Automatic Synthesis of CMOS Operational Amplifiers: A Fuzzy Optimization Approach

Biranchinath Sahu[1] and Aloke K. Dutta[2]
[1]School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, USA
[2]Department of Electrical Engineering, Indian Institute of Technology, Kanpur, India

Abstract

In this paper, we present a method for optimizing and automating the components and transistor sizing for CMOS operational amplifiers (op-amps). The optimization approaches used for the synthesis of analog circuits are found to be very much rigid in terms of capturing human intentions. In this work, we have observed that with the use of fuzzy membership functions, human intentions for expressing wide variety of requirements, e.g., minimize power, maximize gain, etc., which are often conflicting in nature, can be captured effectively in order to formulate the objective function. For each of the performance specifications of a given topology, a membership function is assigned to measure the degree of fulfillment of the objectives and the constraints. A number of objectives are optimized simultaneously by assigning weights to each of them representing their relative importance, and then by clustering together to form the objective function that is solved by an optimization algorithm. We have considered the channel length modulation parameter (λ) for the computation of DC bias point and small signal parameters. The design results obtained from our optimization algorithm showed an excellent match with those obtained from SPICE simulation for a number of op-amp topologies.

1. Introduction

The growing requirements for the single chip mixed-signal designs of very large scale integration (VLSI) together with the continuous trend towards smaller feature sizes and an even higher scale of integration have brought about new dimensions in the analog circuit design complexity. Intuitively, the best way to match the fast technological trend towards single chip analog/digital VLSI systems and meet the time-to-market requirements is to automate the design process. However, in contrast to the fully automated digital designs, analog designs are still handcrafted manually by experienced designers. The development of tools for designing analog ICs is still in a quite primitive stage and lacking the industrial penetration and acceptance already achieved by their digital counterparts.

Analog circuit design is known to be a knowledge-intensive, multiphase, and iterative task that usually stretches over a significant period of time and is performed by designers with a large portfolio of skills. Therefore, it is considered as a form of art. This idea is further triggered by lack of analog circuit design formalisms, i.e., neither there exists a circuit-independent design procedure nor does exist a formal representation, i.e., the equivalent of Boolean algebra in the digital domain, to relate the circuit function to its structure in a consistent way. The main obstacle to such developments is the very nature of analog signals that the circuit deals with, the continuous range of signal amplitudes, and their continuous time dependency. Analog circuit design process can be classified into three subtasks: a) Topology selection, b) Parametric optimization, and c) Layout generation. The designer selects an appropriate topology among various possible alternative architectures, in order to achieve higher performance for a desired application. The second step, after the topology of the circuit and component types are fixed, consists of assigning values to the circuit parameters (e.g., widths and lengths of metal-oxide-semiconductor (MOS) transistors, resistor and capacitor values, bias voltages and currents, etc.), while satisfying the desired performance criteria. Finally, the optimized circuit needs to be transformed into a layout.

In the techniques used for parameter optimization, it is very crucial to formulate the problem in the right manner, and use an optimization algorithm that reflects the user's intentions as accurately as possible. Most of the optimization methods available are very rigid and often difficult to adapt to the design problems without a corresponding loss of accuracy. Consequently, these techniques have the limitation of using rigid optimization problem that is too restrictive, thus eliminating or reducing the possibilities for trade-offs, which are important factors in the overall design process. As a result, the design space becomes limited, and, in most cases, no optimal solution can be found. Another problem is the choice of the starting point, on which the quality of the optimized solution and design time heavily depend on. In most of the proposed techniques, this task is left to the user/designer.

The focal point of this work is aimed towards development of a tool in order to find a set of circuit parameters (or design variables), such that the design objectives are optimized while satisfying performance constraints for different op-amp topologies, given the fact that op-amp is used extensively in mixed-signal systems. Also, an attempt is made to formulate the optimization problem in a realistic manner by capturing the human intentions in the inherently imprecise terms used by the user, e.g., *minimize* power, *small* output resistance, etc. Earlier, the fuzzy set theory [1] has been applied in FPAD [2] to formulate the objective function for optimization from the design objectives and constraints. Our approach is based on the concepts as those described in [2], with a number of differences and generalization in the formulation. Depending on the objectivity associated with a particular performance or constraint, it is transformed into a fuzzy objective, a fuzzy constraint or a strict constraint. We don't consider strict constraints outside the objective function rather define it as a special case of the more general fuzzy constraint. The trade-offs among a number of specifications, depending on the users' interest, are handled by using membership functions. Each of the performance specifications is assigned a weight that reflects its relative importance.

The performance parameters of the circuit, as needed by the optimization routine, are evaluated by using analytical circuit equations, which describe the performance objectives in terms of the design variables. The technology parameters needed for computation are directly read from a technology file, making the synthesis routine technology independent. The time-consuming circuit simulator is avoided inside the optimization loop. The objective function is formulated by clubbing all the membership functions associated with the specifications in proportion to their assigned weights. Finally, the solution to the formulated problem is carried out using Powell's direct search algorithm. Another important issue we have addressed in this work is the consideration of channel length modulation parameter for DC operating point

This research was done at Department of Electrical Engineering, Indian Institute of Technology, Kanpur, India.

computation, which has been neglected by most of the previous approaches.

