536
Order of SDM	2
Quantization levels	2(1-bit)
Form	CIFB

In this work, we have considered a 2nd order SDM architecture as shown in Fig. 7. The specifications of the system are shown in Table 2.

We have modeled the system level synthesis of SDM as an optimization problem including DC Gain (A_{v0}), Unity Gain Frequency UGF , Slew Rate (SR) and Output Swing V_{max} as

design variables and SNR as the only objective:

$$\max_{A_{v0}, UGF, SR, V_{max}} SNR(A_{v0}, UGF, SR, V_{max}) \quad (29)$$

subject to

$$\min \sum (\widehat{A}_{v0}, \widehat{UGF}, \widehat{SR}, \widehat{V}_{max}) \quad (30)$$

where the parameters with $\widehat{\Lambda}$ indicate normalized values. The optimization problem is solved using the SHADE algorithm.

RESULTS AND DISCUSSIONS

We have used the same flow as shown in Fig. 2 for the system level optimization. In this design, the objective is to maximize SNR. The dimension of the system level optimization problem is 10. The population size is considered as 20 for the SHADE algorithm. The optimization algorithm is coded in MATLAB programming environment while the system level simulation is carried out in SIMULINK environment. All the experiments in this paper are carried out on a system with an Intel i-5 processor, 4 Giga-Bytes of RAM running the Linux operating system. The evolution of SNR with generations during optimization process is shown in the Fig. 9. The mean and best SNR of the populations with generations are depicted in this figure. It is observed that the standard deviation of the SNR is minimum after 20 generations. Thus, it can be concluded that the algorithm is robust for maximizing the SNR of the considered ADC. The overall optimization time is approximately 2 hours 25 minutes for 32 generations as shown in the figure. The power spectral density plot of the final solution from the optimization is shown in Fig. 10. It shows that the noise transfer function of the SDM is not affected by the realistic amplifier models with the resultant values of the specifications obtained from the synthesis. The Fig. 11 shows the probability densities of the outputs of both the integrators. The dynamic range of the first and second integrators are 2 volts and 0.2 volts peak to peak respectively. It suggests that the output dynamic range requirement for the amplifiers.

The optimized values of A_{v0}, UGF, SR, V_{max} are listed in Table 3. These parameters of the opamps are the specifications for the circuit level optimization.

Table 3: Constraints and optimized parameters of the second order SDM

Parameter	Max	Min	1st Integ.	2nd Integ.
DC gain (V/V)	1e3	1e2	1000	1000
UGF (MHz)	100	1	70	2
Slew rate (V/ μ Sec)	100	1	60	40
Voltage Swing (V)	1	0.1	.91	.3
Scale Coeff. (b_1, b_2)	1	0	.26	.15

