5

Circuit Sizing and Specification Translation

5.1 Introduction

Circuit sizing is defined as the process of finding out the parameter values of the dimensions of all the transistors (channel lengths and widths) and the values of the resistors and capacitors in a circuit, such that the desired performance objectives are optimized subject to a set of constraints that have to be satisfied [68]. The circuit sizing task is defined at the cell design level of abstraction in the IC design flow. The similar task performed at the architecture design level of abstraction is referred to as the specification translation. The specification translation is defined as the task of mapping the specifications of a block (e.g., a converter) under design to the individual specifications of the sub-blocks (e.g., a comparator) of the chosen block topology, such that the complete block meets its performance objectives optimally subject to a set of constraints that have to be satisfied [66, 139]. The task of circuit sizing and specification translation are thus conceptually the same thing. The concept is represented in Fig. 5.1 [76]. This chapter presents a comprehensive overview of the fundamental concepts behind the analog circuit sizing procedure.

5.2 Circuit Sizing as a Design Space Exploration Problem

The circuit sizing problem is formally cast as a design space exploration problem, which has been briefly introduced in Chapter 2 of this text. This is considered in detail below.

5.2.1 Problem Formulations

The design variables refer to those variables which are used by the circuit designers as decision variables. Design objectives include functional objectives and performance objectives. The functional objectives need to be met by the design in order to be functionally right. The performance objectives, on the

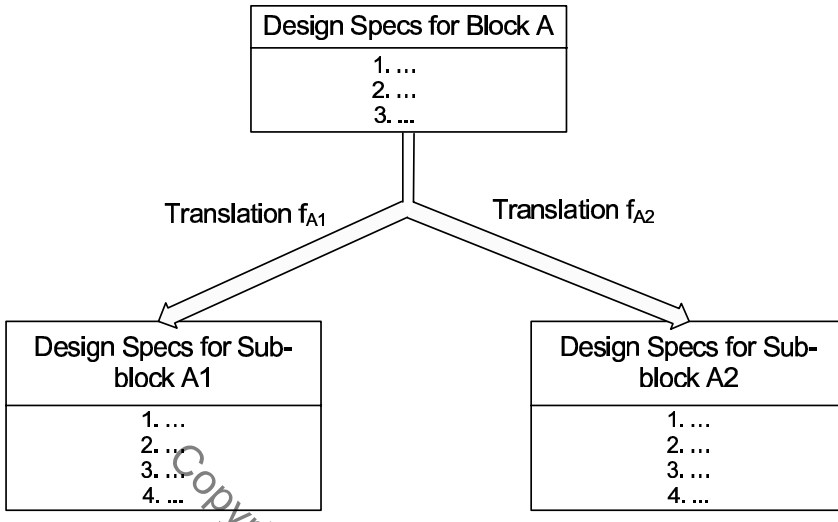


FIGURE 5.1
General design flow of a circuit sizing/specification translation task.

other hand, need to be minimized or maximized). Let us consider the circuit of an output buffer, as shown in Fig. 5.2 for illustration purpose [129]. The design variable set $\bar{\alpha}$ includes transistor dimensions and passive component values. The possible functional objective set is [129] $\bar{\rho_f}$ which includes (1) DC Gain ($A_0 >$ target), (2) input capacitance ($C_{in} <$ target), (3) 3-dB frequency ($f_{3dB} >$ target), and (4) output swing (target $< OS <$ target). The performance objective set $\bar{\rho_f}$ may be power consumption P. The design variable set $\bar{\alpha}^T = \{\alpha_1, \alpha_2, ..., \alpha_n\}^T$ defines a multi-dimensional design space. Each of the design variables α_i is bounded within an upper and a lower boundary. The functional objective and the performance objective parameters are expressed as functions of the design variables, i.e., $A_0(\bar{\alpha}), C_{in}(\bar{\alpha}), f_{3dB}(\bar{\alpha}), OS(\bar{\alpha}), P(\bar{\alpha})$. Then the problem of circuit sizing is formulated as a constrained optimization problem; in particular, for the case of the buffer of Fig. 5.2 as,

$$\begin{aligned} \text{Minimize} \quad & P(\bar{\alpha}) \\ \text{subject to} \quad & A_0(\bar{\alpha}) > \text{target} \\ & C_{in}(\bar{\alpha}) < \text{target} \\ & f_{3dB}(\bar{\alpha}) > \text{target} \\ & \text{target} < OS(\bar{\alpha}) < \text{target} \\ \text{and} \quad & \alpha_{iL} < \alpha_i < \alpha_{iU} \quad i = 1, 2, ..., n \end{aligned} \tag{5.1}$$

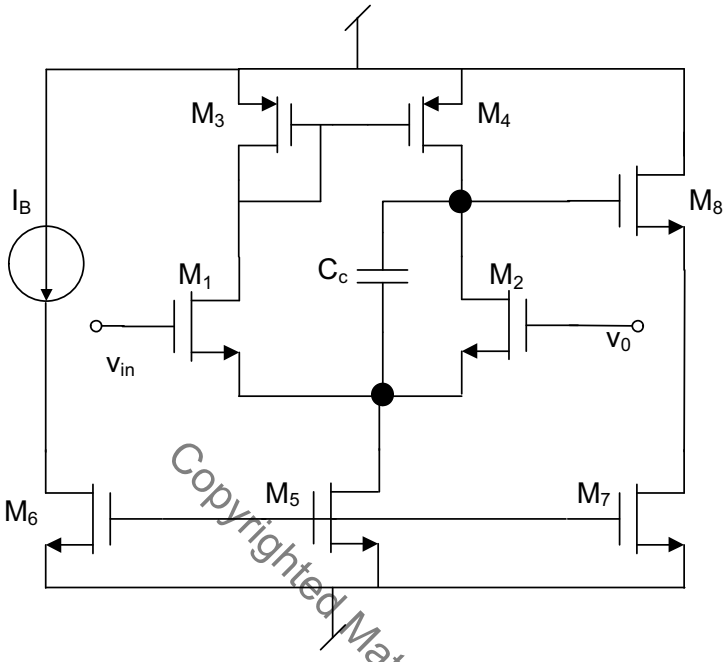


FIGURE 5.2
A CMOS output buffer circuit.