The paper is organized as follows. A review of the previous approaches towards analog design automation is presented in Section 2. Section 3 describes the overview of the methodology adopted in this work. The formulation of the objective function along with the concept of the fuzzy constraints and the membership functions is presented in Section 4. Section 5 gives design results of three types of commonly used CMOS op-amps for a set of performance specifications and constraints. Finally, the limitations, summary and conclusions have been presented.

2. Previous work

With the surge of research interest in analog design automation (DA) in the last decade, a plethora of prototype systems, many of which are capable of handling full-custom designs, have been reported. A number of start-up companies, e.g., Neolinear Inc., Barcelona Design Automation Inc., etc., are reported to have tools for automatic design of analog circuits [3]. A survey of the progress in the area of the analog DA can be found in the literature [4-6]. In general, the previous approaches can be classified in three categories: a) Layout based approach, b) Knowledge based approach, and c) Optimization based approach.

The layout-based approach is an adaptation of extensively used standard cell, gate array, and parameterized cell methods found in the digital domain and designs are controlled to a large extent by layout. Analog arrays are pre-designed and laid-out blocks of different sizes, configurations, and levels of complexity, varying from single component arrays to circuit arrays. However, neither do they provide the necessary design flexibility required for high performance analog circuits nor they are very much cost-effective. Knowledge-based systems exploit domain knowledge in order to design analog ICs, and they address the design task in a full custom way, thereby allowing for maximum flexibility and a potentially better coverage of the circuits' performance space. These can be classified in three categories: a) hierarchical, b) fixed topology, and c) combined hierarchical and fixed topology approach. The hierarchical design approach involves partitioning the circuit (or system) into smaller distinct parts, each of these parts is assigned a set of specifications which, if met, then the combination of these parts will yield the desired circuit performance. The process is repeated in a similar manner for smaller blocks at different hierarchical levels. These systems maintain the greatest degrees of freedom, and, thus, a small architecture library can lead to a large number of different topologies with wide performance spectra. The various systems that use this kind of approach are OASYS [7], BLADES [8], etc. The fixed topology approach employs a sizing method in order to compute appropriate sizes for the devices within a given fixed circuit topology. These fixed, un-sized, device level circuit topologies are stored in a knowledge base together with the necessary domain knowledge for dimensioning the devices. Some of the systems reported in the literature that follow this approach are IDAC [9], OPASYN [10], OAC [11], etc. There are some knowledge-based design methods, which combine features of both the hierarchical and the fixed topology approaches, e.g., ASAIC [12], ISSAC [13], ISAID [14], etc.

The optimization-based design approach uses recent advances in the optimization theory and algorithms, and relates these to the parametric optimization of analog ICs. The synthesis problem is formulated as one of mathematical programming. The circuit performances are considered to be the objective functions, which are to be minimized or maximized subject to a set of specification constraints. Optimization based design approaches can be broadly classified into two categories: a) Simulation-based optimization, b) Analytical equation based optimization. Historically, the very first attempt towards analog DA were numerical optimization based. Systems such as DELIGHT.SPICE [15], ECSTACY [16], ADOPT [17], and ASTRX/OBLX [18] consider the sizing of the individual transistors in a given circuit topology as an optimization problem. Typically, these systems employ optimization algorithms, which iteratively adjust the individual transistor sizes in order to meet the constraints and objectives specified by the user. A simulator is used within the optimization loop to assess the performance of the circuit during each iteration. These design approaches are referred to as simulation based optimization. The time consuming and expensive simulator inside the optimization loop is avoided by using simplified but sufficiently accurate analytical models that predict circuit performances, and this approach is referred to as analytical equation based optimization. A number of prototypes have come out in recent times, which use this technique, e.g., OPASYN [10], STAIC [19], FPAD [2], FASY [20], and those reported by Mandal and Viswanathan[21], Maulik et al. [22], and Hensherson et al. [23].

The salient features of our proposed implementation with respect to other approaches are as follows:

1. A generalized formulation of the objective function (specifications and constraints) is presented , which supports the imprecision and vagueness inherently associated with the real-world formulation of any design problem. The formulation takes into account performance tolerances and allows varying degrees of trade-off measures for a particular solution.
2. An automatic initial sizing procedure deduces the initial values for the various design variables for the optimization routine, based on the input specifications, design knowledge, and heuristics.
3. Analytic circuit equations and device models are used to avoid the time-consuming circuit simulator inside the optimization loop. To allow trade-offs and limit repeated modifications of the input specifications, they are not assigned precise target values; instead, each specification is formulated as a fuzzy set, i.e., a range of possible values are assigned to each of them with varying degrees of acceptability.
4. Finally, we have used a more accurate analysis, i.e., *the channel length modulation parameter* (λ) is taken into consideration. The computation of the DC operating point is performed iteratively.

3. The fuzzy optimization approach: An overview

The flow chart of the synthesis procedure adopted in this work is shown in Fig. 3.1. Once the circuit topology (the type of op-amp) is selected, the program starts with a set of user-defined input performance specifications, and deduces the optimal set of design parameter values (e.g., capacitors and biasing resistors, lengths and widths of MOS transistors, bias currents, etc.). Each of the performance specifications is either an *objective* or a *constraint*, specified by the designer. A *constraint* can either be a *fuzzy constraint* or a *strict constraint*.