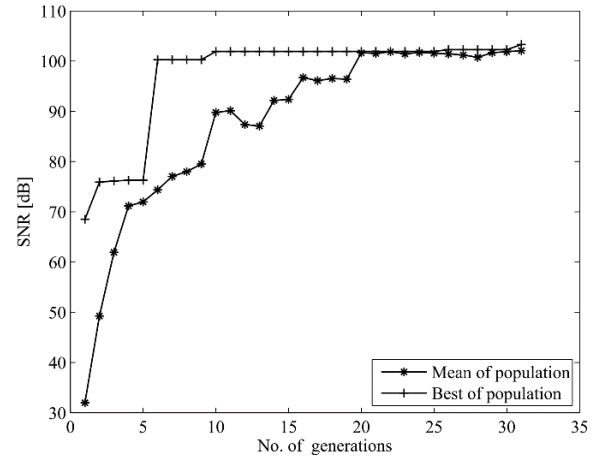


Figure 9: Evolution of the SNR with respect to generations

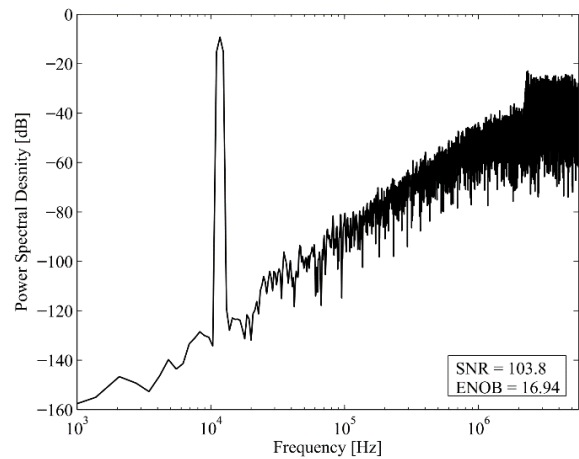


Figure 10: PSD of final solution generated from system level optimization

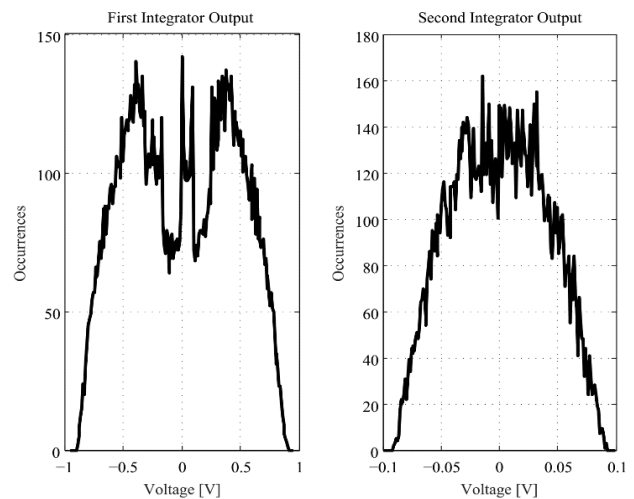


Figure 11: Distribution of the output signal integrators of the SDM

We have considered 180nm CMOS process technology devices for the SDM design with a supply voltage of 3.3 volts. For the first integrator, since the range output requirement is high (2V), we consider using a Miller compensated two-stage amplifier topology (TSAMP) as shown in Fig. 12. For the second integrator, since the performance requirements are quite gentle, we use a high-performance self-biased folded-cascode amplifier topology (FCAMP) [17] as shown in Fig. 3. We used the SHADE algorithm to synthesize both the circuit topologies to achieve the specification tabulated in Table 3 .

The convergence plots of the SHADE algorithm for both the opamps (Fig. 12 and Fig. 3) are shown in Fig. 13. Despite tighter constraints for both the circuits, the SHADE algorithm has managed to converge in less than 70 generations. The performances obtained from the synthesis are listed in Table 4 and they are above the specifications mentioned in Table 3. The design variables of both the circuits and their synthesized values are listed in Table 5. We have observed that there are no unusual or extreme ratios of transistor width and lengths when compared to typical manual designs. The DC, AC, and transient performances are plotted in Fig. 14, Fig. 15 and Fig. 16 respectively. These are obtained from the responses of the synthesized designs. The synthesis procedures are often criticized for having the circuits designed in technically meaningless regions. To avoid the problem, we have used the proper sizing rules or operating point constraints [18]. These plots are presented to show the trustworthiness of the automated synthesis approach. In general, the designs are finalized when the performances are subjected to process, supply voltage and temperature (PVT) variations. We limit our study to nominal values of PVT and synthesis based on PVT considerations is beyond the scope of this work.