5.2.2 Solution Techniques

There are two broad categories of techniques for solving problems of (5.1). These are (1) analytical techniques and (2) iterative techniques. Unfortunately, even for elementary analog circuits like that shown in Fig. 5.2 exact analytical solution to the sizing problem is not possible. The primary reasons are

1. The design equations (i.e., the functional relationships between the various functional and performance objectives on one hand and the various design variables on the other hand, which are nothing but the performance and feasibility models) often cannot be expressed accurately in pure analytic form, especially in the nano-scale regime.
2. Even if analytical representations of the design equations are possible, these are often highly nonlinear, and consequently unsolvable analytically. The situation becomes more complicated when the dimensions of the design variable space and the design objective space become large.
3. Even if the design equations are simple, analytical solution of (5.1)

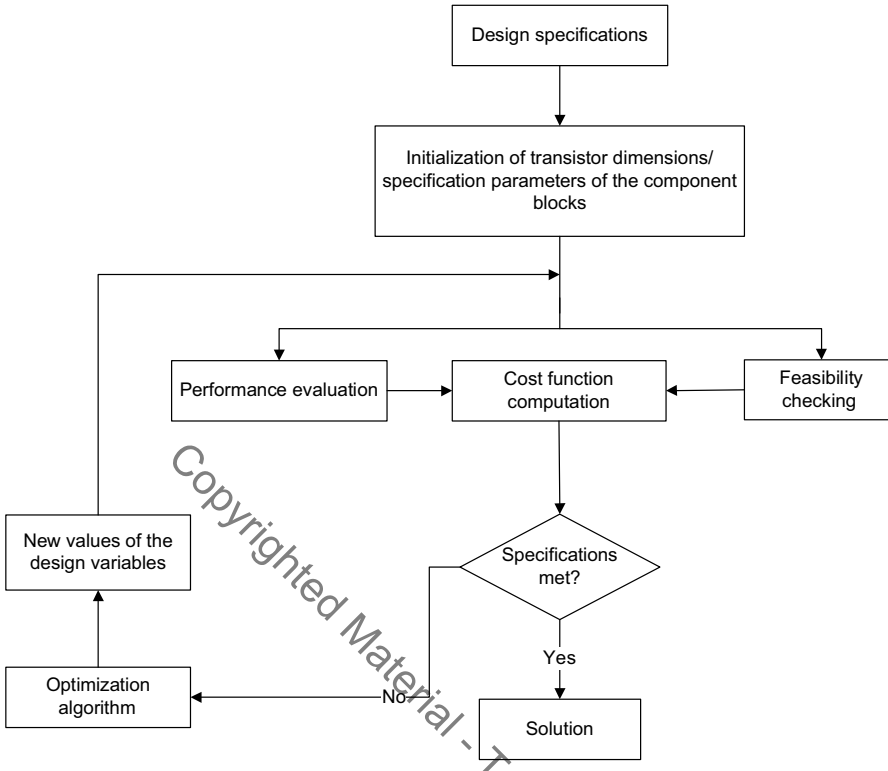


FIGURE 5.3
Design space exploration process for circuit sizing/specification translation task.

requires computation of the first and second derivatives of the design equations, which are in many cases very difficult to evaluate.

Therefore, analog circuits are most conveniently sized by using an iterative, dynamic process. This in turn can be done manually by the designers or through some automated design space exploration procedure.

5.2.3 Design Flow

The general flow of an automated design space exploration procedure for circuit sizing using optimization algorithm is shown in Fig. 5.3. The procedure starts with a set of design specifications. This consists of formal descriptions of the various functional and performance objectives and the boundaries of the design variables. The design variables are initialized to some random values within the boundaries. The various design equations (both performance models and feasibility models) are used to construct a cost function. The cost

function is evaluated with the initial values of the design variables and the evaluated results are checked against the design specifications. The subsequent values of the design variables are generated through some optimization algorithms. The process is done iteratively until a set of design variables is obtained for which the desired design specifications are satisfied.

The two important modules of said procedure are (1) evaluation of cost function through performance and feasibility models and (2) optimization algorithms. Chapter 4 of this book discusses in detail the construction of the performance and feasibility models. The implementation of the latter is discussed in the next section.

5.2.3.1 Evaluation of Cost Functions

The cost function is evaluated through some design equations. These design equations can be analytical equations or some learning network such as ANN and LS-SVM network, depending upon the complexity of the problem and the degree of accuracy required.

5.3 Particle Swarm Optimization Algorithm (PSO)

Particle Swarm Optimization is a population-based search algorithm inspired by the behavior of biological communities that exhibit both individual and social behavior; examples of these communities are flocks of birds, schools of fish, and swarms of bees. Kennedy and Eberhart introduced the concept of function-optimization by means of a particle swarm [99]. A swarm is a collection of individuals or particles. Particles are conceptual entities, which fly through the multi-dimensional design space. At any particular instant, each particle has a position and a velocity. The position vector of a particle with respect to the origin of the design space represents a trial solution of the design problem. Particles move randomly in the entire design space with velocity which is dynamically adjusted according to its own flying experience and the flying experience of the swarm. The movements of the particles are controlled by updating the position and velocity vectors of each individual particle in an effort to find the optimum solution. The position of each particle is represented by a set of coordinates in the n-dimensional (n being the number of design variables) design space of the exploration problem. The position vector of the i^{th} particle in an n-dimensional design space is given as

$$\bar{x}_i = [x_{i1}, x_{i2}, ..., x_{in}]^T \tag{5.2}$$

and the corresponding velocity vector is given by

$$\bar{v}_i = [v_{i1}, v_{i2}, ..., v_{in}]^T \tag{5.3}$$

The position corresponds to the design variables that need to be optimized. At the beginning, a population of particles is initialized with random positions and random velocities. The population of such particles is called swarm S. At each time step, all particles adjust their positions and velocities, i.e., directions in which particles need to move in order to improve their current position, thus their trajectories. The changes in the positions of the particles in the design space are based on the social-psychological tendency of individuals to emulate the success of other individuals. Thus each particle tends to be influenced by the success of any other particle it is connected to.

5.3.1 Dynamics of a Particle in PSO

There are two main versions of the PSO algorithm, local and global. In the local version, each particle moves toward its best previous position and toward the best particle within a restricted neighborhood. In the global version of PSO, each particle moves toward its best previous position and toward the best particle of the whole swarm. The global version is actually a special case of the local version where the neighborhood size is the size of the swarm. The position and velocity of each particle is updated according to the following two equations [100]

$$\bar{v}_i^{(t+1)} = \omega.\bar{v}_i^{(t)} + c_1 r_1.\left(\bar{p}_{besti}^{(t)} - \bar{x}_i^{(t)}\right) + c_2 r_2.\left(\bar{g}_{besti}^{(t)} - \bar{x}_i^{(t)}\right) \tag{5.4}$$

$$\bar{x}_i^{(t+1)} = \bar{x}_i^{(t)} + \bar{v}_i^{(t+1)} \tag{5.5}$$

where $\bar{p}_{besti}$ represents personal best experience and $\bar{g}_{besti}$ represents the best position found so far in the neighborhood of the particle. When the neighborhood size is equal to the swarm size, $\bar{g}_{besti}$ is referred to as the globally best particle in the entire swarm.