Each of the specifications and constraints is assigned a weight, which represents its relative importance in the set of objectives. The details of the objective function formulation is described in the next section. A set of constant parameters, e.g., supply voltage, load capacitance, maximum length and width of transistors, etc., along with the performance specifications is supplied by the user. The next step is to obtain an initial solution for the objective function using automatic device sizing procedure based on the circuit knowledge, basic assumptions, and first-order simplified analytical equations. The designer may choose to skip the initial sizing procedure and provide an initial guess as an input to the optimization module.

The fuzzy optimization module starts with the initial solution and identifies a set of parameter values that satisfy the design objectives and constraints within a reasonable tolerance. The program intends to find a solution, which fulfils as many requirements specified by the user as possible, and is aimed at arriving at a solution in the design space such that the designed values of the performances are close to the specified targets. After

optimization, a net list of the circuit is created by the program, which can be directly used for SPICE simulation, in order to verify the designed circuit. During the optimization process, all the technology dependent parameters, e.g., minimum channel length and threshold voltage of MOS transistors, current gain and early voltage of bipolar transistors, etc., are read from separate technology files that contain the parameters for a fabrication process. In order to optimize a circuit for another technology, only the technology file needs to be changed, thus rendering this approach technology independent. Once the optimized circuit is obtained, it can be fine-tuned using a simulation-based optimizer, e.g., DELIGHT.SPICE [15], in order to take into account the temperature and process variations. The tasks of final tune and layout generation have not been implemented in this work.

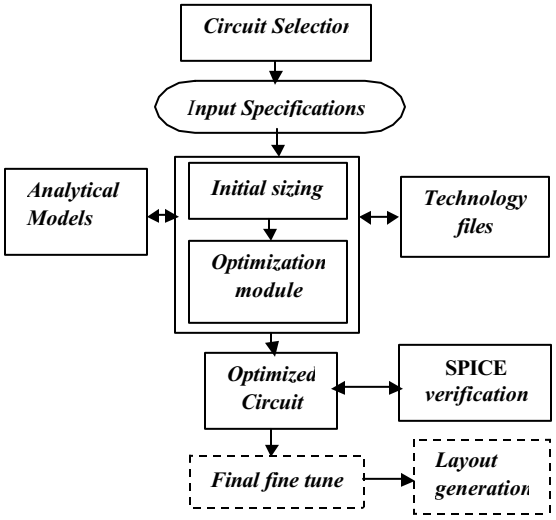

Fig.3.1: The general structure of the synthesis algorithm.

4. Theory of the objective function formulation

The design problem can be formulated as a mathematical programming one in the following manner:

Maximize or minimize $\{f_1(x), \ldots\ldots\ldots, f_m(x)\}$,
subject to: $g_i(x) \leq$ *or* $\geq$ *or* $= spec_i$,
for $i = 1, 2, \ldots, n,$ and $x_{min} \leq x \leq x_{max}$, (4.1)

where $f_j(x)$ $[j=1, 2, \ldots, m]$ are m objective functions to be minimized or maximized, $g_i(x) \leq$ or $\geq$ or $= spec_i$ are n constraints to be satisfied, $spec_i$ is the limiting value of the i^{th} specification, x is the vector of the design parameters (e.g., lengths and widths of the various transistors, values of resistors and capacitors, bias voltages and currents, etc.), and x_{min} and x_{max} are the minimum and the maximum values of the design parameters respectively.

During the formulation of the design problem as a mathematical programming function, the requirements of the designer are not always well defined. To model the design problem as given by Eqn.(4.1), the designer is often forced to define in strict mathematical terms rather than in real world terms, which are more diversified in nature. However, the most commonly used terms, e.g., *minimize*, *small*, *very large*, *substantially higher than*, etc., for expressing the objectives and constraints possess fuzzy meanings, and, therefore, are difficult to express by precise values required for the problem formulation. Therefore, with each objective function $f_i(x)$ in Eqn.(4.1), a fuzzy set is associated that represents the fuzzy meaning of each of the performance objectives and what the designer actually wants to achieve with that. The terms used by the designer, e.g., *high* gain, *small* output resistance, *maximize* slew rate, *minimize* area and power, *maximize* bandwidth, etc., can be formulated using fuzzy sets and treated as fuzzy objectives. While attempting to minimize a performance function, designers often stop the search procedure when it attains a minimum value, even though that minimum may be local instead of global. Additional searching may be extremely time consuming with no apparent improvement in the objective function. In order to overcome the above problem, with each objective function given in Eqn.(4.1), one fuzzy set is associated, which formulates the fuzzy meaning of *minimize (maximize),* and, thereby, the intention of the designer is expressed more precisely. In the same manner, the performance specifications stated in real world terms, e.g., *high* gain, *small* output resistance, etc., are formulated using fuzzy sets and are treated as fuzzy objectives.