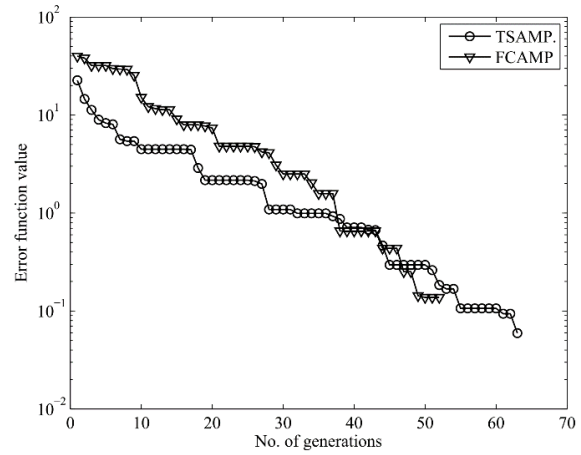


Figure 13: Error Function with respect to No. of generations

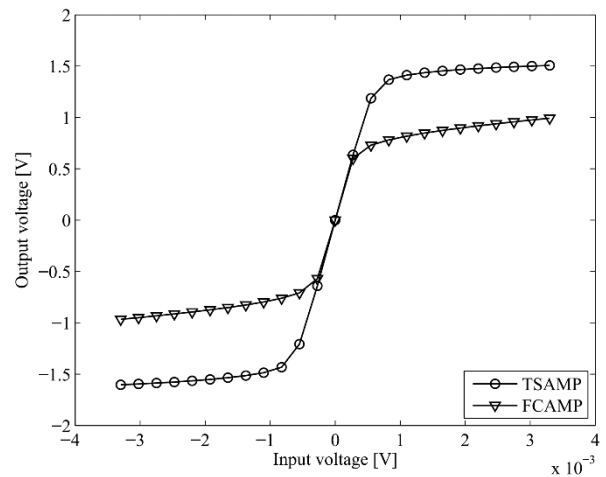


Figure 14: DC Transfer characteristics of the amplifiers

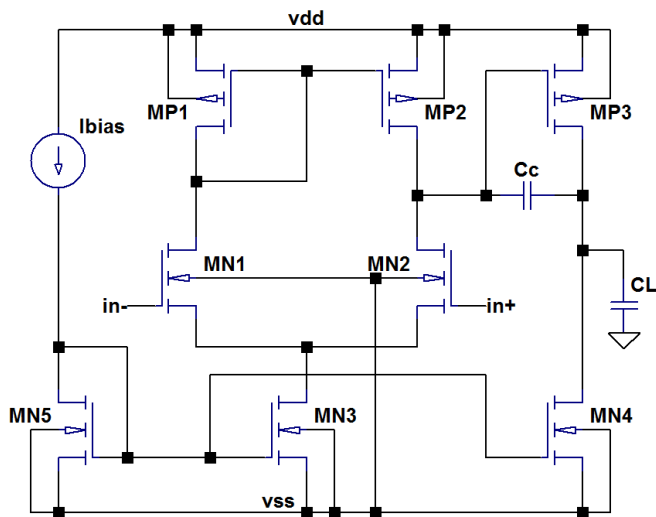


Figure 12: Schematic of Two Stage Miller compensated amplifier

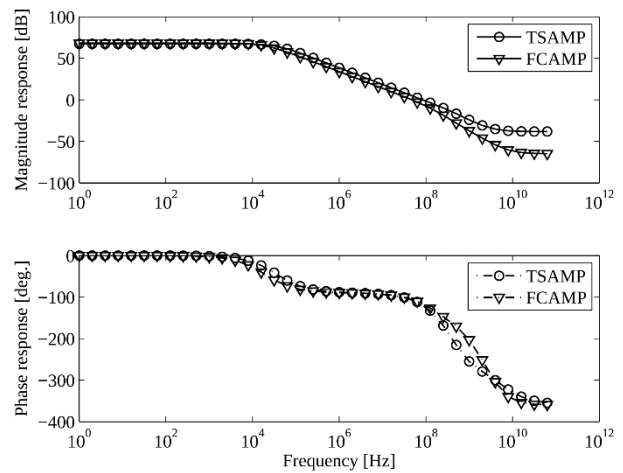


Figure 15: Magnitude and Phase responses of the amplifiers

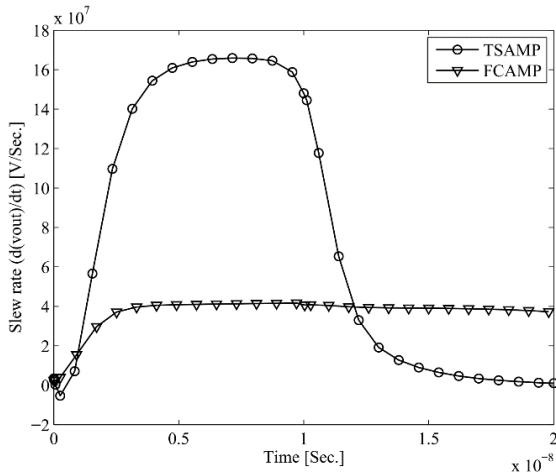


Figure 16: Transient response of the amplifiers

Table 4: Performances obtained through circuit level synthesis of schematics in Fig. 12 and Fig. 3

Performance	TSAMP	FCAMP
UGF(MHz)	83.8	47.3
PM(deg.)	60.9	74.4
DC Gain(V/V)	2338.4	2570.2
Slew Rate(V/μSec.)	165.8	41.6
Neg. Swing(V)	-1.30	-0.51
Pos. Swing(V)	1.25	0.54
Offset(mV)	-0.17	0.21
Area(μm) ²	.97	.93
Power(mW)	4.2	.43
Synthesis time(Sec.)	630	495

Table 5: Design variables and their values of the opamps used in the first and second integrators

TSAMP	Value	FCAMP	Value
$MN_{1,2}(W/L)$	8.00/0.79	$MN_{1,2}(W/L)$	11.19/0.53
$MN_3(W/L)$	19.88/0.88	$MN_{3,4}(W/L)$	1.78/0.91
$MN_4(W/L)$	33.37/0.50	$MN_{5,6}(W/L)$	4.80/0.42
$MN_5(W/L)$	17.08/0.44	$MN_7(W/L)$	2.80/0.92
$MP_{1,2}(W/L)$	15.90/0.45	$MP_{1,2}(W/L)$	4.70/0.51
$MP_3(W/L)$	72.63/0.40	$MP_{3,4}(W/L)$	5.70/0.84
Cc	2.3pF	$MP_{5,6}(W/L)$	39.66/0.40
Ibias	340μ A	$MP_7(W/L)$	29.57/0.85