The first term in the velocity updating formula represents the inertial velocity of the particle. ω is referred to as the "inertia factor". Since it is the tendency to maintain the previous direction, it is called inertia. The second term represents the competition between the personally best position $\bar{p}_{besti}$ that each individual particle has experienced and its current position. c_1 is termed as "self-confidence" [171]. The third term represents the particle's social cognition or cooperation between the globally best position that one particle of the swarm has found and the current position of the particle. c_2 is termed as "swarm confidence" [171]. r_1 and r_2 stand for a uniformly distributed random number in the interval $[0, 1]$. These are used to give diversity to the particles. The particle updates itself constantly sharing the information both from itself and the entire swarm in such a way that enables the particles to move toward the optimum solution. The dynamics of the particle in a PSO algorithm is illustrated in Fig. 5.4.

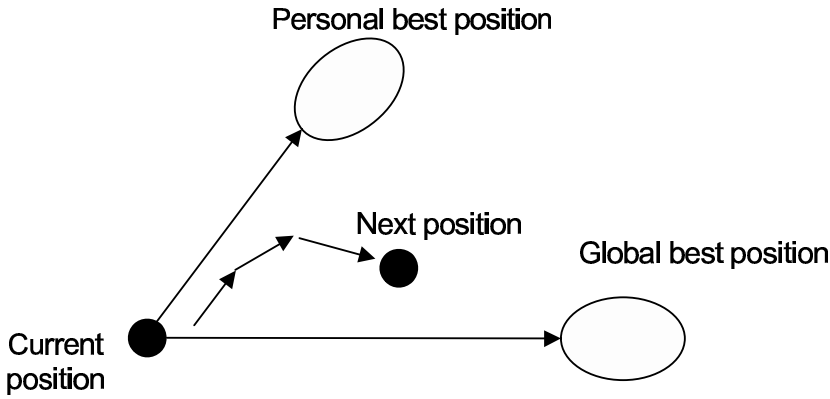


FIGURE 5.4
Illustration of the dynamics of a particle.

5.3.2 Flow of the Algorithm

The algorithm starts with random initialization of the position and velocity of all the particles in a swarm. At the start of the simulation run, this initial position of each particle is taken as the p_{best} of the respective particle and the first g_{best} is obtained from among these initial positions. A fitness function is used to search for the best position. As discussed in Chapter 2, this fitness function is computed based upon the cost function of the problem. The fitness function is computed based upon the current position of each particle. If the current fitness value of a particle is better than the corresponding p_{best}, then the p_{best} location is updated with the current position value. If the current best fitness value is better than the g_{best}, then g_{best} position is replaced by the current best position value of the entire swarm. The next position of each particle in the swarm is calculated based upon the dynamic equations (5.4) and (5.5). This iterative process continues until some termination criteria are satisfied. This process is iterated, in general, for a certain number of time steps, or until some acceptable solution has been found by the algorithms or until an upper limit of CPU usage has been reached. The flow chart of the PSO algorithm is shown in Fig. 5.5.

5.3.3 Selection of Parameters for PSO

The main parameters of a simple PSO algorithm are $\omega, c_1, c_2, V_{max}$ and the swarm size S. The settings of these parameters determine how it optimizes the design space exploration. However, it may be noted that the same parameter settings do not guarantee success in different problems. Therefore, it is essential for the designers to understand the effects of the different settings, so that it is possible to pick a suitable setting from problem to problem [39].

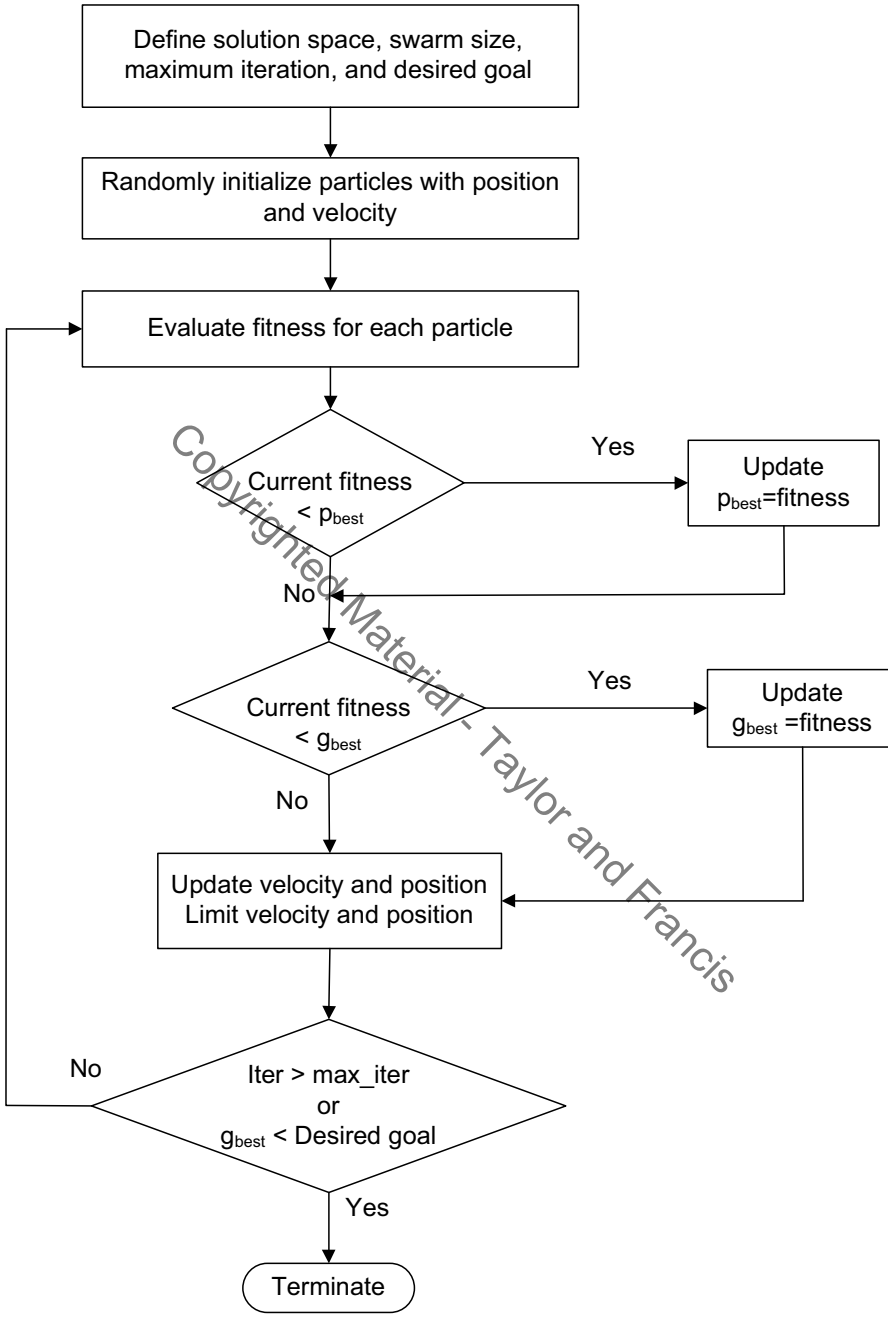


FIGURE 5.5
Flow chart of the PSO algorithm.