For each fuzzy objective, a *membership function* is defined, which associates a grade of membership $\mu_{fi}(x)$ with each $f_i(x)$ of the objective function that reflects the degree of acceptability of the performance value. If D_{fi} is the interval of the possible values of $f_i(x)$, then μ_{fi} is defined as [2]

$$\mu_{fi} : D_{fi} \rightarrow [0, 1], \quad f_i(x) \rightarrow \mu_{fi}(x), \qquad (4.2)$$

where $\mu_{fi}(x)$ is a real number in the interval [0, 1], reflecting the degree of fulfillment of the fuzzy objective associated with the objective function $f_i(x)$. More clearly, it can be stated as: $\mu_{fi}(x) = 1$ means that $f_i(x)$ is fully satisfied; on the other hand, if $\mu_{fi}(x) = 0$, then $f_i(x)$ is not satisfied at all, which means that it takes a value that is totally unacceptable to the designer. An intermediate value of $\mu_{fi}(x)$ between zero and unity reflects the acceptability of that particular performance value. It is quite obvious that the closer the value of $\mu_{fi}(x)$ is to unity, the better is the solution. The formulation of the problem as per Eqn.(4.1) can thus be replaced as follows:

Maximize: $\{\mu_{f1}, \mu_{f2}, \ldots\ldots\ldots\ldots, \mu_{fm}\}$,
subject to: $g_i(x) \leq$ or $\geq$ or $= spec_i$
for $i = 1, 2, \ldots, n,$ and $x_{min} \leq x \leq x_{max}$. (4.3)

Obtaining the design vector x that fully satisfies Eqn.(4.3) is generally not a trivial task, since the design objectives are often conflicting in nature. In that case, one has to yield to some compromises by permitting some tolerances in the inequalities. Thus, the symbol "$\leq$" in Eqn.(4.3) is replaced by the fuzzy version of "$\lesssim$", which means *essentially smaller than* [1]. Similarly "$\geq$" and "=" are replaced by their fuzzy versions *essentially greater than* and *essentially equal to* respectively. This formulation allows some tolerance in the constraint and provides a measure of it. Thus, the constraint $g_i(x) \leq$ or $\geq$ or $= spec_i$ becomes a *fuzzy constraint.* Each fuzzy constraint is characterized by a membership function $\mu_{gi}(x)$, which reflects the degree of fulfillment of $g_i(x) \leq$ or $\geq$ or $= spec_i$. For a *fuzzy objective*, the membership function must reflect a constant effort to improve the corresponding performance function $f_i(x)$. On the other hand, in case of a *fuzzy constraint*, once the threshold is achieved, no further effort is made to improve the performance function. In case of *strict constraint* no tolerance is allowed.

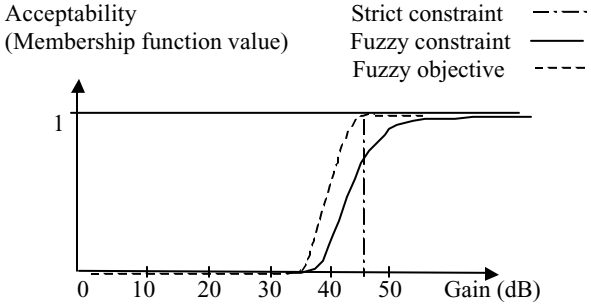

Fig. 4.1: A comparison of the membership functions for the *fuzzy objective*, the *fuzzy constraint* and the *strict constraint* for the specification *Gain > or = 45 dB.*

Figure 4.1 shows a comparison of the *fuzzy objective*, the *fuzzy constraint* and the *strict constraint*, when associated with the

performance specification *Gain > or = 45 dB.* It is quite obvious the figure that when the specification *Gain > or = 45 dB* is modeled as a *fuzzy constraint*, with the objective function being one of the various specifications, e.g., *minimize area, minimize power dissipation*, etc., the value of the membership function improves only up to the point where the specification is fully met. On the other hand, when it is modeled as a *fuzzy objective*, the membership function value, while continuously increasing itself towards unity, pulls the value of the gain beyond the specified target value. Thus, the membership function value reaches unity at a very high value of gain (may be 80 dB or so in this case), and, thereby, an effort is put to maximize the performance specification.

Considering the various types of objectives and constraints, the membership functions can be broadly categorized into three items, i.e., ***greater than equal (≥), less than equal (≤), and equal to (=)***. In this work, each category of the membership functions are implemented in five different ways, thereby different profiles of variations can be obtained. An example of the implementation ***less than equal (≤)*** is presented below [2].

$$\mu_{gi}(x) = \exp\left\{-\left(\frac{g_j(x) - spec_j}{p_i/2.0}\right)^2\right\}, \quad g_j \text{ f } spec_j \qquad (4.4)$$

$$= 1. \qquad g_j \le spec_j$$

where $spec_j$ and p_j are the values of specification and tolerance respectively. Depending on the tolerance value, the constraint can be either fuzzy (finite value of p_j) or strict (p_j equals zero).