CONCLUSIONS

In this paper, we have presented a hierarchical synthesis method for the second order sigma delta modulator. Initially, a model for the SDM with non-idealities of the opamps and their effect on the noise transfer function is discussed. The performance of SHADE algorithm is compared with the standard DE algorithm for circuit synthesis. From the comparison, it is concluded that the SHADE algorithm performs better regarding the rate of convergence and rate of success. The main advantages of this algorithm are that it has only one control parameter. Further, using the SHADE algorithm, we have optimized the parameters of the opamp with the objective to maximize the SNR and minimize the opamp specifications. Later the specifications obtained through the system level optimization for the opamps are applied to circuit level synthesis. Here we have considered two different architectures for the two integrators in the SDM. First, a miller compensated two-stage opamp for high output swing, and second a self-biased folded cascode structure for low power consumption. From the circuit level synthesis process, we have obtained component values of the opamps used in the SDM.

ACKNOWLEDGMENT

The authors would like to thank the University Grants Commission (UGC), Government of India for providing necessary support to carry out this research work.

REFERENCES

- [1] L. Qian, Z. Bi, D. Zhou and X. Zeng, "Automated Technology Migration Methodology for Mixed-Signal Circuit Based on Multistart Optimization Framework," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 11, pp. 2595-2605, 2015.
- [2] H. Y. Koh, C. H. Sequin and P. R. Gray, "OPASYN: a compiler for CMOS operational amplifiers," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, no. 2, pp. 113-125, 1990.
- [3] G. Van der Plas, G. Debyser, F. Leyn, K. Lampaert, J. Vandebussche, G. Gielen, W. Sansen, P. Veselinovic and D. Leenarts, "AMGIE-A synthesis environment for CMOS analog integrated circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 9, pp. 1037-1058, 2001.
- [4] W. Nye, D. C. Riley, A. Sangiovanni-Vincentelli and A. Tits, "DELIGHT.SPICE: an optimization-based

- system for the design of integrated circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 7, no. 4, pp. 501-519, 1988.
- [5] R. Phelps, M. Krasnicki, R. A. Rutenbar, L. R. Carley and J. Hellums, "ANACONDA: robust synthesis of analog circuits via stochastic pattern search," in *Proceedings of the IEEE 1999 Custom Integrated Circuits Conference*, San Diego, CA, 1999.
- [6] E. Martens and G. Gielen, "Classification of analog synthesis tools based on their architecture selection mechanisms," *Integration, the VLSI Journal*, vol. 41, no. 2, pp. 238-252, February 2008.
- [7] Kruiskamp and D. L. Wim, "DARWIN: CMOS opamp Synthesis by Means of a Genetic Algorithm," in *32nd Design Automation Conference*, San Francisco, CA, 1995.
- [8] S. L. Sabat, S. K. Kumar and S. K. Udgata, "Differential evolution and swarm intelligence techniques for analog circuit synthesis," in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, Coimbatore, 2009.
- [9] P. N. Suganthan and S. Das, "Differential Evolution: A Survey of the State-of-the-Art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4-31, 2011.
- [10] S. Kudikala, S. L. Sabat and S. K. Udgata, "Performance study of harmony search algorithm for analog circuit sizing," in *2011 International Symposium on Electronic System Design*, Kochi, Kerala, 2011.
- [11] A. K. Qin, V. L. Huang and P. N. Suganthan, "Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398-417, April 2009.
- [12] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for Differential Evolution," in *IEEE Congress on Evolutionary Computation*, Cancun, 2013.
- [13] R. Storn and K. Price, "Differential Evolution - A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 4, no. 11, pp. 341-359, 1997.
- [14] J. D. Reiss, "Understanding sigma-delta modulation: The solved and," *Journal of the Audio Engineering Society*, vol. 56, no. 1/2, p. 49-64, 2008.
- [15] B. E. Boser and B. A. Wooley, "The design of sigma-delta modulation analog-to-digital converters," *IEEE Journal of Solid-State Circuits*, vol. 23, no. 6, p. 1298-1308, 1988.
- [16] S. Brigati, F. Francesconi, P. Malcovati, D. Tonietto and A. Baschirotto, "Modeling sigma-delta modulator non-idealities in SIMULINK(R)," in *Proceedings of the 1999 IEEE International Symposium on Circuits and Systems, 1999. ISCAS '99.*, Orlando, FL, 1999.
- [17] P. Mandal and V. Visvanathan, "A self-biased high performance folded cascode CMOS op-amp," in *Proceedings Tenth International Conference on VLSI Design*, Hyderabad, 1997.
- [18] T. Massier, . H. Graeb and . U. Schlichtmann, "The Sizing Rules Method for CMOS and Bipolar Analog Integrated Circuit Synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 12, pp. 2209-2222, December 2008.