5.3.3.1 Inertia Weight ω

The momentum of the particle is controlled by the inertia weight. Therefore, if $\omega << 1$, quick changes of direction of the movement of a particle is possible. If $\omega = 0$, the concept of velocity is lost and the particle moves in each step without the knowledge of the previous velocity. On the other hand, setting ω high (> 1) produces the same effect as setting c_1 and c_2 low. The particles can hardly change their directions which implies a larger area of exploration as well as a reluctance against convergence toward optimum. Therefore, in short high settings near 1 facilitate global search, and lower settings in the range $[0.2, 0.5]$ facilitate rapid local search [39]. It has been reported that when V_{max} is not small (≥ 3), an inertia-weight of 0.8 is a good choice [170].

5.3.3.2 Maximum Velocity V_{max}

The maximum change that one particle can undergo in its positional coordinates during an iteration is determined by maximum velocity V_{max}. The commonly used approach is to set the entire range of the design space as the maximum velocity V_{max}. The original idea of using this parameter is to avoid explosion and divergence. However, with the use of ω in the velocity update formula, the maximum velocity parameter becomes unnecessary to some extent. Therefore, sometimes this parameter is not used. In spite of this fact, the maximum velocity limitation can still improve the search for optima in many cases [39].

5.3.3.3 Swarm Size S

A common practice in selecting the swarm size is to limit the number of particles to the range 2060 [39]. It has been shown that though there is a slight improvement of the optimal value with increasing swarm size, a larger swarm increases the number of function evaluations to converge to an error limit.

5.3.3.4 Acceleration Coefficient c_1 and c_2

An usual choice for the acceleration coefficients c_1 and c_2 is to take $c_1 = c_2 = 1.494$ [98]. An extensive study of the acceleration factor of PSO can be found in [170]. Some researchers prefer to change these parameters in an adaptive manner as follows [148]:

$$c_1 = (c_{1f} - c_{1i}).\frac{iter}{MAXITER} + c_{1i} \tag{5.6}$$

$$c_2 = (c_{2f} - c_{2i}).\frac{iter}{MAXITER} + c_{2i} \tag{5.7}$$

where c_{1i}, c_{1f}, c_{2i} and c_{2f} are constants, $iter$ is the current iteration number and $MAXITER$ is the number of maximum allowable iteration. The basic idea behind the adaptive change of the acceleration coefficients is to boost

the global search over the entire search space in the initial part of the search procedure and to encourage the particles to converge to global optima at the end of the search.

5.4 Case Study 1: Design of a Two-Stage Miller OTA

The two-stage OTA circuit that has been considered in the present case study is shown in Fig.5.6. The design variables are the transistor dimensions, bias current and the compensation capacitor C_c. The various design specifications are (1) open loop gain A_v, (2) gain-bandwidth GBW, (3) slew rate SR, (4) input common mode range $ICMR$, (5) output voltage swing and (6) power dissipation P_{diss}.

The various design equations related to the manual sizing of the two-stage CMOS OTA are summarized below based on [5]

1. From the desired phase margin, i.e., for a 60° phase margin, it is

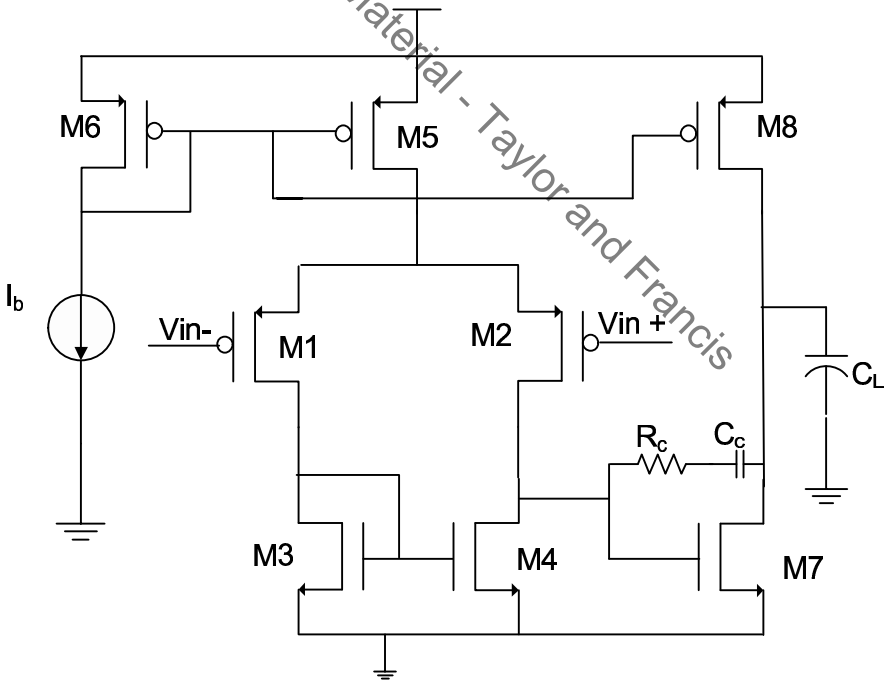


FIGURE 5.6
Schematic diagram of the Miller OTA.

chosen that

$$C_c > 0.22C_L \tag{5.8}$$

where C_c is the compensation capacitor and C_L is the load capacitor.