After the objectives and the constraints are fuzzified, i.e., the corresponding membership functions are defined, Eqn.(4.3) becomes:

$$\text{Maximize: } \{\mu_{f1}, \mu_{f2}, \ldots\ldots \mu_{fm}, \mu_{g1}, \mu_{g2}, \ldots\ldots, \mu_{gn}\}$$
$$\text{subject to: } x_{min} \le x \le x_{max}. \qquad (4.5)$$

In order to solve the above problem, the membership functions μ_{fi} and μ_{gi} need to be composed into a single synthesis membership function $\mu_D(x)$ by applying some algebraic operators. One possible method is to assign weights to each membership function, thereby giving a relative importance to each of them [2], and this has been implemented in this work. Mathematically, it can be given by:

$$\mu_D(x) = \sum_{i=1}^{m} w_i \mu_{fi} + \sum_{i=1}^{n} w_i \mu_{gi}, \qquad (4.6)$$

where w_i and w_j are the weights giving the relative importance of each of the objectives and the constraints. The single synthesis membership function $\mu_D(x)$ can be thought of as a global design quality measure or a figure of merit for a particular design as a function of the design parameters x. Ideally $\mu_D(x) = 1$, which indicates that all the design objectives and constraints are met. On the other hand, $\mu_D(x) = 0$ indicates that there is no feasible solution for the design specifications. Finally, the formulation of the design problem becomes:

$$\text{Maximize: } \mu_D(x)$$
$$\text{subject to: } x_{min} \le x \le x_{max}. \qquad (4.7)$$

This formulation has a number of advantages over Eqn.(4.1). Since the maximum value of $\mu_D(x)$ is known to be equal to unity, hence, each iteration gives an idea about how far one is from the optimal solution. It interacts more with the user, since one can manipulate the shape of the membership functions and their compositions. All the objectives and constraints are taken care of in the clustered objective function making the formulation a generalized one.

5. Design Examples

In order to determine the viability of the proposed approach we have designed three commonly used topologies of op-amps. These are the simple op-amp also known as operational transconductance amplifier (OTA), the basic two-stage (BTS) op-amp, and the symmetrical OTA. The computation of the DC operating point is not a straightforward task for MOS circuits when the channel length modulation parameter (λ) is taken into consideration. To obtain the quiescent point, the equations describing the circuit behavior are solved iteratively. Our approach is similar to the one proposed in [21]. It is observed that in CMOS analog circuits, current through the current source transistors essentially determines the dc currents through all the transistors. Then, from the respective drain currents and the sizes of the transistors, their gate-to-source voltages can be obtained, and, subsequently, various node voltages can be found. In the first step, the parameters ($1+\lambda V_{DS}$) and threshold voltages (V_t) are updated based on the V_{DS} and V_{SB} values obtained in the previous iteration using the following expression:

$$V_t = V_{to} + \gamma\left(\sqrt{V_{SB} + 2\varphi_F} - \sqrt{2\varphi_F}\right), \qquad (5.1)$$

where V_{to} is the threshold voltage of the transistor at zero back bias, and V_t is the threshold voltage with back bias. The parameters V_{to}, γ, and φ_F are technology dependent, and are taken from a technology file (0.8 μ n-well process [24]). In the next step, the drain currents (I_D) of all the transistors are computed using the circuit knowledge of the topology. The gate-to-source voltages (V_{GS}) are determined from I_D, (W/L), ($1+\lambda V_{DS}$), and V_t of the transistors using the following expression:

$$V_{GS} = V_t + \left[\frac{I_D}{k'\left(\frac{W}{L}\right)(1+\lambda V_{DS})}\right]^{\frac{1}{2}}. \qquad (5.2)$$

The value of k' is determined from the technology dependent parameters μ and C'_{ox}. In the subsequent step, from the gate to source voltages of the transistors, the various node voltages (V_n) are determined. In the final step, V_{DS} and V_{BS} values of all the transistors are evaluated from the node voltages obtained from the previous step. In the subsequent iterations, the values of V_{DS} and V_{BS} are used to obtain a more accurate estimate of the operating point. When for each node voltage, the results obtained from two consecutive iterations are very close to each other (10^{-6} V in our program), the loop terminates. We have used Shichman-Hodges model [24] for all computational purposes. For brevity, we have described some of the steps in the design of the basic two-stage op-amp and presented the results for the other two.

5.1 The Basic Two-Stage Op-amp

The schematic of the basic two-stage CMOS op-amp is shown in Fig.5.1.1. The design variables for this circuit are the lengths (L) and widths (W) of the transistors, the bias current source I_{bias}, and compensating capacitor C_C. The supply voltages V_{DD} and V_{SS}, and the load capacitance (C_L) are taken as constants specified by the user. In order to get the independent design variables, which are to be varied during the optimization process, the following assumptions are made. The input transistors M1 and M2 are matched differential pairs, and, therefore, $(W/L)_1 = (W/L)_2$. The transistors M3 and M4 are current sources also acting as active load, and, hence, $(W/L)_3 = (W/L)_4$. The bias current source consists of transistors M7 and M8, and, therefore, $(W/L)_7 = (W/L)_8$. The PMOS load transistor M5 of the gain stage has independent W/L ratio, and it determines the current though that stage. The aspect ratio of the driver M6 depends on the current through the second stage. The drain-source voltage of M6 is fixed (as the DC output node voltage is taken to be equal to zero), and its gate-source voltage is obtained from the drain-source voltage of M4, and, therefore, only its length L6 is taken to be an independent variable. Hence, the independent design variables of the above circuit topology are W1, L1, W3, L3, W7, L7, W5, L5, L6, C_C, and I_{bias}. The mathematical equations used

for the computation of the various performance specifications can be found in [25].

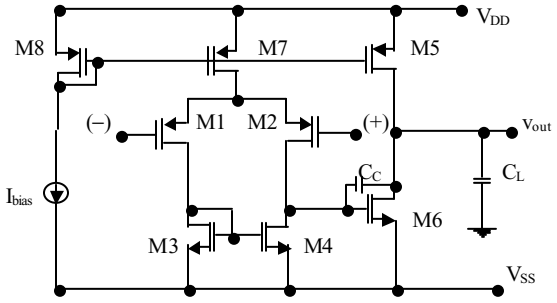

Fig.5.1.1: Schematic of the basic two-stage CMOS op-amp.

In order to determine the initial values of the design variables, an algorithm is developed and implemented. The algorithm deduces the design variables from the performance specifications specified by the user and the circuit knowledge using the simple square-law current equation of the MOS transistors. The equations used for the purpose can be found in [26]. These initial values are supplied to the optimization algorithm in order to solve the objective function. The output results obtained from the optimization program for a set of performance specifications along with the results obtained from SPICE simulation are listed in Table 5.1.1. The supply voltages V_{DD} and V_{SS}, and the value of the load capacitance (C_L) are taken to be ± 5 V and 10 pF respectively. The values of the PSRR and the RMS noise are calculated at frequencies of 100 kHz and 1 kHz respectively. The values of all the parasitic coupling capacitances required for the calculation of the power-supply rejection ratios are taken to be 0.2 pF. The design variables obtained from the optimization program and subsequently used for SPICE simulation are presented in Table 5.1.2. The constants used for the other two op-amp topologies are same as those mentioned in this section.

Table 5.1.1 Design results along with SPICE simulations

Specification	*Type*	*Wt*	*Target*	*Design*	*SPICE*
1. Gain (dB)	≥	0.1	90	93.133	96.71
2. UGF (MHz)	≥	0.1	10	10.155	13.2
3. PM (degree)	≥	0.05	45	46.72	42.3
4. SR (V/µs)	≥	0.1	5	6.084	7.16
5. CMR+ (V)	≥	0.05	3	3.409	3.435
6. CMR– (V)	≤	0.05	-3	-4.935	-4.913
7. Voutmax (V)	≥	0.05	3.5	4.712	4.675
8. Voutmin (V)	≤	0.05	-3.5	-4.935	-4.913
9. CMRR (dB)	≥	0.1	90	99.609	103.96
10. $PSRR_{DD}$ (dB)	≥	0.05	60	119.763	117.38
11. $PSRR_{SS}$ (dB)	≥	0.05	60	59.998	61.35
12. RMS noise (nV/√Hz)	≤	0.05	10	8.662	8.547
13. PD (mW)	≤	0.1	5	1.474	1.546
14. AREA (μ^2)	≤	0.1	500	462.874	462.874

Table 5.1.2 Design variables

Sl. No.	*Design variable*	*Obtained value*
1.	W/L of M1, M2 (µm/µm)	112.413/0.8
2.	W/L of M3, M4 (µm/µm)	80.21/0.93
3.	W/L of M5, M7, M8 (µm/µm)	17.444/0.989
4.	W/L of M6 (µm/µm)	102.42/0.8
5.	Compensating capacitor C_C (pF)	7.953
6.	Bias current (Ibias) (µA)	48.38

5.2 Single Stage Operational Amplifier

Figure 5.2.1 shows the schematic of a single stage CMOS op-amp. The design results obtained from the optimization routine along with those obtained from SPICE simulations are presented in Table 5.2.1. The design parameters are given in Table 5.2.2.

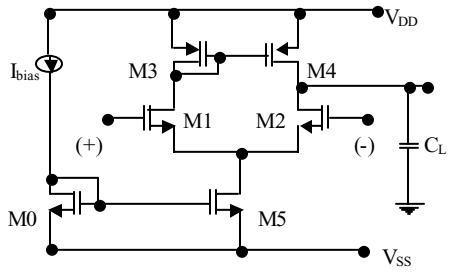

Fig 5.2.1: Schematic of a single stage op-amp.

Table 5.2.1 Design results along with SPICE simulations

Specification	*Type*	*Wt*	*Target*	*Design*	*SPICE*
1. Gain (dB)	≥	0.1	45	51.15	51.49
2. UGF (MHz	≥	0.1	10	14.08	14.2
3. PM (degree)	≥	0.05	80	82.11	85.5
4. SR (V/µs)	≥	0.1	5	4.98	5.19
5. CMR+ (V)	≥	0.05	3.0	4.55	4.3
6. CMR– (V)	≤	0.05	-3.0	-3.47	-3.7
7. Voutmax (V)	≥	0.05	3.5	4.277	4.366
8. Voutmin (V)	≤	0.05	-3.5	-4.439	-4.381
9. CMRR (dB)	≥	0.1	60	89.617	89.915
10. $PSRR_{DD}$ (dB)	≥	0.05	50	50.507	50.00
11. $PSRR_{SS}$ (dB)	≥	0.05	50	54.537	55.21
12. RMS noise (nV/√Hz)	≤	0.05	10	4.334	4.536
13. PD (mW)	≤	0.1	5	0.921	0.879
14. AREA (μ^2)	≤	0.1	200	145.88	145.88

Table 5.2.2 Design variables for the simple op-amp

Sl. No.	*Design variable*	*Obtained value*
1.	W/L of M1, M2 (µm/µm)	70.453/0.824
2.	W/L of M3, M4 (µm/µm)	1.967/1.232
3.	W/L of M0, M5 (µm/µm)	5.728/2.108
4.	Bias current (Ibias) (µA)	41.807

5.3 Symmetrical Op-amp

Figure 5.3.1 shows the schematic of a symmetrical CMOS op-amp. The design results obtained from the optimization module along with SPICE simulations results are presented in Table 5.3.1. The design variables are given in Table 5.3.2.