2. The tail current is selected from the slew rate requirement

$$I_5 = SR.C_c \tag{5.9}$$

3. Find g_{m1} from GBW and C_C using the formula

$$g_{m1} = GBW.C_c \tag{5.10}$$

Thereafter $(W/L)_1$ is calculated as

$$\left(\frac{W}{L}\right)_1 = \left(\frac{W}{L}\right)_2 = \frac{g_{m1}^2}{2K_pI_1} = \frac{g_{m1}^2}{K_pI_5} \tag{5.11}$$

where $I_1 = I_5/2$ and $K_p = \mu_0 C_{ox}$

4. From the positive ICMR requirement

$$V_{IC}(max) = V_{DD} - V_{SD5}(sat) - V_{SG1} \tag{5.12}$$

The current flowing through M1 is $I_5/2$. Therefore, it can be written that

$$V_{IC}(max) = V_{DD} - \sqrt{\frac{I_5}{\beta_1}} - |V_{T1}| - V_{SD5}(sat) \tag{5.13}$$

from which $V_{SD5}(sat)$ is calculated. Thereafter, $(W/L)_5$ is calculated as

$$\left(\frac{W}{L}\right)_5 = \frac{2I_5}{K_pV_{SD5(sat)}^2} \tag{5.14}$$

5. From the negative ICMR requirement, it follows that

$$V_{IC}(min) = V_{S1} - V_{SG1} = V_{SD1} + V_{GS3} - V_{SG1} = V_{GS3} - |V_{T1}| \tag{5.15}$$

since the transistor M1 is to remain in saturation. From this V_{GS3} is calculated, thereafter β_3 and hence $(W/L)_3$ is calculated using the following

$$V_{GS3} = \sqrt{\frac{I_5}{\beta_3}} + V_{T3} \tag{5.16}$$

Since $V_{GS3} = V_{GS4}$ and M3 and M4 form a current mirror, we have

$$\left(\frac{W}{L}\right)_3 = \left(\frac{W}{L}\right)_4 \tag{5.17}$$

6. For 60° phase margin, it is required that

$$g_{m7} \geq 10 \times g_{m1} \tag{5.18}$$

To achieve proper mirroring of the first stage current mirror load, it is required to ensure that $V_{GS3} = V_{GS7}$. It is easy to show that this requires

$$\left(\frac{W}{L}\right)_7 = \left(\frac{W}{L}\right)_3 . \frac{g_{m7}}{g_{m3}} \tag{5.19}$$

where g_{m3} is found out from

$$g_{m3} = \frac{2I_3}{V_{GS3} - V_{T3}} = \frac{I_5}{V_{GS3} - V_{T3}} \tag{5.20}$$

Hence calculate $\left(\frac{W}{L}\right)_7$

7. The current through M7 is calculated as

$$I_7 = \frac{g_{m7}^2}{2K_n \left(\frac{W}{L}\right)_7} \tag{5.21}$$

Since $I_7 = I_8$, and $V_{SG8} = V_{SG5}$, $\left(\frac{W}{L}\right)_8$ is calculated as

$$\left(\frac{W}{L}\right)_8 = \left(\frac{W}{L}\right)_5 . \frac{I_7}{I_5} \tag{5.22}$$

8. The gain is calculated as

$$A_v = \frac{2g_{m2}g_{m7}}{I_5 (\lambda_2 + \lambda_4) I_7 (\lambda_7 + \lambda_8)} \tag{5.23}$$

The PSO is utilized for design specifications of gain $A_v \geq 45dB$, phase margin $> 45^0$, $GBW \geq 2.5MHz$ $0.05V \leq ICMR \leq 0.6V$, $SR = 5V\mu s$. The target objective is to reduce the total MOS transistor area to smaller than $3\mu m^2$.

Since the target gain is not high and the target area is very small, it is preferred to take the channel length to be $0.1\mu m$. The design variables are the channel widths W and the compensation capacitor C_c. The inputs of PSO are set as $V_{DD} = 1V$, $V_{Tn} = 0.466V, |V_{Tp}| = 0.412V, K_n = 239\mu A/V^2, K_p = 36\mu A/V^2$. The swarm size is 35, $\omega = 0.99, C_1 = C_2 = 2$. The constraint functions are based on the design equations outlined above. The equality constraints are used to reduce the number of design variables. The target values of the specifications are selected so as to have sufficient guard band. The design process is iterated over 1000 epochs with a total execution time of about

TABLE 5.1
Aspect Ratios of Each Transistor of Case Study 1

Transistors	W/L
M1,M2	6.7
M3,M4	27.3
M5,M6	10.2
M7	52.69
M8	9.826
C_L	2.2pF

TABLE 5.2
Comparison between PSO and Simulation Results of Case Study 1

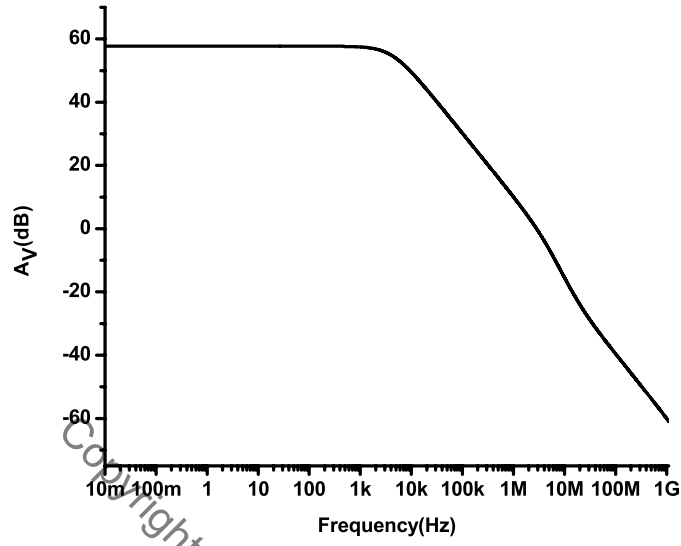
Parameters	PSO	SPICE
A_0	48 dB	57.7 dB
PM	60^0	50^0
GBW	3MHz	2.82MHz
CMRR		60.78 dB
ICMR	0.1 to 6V	0.05 to 0.60V
Slew rate	$5V/\mu s$	+ve $1.045V/\mu s$ and -ve $4.3V/\mu s$

5s with an Intel Core 2 duo processor. The (W/L) values of the various MOS transistors are tabulated in Table 5.1.

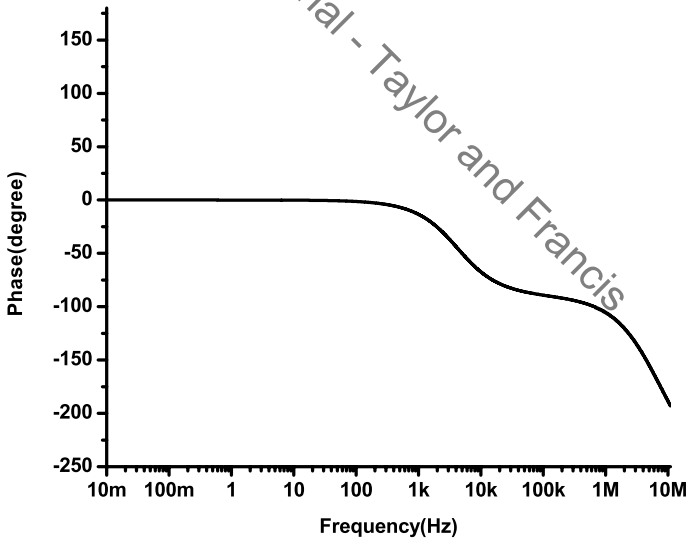
In order to validate the synthesized results, the design is implemented in a SPICE environment and simulated with 45nm CMOS technology with 1V supply. The simulation results are shown in Fig. 5.7(a), 5.7(b), 5.8(a), 5.8(b) and tabulated in Table 5.2

5.5 Case Study 2: Synthesis of on-Chip Spiral Inductors

This case study is in continuation of that described in Chapter 4 of this text. The task of on-chip spiral-inductor synthesis refers to the process of determining the layout geometric parameters from electrical specifications. The layout geometry parameters are (i) the outer diameter d, (ii) the number of turns N, (iii) the metal width W and (iv) the spacing between the metal traces s. The spiral-inductor-synthesis procedure helps the designer to make a trade-off analysis between the competing objectives, namely, Q, SRF, and outer diameter d, for a given L. The synthesis flow is shown in Fig. 5.9 [122]. The objective of the synthesis methodology is to find a set of layout parameters which will give the desired inductance value within acceptable error.

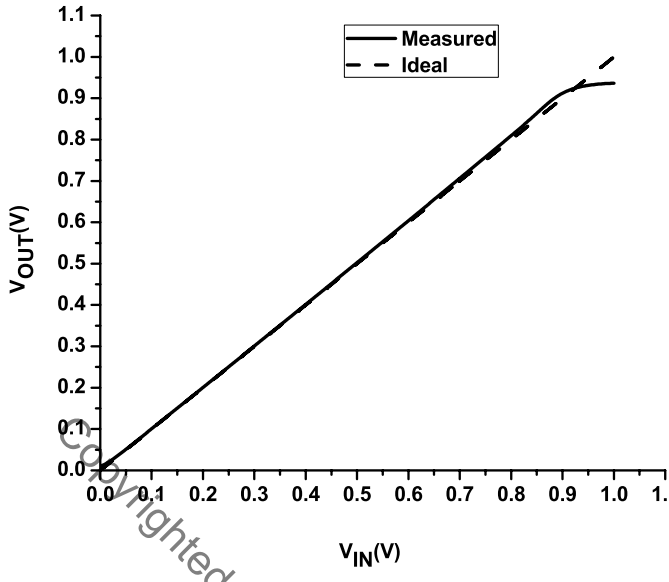


(a) Gain vs. frequency plot of the synthesized OTA in Case Study 1

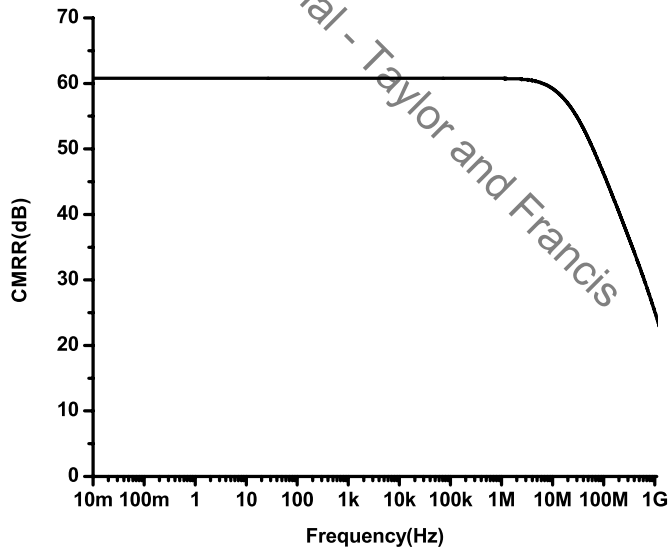


(b) Phase vs. frequency plot of the synthesized OTA in Case Study 1

FIGURE 5.7
AC simulation results of the synthesized OTA in Case Study 1.



(a) ICMR plot



(b) CMRR plot

FIGURE 5.8
ICMR and CMRR results of the synthesized OTA in Case Study 1.

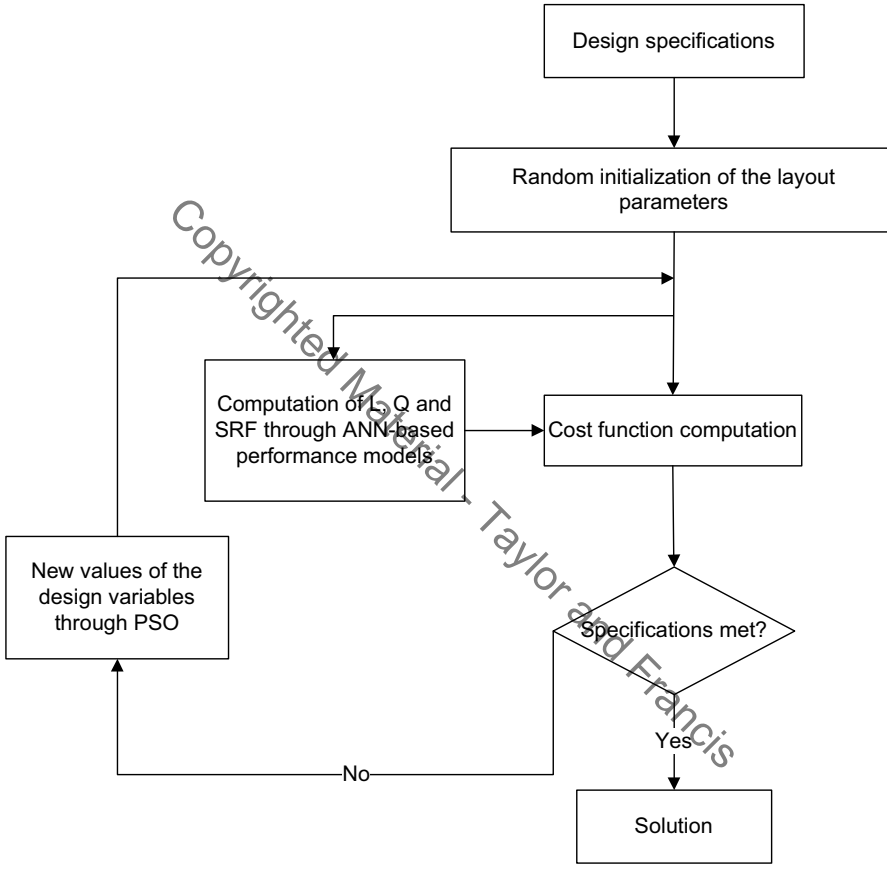


FIGURE 5.9
Flow chart of the on-chip spiral inductor synthesis problem.