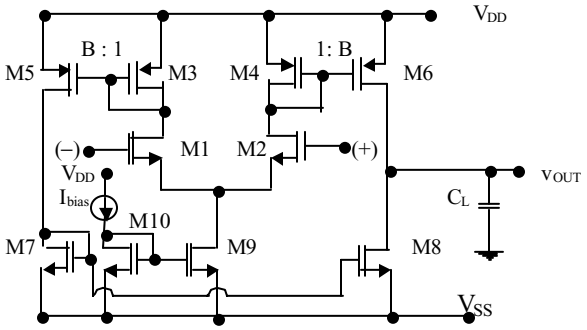

Fig.5.3.1: Schematic of a symmetrical CMOS op-amp.

Table 5.3.2 Design variables

Sl. No.	*Design variable*	*Obtained value*
1.	W/L of M1, M2 (µm/µm)	97.442/0.8
2.	W/L of M3, M4 (µm/µm)	1.605/3.152
3.	W/L of M5 (µm/µm)	2.991/3.152
4.	W/L of M6 (µm/µm)	3.098/3.152
5.	W/L of M7 (µm/µm)	1.426/1.352
6.	W/L of M8 (µm/µm)	1.329/3.152
7.	W/L of M9, M10 (µm/µm)	1.652/1.849
8.	Bias current (Ibias) (µA)	50.2

Table 5.3.1 Design results along with SPICE simulations

Specification	*Type*	*Wt.*	*Target*	*Design*	*SPICE*
1. Gain (dB)	≥	0.1	60	61.637	60.62
2. UGF (MHz)	≥	0.1	20	30.125	29.0
3. PM (degree)	≥	0.05	50	67.88	62.0
4. SR (V/µs)	≥	0.1	5	5.002	5.35
5. CMR+ (V)	≥	0.05	3.0	3.71	3.835
6. CMR− (V)	≤	0.05	−3.0	-3.11	-3.16
7. Voutmax (V)	≥	0.05	3.5	3.71	3.835
8. Voutmin (V)	≤	0.05	-3.5	-3.71	3.835
9. CMRR (dB)	≥	0.1	60	60.65	79.37
10. $PSRR_{DD}$ (dB)	≥	0.05	50	69.189	72.337
11. $PSRR_{SS}$ (dB)	≥	0.05	50	64.412	65.21
12. RMS noise (nV/√Hz)	≤	0.05	10	4.166	4.259
13. PD (mW)	≤	0.1	5	2.001	2.14
14. AREA (μ^2)	≤	0.1	200	200.03	200.0

6. Discussions and Conclusion

In this work we have applied the concept of fuzzy membership function in order to model the real world terms, e.g., *high, maximize, minimize,* etc., used by the designer while specifying the performance objectives in the design of operational amplifiers. The proposed formulation takes care of both the objectives and constraints in order to formulate a generalized objective function. The use of analytical equations makes the simulation faster in side the optimization loop. Also, the *channel length modulation parameter* is considered exclusively for the computation of dc operating point and small signal parameters, which has a significant impact on the design of deep sub-micron transistors. The design obtained from our program can be fine-tuned using circuit simulation based approach [15] with the use of more accurate device models (e.g., BSIM3 and the more recent BSIM4).

A number of issues need to be addressed in order to make the prototype tool developed in this work more efficient. The proposed method may not lead to the globally optimal solution. As long as the solution satisfies the desired performance specifications the algorithm stops exploring alternative designs the search space. As a part of further development, we are exploring the use of geometric optimization techniques [21,23] for obtaining globally optimal designs. With the shrinking device sizes, more accurate analytical expressions are needed inside the optimization loop. These equations of performance parameters can be derived using a symbolic simulator [13] and thus can be automated for any circuit topology. However, there is always a trade-off between performance and accuracy. Exact model equations tend to be complex and require more CPU time. This can be taken care of by dynamically changing the complexity of models as the algorithm converges to its solution, and, needs to be implemented. In this work, we have used a constant channel length modulation model, however, today's technology might require a more efficient one in order to obtain accurate results. Also parasitic capacitances due to the interconnects need to be extracted from the layout and incorporated during the optimization process. Automatic topology selection is another task that can be implemented on the top of this tool, and the designer can choose the best possible circuit topology for a specific application.

We have developed the optimization modules for three op-amp topologies. However, this approach can be easily extended to other op-amp topologies and cell level analog blocks.