TABLE 5.3
Synthesized Values of Inductor Layout Geometry Parameters

$L(nH)$	$d(\mu m)$	$W(\mu m)$	N	$s(\mu m)$	Q	$SRF(GHz)$
3.9999	275	15.1	4.0	2.4	3.6122	10.257
3.9968	284	13.1	3.0	1.4	3.639	11.587
4.0041	252	11.7	3.9	2.5	3.1102	11.207
4.0032	300	19.0	3.5	1.3	4.2953	9.1302

The design variable vector is $\bar{\alpha} = [d, N, W, s]^T$. The cost function is formulated as [122]

$$\begin{aligned} \text{Minimize} \quad & L_{target} - L_{ANN} \\ \text{subject to} \quad & N_{min} \leq N \leq N_{max} \\ & d_{min} \leq d \leq d_{max} \\ & W_{min} \leq W \leq W_{max} \\ & s_{min} \leq s \leq s_{max} \\ & d \geq 2N(W+s) - 2s \\ & SRF \geq SRF_{min} \end{aligned} \tag{5.24}$$

The variables bounds are $d = 100 - 300\mu m, W = 8 - 24\mu m, N = 2 - 6, s = 1 - 4\mu m$ and $SRF_{min} = 6GHz$.

The PSO algorithm generates a swarm of particles, each representing a combination of layout parameters in the given design space. For each combination of the design variables, the performance parameters are computed from a pre-constructed ANN-based performance model. Cost function is computed using these electrical parameter values. The design variables are then updated according to the minimum cost following the PSO algorithm. This process continues until a desired cost function objective is achieved or the maximum number of iterations is executed. The error value is set to $0.0001nH$ and the maximum number of iterations is taken to be 1000.

Table 5.3 shows the layout geometries of the inductors as synthesized by the proposed approach for a desired inductance value of $4nH$ at $1GHz$ operating frequency [122]. A set of sample 4 layout geometries are reported here. This helps the designer to make a trade-off between Q, area (outer diameter), and SRF. It is to be noted that it may not be feasible to fabricate all the inductor geometries synthesized by this approach due to the design rules of a particular process. For such cases the design values need to be rounded off to the nearest grid point while doing the layout. To validate the accuracy of the synthesis approach, the synthesized inductors are simulated with the IE3D EM simulator. The synthesized inductors satisfy the desired design specifications. This is demonstrated in Table 5.4. The L, Q, and SRF of these inductors were extracted from simulated S-parameters. The synthesized inductors show reasonable matching with the EM simulated results.

TABLE 5.4
Verification of the Synthesized Inductor Geometry through EM Simulation

	$L(nH)$	Q	$SRF(GHz)$	$d(\mu m)$	$W(\mu m)$	N	$s(\mu m)$
PSO	4.0032	4.2953	9.1302	300	19.0	3.5	1.3
EM	3.9389	4.150	9.5200	300	19.2	3.5	1.1
Error (%)	1.60	3.38	4.26				

5.6 Case Study 3: Design of a Nano-Scale CMOS Inverter for Symmetric Switching Characteristics

This case study, based on the published literature [131] presents a technique for the modeling and design of a nano-scale CMOS inverter circuit using an ANN and PSO algorithm such that the switching characteristics of the circuit is symmetric. This means that (i) the difference between the output rise time (τ_R) and fall time (τ_F) and (ii) the difference between the output propagation delay times, high-to-low (τ_{PHL}) and low-to-high (τ_{PLH}) should be minimum. The transistor channel widths W_n, W_p and the load capacitor C_L are the design parameters. The value of the rise/fall time of the input signal will be taken from the user. The problem is therefore written as [131]

$$\begin{aligned}
\text{Minimize} \quad & \frac{|\tau_F - \tau_R|}{\tau_F} + \frac{|\tau_{PHL} - \tau_{PLH}|}{\tau_{PHL}} \\
\text{subject to} \quad & (\tau_F)_{min} \le \tau_F \le (\tau_F)_{max} \\
& (\tau_R)_{min} \le \tau_R \le (\tau_R)_{max} \\
& (\tau_{PHL})_{min} \le \tau_{PHL} \le (\tau_{PHL})_{max} \\
& (\tau_{PLH})_{min} \le \tau_{PLH} \le (\tau_{PLH})_{max} \\
& 0.45 \times V_{SP} \le V_{SP} \le 0.55 \times V_{SP} \\
\text{and} \quad & (W_n)_{min} \le W_n \le (W_n)_{max} \\
& (W_p)_{min} \le W_p \le (W_p)_{max} \\
& (C_L)_{min} \le C_L \le (C_L)_{max}
\end{aligned} \tag{5.25}$$

The design flow of this circuit sizing problem is shown in Fig. 5.10. The various performance parameters are evaluated through an ANN-based performance model. Thus the PSO algorithm would result in the exact values of the design parameters which minimize the cost function value and satisfy the specified constraints. The supply voltage V_{DD} is taken to be 1.0V. The swarm size is taken to be 30. The acceleration parameters are taken as $c_1 = c_2 = 1.49618$ and the inertia weight factor is $\omega = 0.7298$. This ensures