7. References

1. L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision process," *IEEE Trans. System, Man and Cybernetics*, vol. SMC-3, pp.28-44. January 1973.
2. M. Fares and B. Kaminska, "FPAD: A fuzzy nonlinear programming approach to analog circuit design," *IEEE Trans. Computer-Aided Design*, vol. 14, no. 7, pp. 785–793, July 1995.
3. B. Martin, "Automation comes to analog," *IEEE Spectrum*, pp. 70-75, June 2001.
4. M. Ismail and J. Franca, *Introduction to Analog VLSI Design Automation*, Kluwer Academic Publishers, London, 1990.
5. L. R. Carley and R. A. Rutenbar, "How to automate analog IC design," *IEEE Spectrum,* vol. 25, no. 8, pp. 26–30, August 1988.
6. J. H. Huijsing, R. J. Plassche, and W. Sansen, *Analog Circuit Design*, Kluwer Academic Publishers, London, 1993.
7. R. Harjani, R. A. Rutenbar, and L. R. Carley, "OASYS: A framework for analog circuit synthesis," *IEEE Trans. Computer-Aided Design*, vol. 8, no. 6, pp. 680–691, June 1990.
8. F. El-Turkey and E. E. Perry, "BLADES: An artificial intelligence approach to analog circuit design," *IEEE Trans. Computer-Aided Design*, vol. 8, no. 12, pp. 1247–1265, December 1989.
9. M. R. Degrauwe, O. Nys, and H. J. Oguey, "IDAC: An interactive design tool for analog CMOS circuits," *IEEE J. Solid-State Circuits*, vol. SC−22, no. 6, pp. 1106–1116, December 1987.
10. H. Y. Koh, C. H. Sequin, and P. R. Gray, "OPASYN: A compiler for CMOS operational amplifiers," *IEEE Trans. Computer-Aided Design*, vol. 9, no. 2, pp. 113–125, February 1990.
11. H. Odonera, H. Kanbara, and K. Tamaru, "Operational-amplifier compilation with performance optimization," *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 466–473, April 1990.
12. G. G. E. Gielen, H.C.C. Walscharts, and W. M. C Sansen, "Analog circuit design optimization based on symbolic simulation and simulated annealing," *IEEE J. Solid-State Circuits*, vol. SC−25, no. 3, pp. 707–713, June 1990.
13. G. G. E. Gielen, H. C. C. Walscharts, and W. M. C. Sansen, "ISSAC: A symbolic simulator for analog integrated circuits," *IEEE J. of Solid-State Circuits*, vol. SC−24, no. 6, pp. 1587–1596, December 1989.
14. C. Toumazou and C. A. Makris, "Analog IC design automation: Part I – Automated circuit generation: New concepts and methods," *IEEE Trans. Computer-Aided Design*, vol. 14, no. 2, pp. 218–238, February 1995.
15. W. Nye, D. C. Riley, A. Sangiovanni, and A. L. Tits, "DELIGHT.SPICE: An optimization-based system for the design of integrated circuits," *IEEE Trans. Computer-Aided Design*, vol. 7, no. 4, pp. 501–519, April 1988.
16. J. M. Shyu and A. Sangiovanni-Vincetelli, "ECSTASY: A new environment for IC design optimization," in *Proc. IEEE Inter. Conf. Computer-Aided Design (ICCAD),* pp. 484–487, 1988.
17. J. C. Lai, J. S. Kueng, H. J. Chen, and F. J. Fermandez, "ADOPT – A CAD tool for analog circuit design," *IEEE Circ. Devices*, vol. 18, no. 2, pp. 29–30, March 1988.
18. E. S. Ochotta, R. A. Rutenbar, and L. R. Carley, "Synthesis of high performance analog circuits in ASTRX/OBLX," *IEEE Trans. Computer-Aided Design*, vol. 15, no. 3, pp. 273–293, March 1996.
19. J. P. Harvey, M. I. Elmasry, and B. Leung, "STAIC: An interactive framework for synthesizing CMOS and BiCMOS analog circuits," *IEEE Trans. Computer-Aided Design*, vol. 11, no. 11, pp. 1402-1417, November 1992.
20. A. Torralba, J. Chavez, and L.G. Franquelo, "FASY: A fuzzy logic based tool for analog synthesis," *IEEE Trans. Computer-Aided Design*, vol. 15, no. 7, pp. 785–793, July 1996.
21. P. Mandal and V. Viswanathan, "A new approach for CMOS Op-Amp Synthesis," *12th International Conference on VLSI Design*, pp. 189–194, January 1999.
22. P. C. Maulik, L. R. Carley, and D. J. Allstot, "Sizing of cell level analog circuits using constrained optimization techniques," *IEEE J. Solid-State Circuits*, vol. SC−28, no. 3, pp. 233–241, March 1993.
23. M. Hensherson, S. P. Boyd, and T. H. Lee, "Optimal design of CMOS op-amp via geometric programming," *IEEE Trans. Computer-Aided Design*, vol. 20, no. 1, pp.1–21, January 2001.
24. P. R. Gray and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, 3rd Edition, John Wiley and Sons, New York, 1993.
25. K. R. Laker and W. M. C. Sansen, *Design of Analog Integrated Circuits and Systems*, McGraw-Hill, New Jersey, 1994.
26. P. E. Allen and D. Holberg, *CMOS Analog Circuit Design,* 2nd Edition, Oxford University Press, To be published.