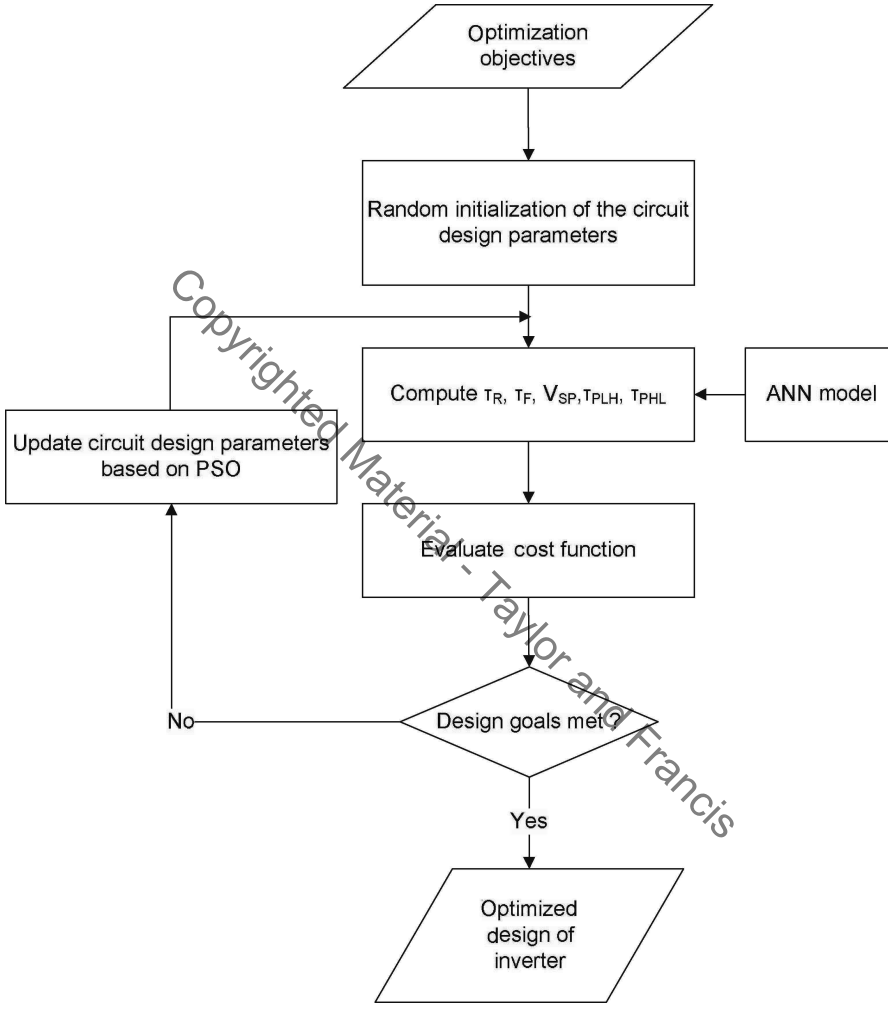


FIGURE 5.10
Flow chart of the inverter design problem.

TABLE 5.5
Delay Constraints and Design Parameter Bounds

Sample	$C_L(pF)$	$W_n(nm)$	$W_p(nm)$	$\tau_F(ns)$	$\tau_R(ns)$	τ_{PHL} (ns)	τ_{PLH} (ns)
1.	0.5-2.5	45-135	90-940	0.1-15	0.1-15	0.05-8.0	0.05-8.0
2.	0.5-2.5	45-110	90-620	0.1-15	0.1-15	0.05-8.0	0.05-8.0
3.	0.5-1.5	45-135	90-940	0.1-15	0.1-15	0.05-8.0	0.05-8.0
4.	1.0-3.0	60-160	160-945	0.1-15	0.1-15	0.05-8.0	0.05-8.0
5.	1.5-3.5	60-135	135-840	0.1-15	0.1-15	0.05-8.0	0.05-8.0
6.	0.3-2.0	45-90	90-540	0.1-8.0	0.1-8.0	0.05-6.0	0.05-6.0
7.	0.6-1.9	60-160	135-910	0.1-7.5	0.1-7.5	0.05-5.5	0.05-5.5

TABLE 5.6
Synthesis Results: $\tau_{in} = 1ns$

Sample	C_L (pF)	W_n (nm)	W_p (nm)	$\tau_F(ns)$	$\tau_R(ns)$	τ_{PHL} (ns)	τ_{PLH} (ns)	$V_{sp}(V)$
1.	0.83	128.37	221.93	5.1546	5.1363	2.5277	2.4648	0.4890
2.	0.76	69.44	109.08	10.5131	10.5820	4.8127	4.8025	0.4861
3.	0.81	125.82	217.53	5.0244	5.1146	2.6854	2.6731	0.4879
4.	1.02	157.91	273.01	5.2138	5.2015	2.5812	2.5637	0.4851
5.	1.66	134.70	232.88	8.8279	8.8588	4.6977	4.6785	0.4800
6.	0.42	65.42	113.10	5.4471	5.4233	2.5805	2.5332	0.4887
7.	0.61	95.52	165.14	5.3867	5.3678	2.7219	2.7581	0.4876

good convergence of the PSO algorithm. The maximum number of iterations that has been considered is 1000.

A set of seven samples has been chosen. For each sample, the desired rise time, fall time, low-to-high and high-to-low output propagation delay times are kept within a certain constraint, defined by an upper limit and a lower limit. Similarly, the design parameters are also kept within a specified bound. These are tabulated in Table 5.5 [131]. The value of the input rise time/fall time τ_{in} is assumed to be $1ns$. The synthesized values of the design parameters corresponding to which the cost function is minimized and the constraints are satisfied for all the case studies, are shown in Table 5.6 [131]. It also contains the corresponding values of the performance parameters. It is observed from Table 5.5 and 5.6, that the synthesized parameters satisfy the design constraints.

In order to validate the results obtained through PSO optimization, the design samples are selected and implemented at the transistor level. The PSO synthesized transistor widths and output load capacitor values have been considered. The channel length is taken as 45nm with 1.0V supply. Transient simulation is then performed using SPICE simulation. A comparison between the PSO generated results and SPICE results is provided in Table. 5.7-5.8

TABLE 5.7
Comparison between PSO Results and SPICE Results: τ_R and τ_F

Sample	PSO Results			SPICE Results		
	$\tau_R(ns)$	$\tau_F(ns)$	Diff(ns)	$\tau_R(ns)$	$\tau_F(ns)$	Diff(ns)
1.	5.1363	5.1546	0.0183	5.0674	5.2532	0.1858
2.	10.5820	10.5131	0.0689	10.6200	9.8351	0.7849
3.	5.1146	5.0244	0.09012	5.0535	5.3356	0.2821
4.	5.2015	5.2138	0.0123	5.0398	5.4025	0.3627
5.	8.8588	8.8279	0.0309	8.6485	8.5699	0.0786
6.	5.4233	5.4471	0.0238	5.2171	5.6551	0.438
7.	5.3678	5.3867	0.0189	5.2649	5.6741	0.4092

TABLE 5.8
Comparison between PSO Results and SPICE Results: τ_{PHL} and τ_{PLH}

Sample	PSO Results			SPICE Results		
	$\tau_{PHL}(ns)$	$\tau_{PLH}(ns)$	Diff(ns)	$\tau_R(PHL)$	$\tau_F(LH)$	Diff(ns)
1.	2.5227	2.4648	0.0579	2.6455	2.4367	0.2088
2.	4.8217	4.8025	0.0192	4.9417	4.9256	0.0161
3.	2.6854	2.6731	0.0123	2.8366	2.4292	0.4074
4.	2.5812	2.5637	0.0175	2.5157	2.4210	0.0947
5.	4.6977	4.6785	0.0192	4.7861	4.5264	0.2597
6.	2.5805	2.5332	0.0473	2.5771	2.4957	0.0814
7.	2.7219	2.7581	0.0362	2.8629	2.6367	0.2262

[131]. It is observed that the PSO generated designs yield very good results even when simulated at the SPICE level as far the symmetry of the switching characteristics is considered.

5.7 The g_m/I_D Methodology for Low Power Design

By now it is clear to the readers that sizing of an analog circuit while meeting simultaneously a large number of objectives like a prescribed gain-bandwidth product, minimal power consumption, minimal area, low-voltage design, dynamic range, non-linear distortion, etc., is a very difficult task. This becomes more complicated when the transistors are designed with nano-scale technology. Optimization algorithms are attractive without any doubt, but they require translating not always well-defined concepts into mathematical expressions. The interactions amid semiconductor physics and analog circuits are not always easy to implement [93].

This section presents a methodology for sizing of CMOS analog circuits so

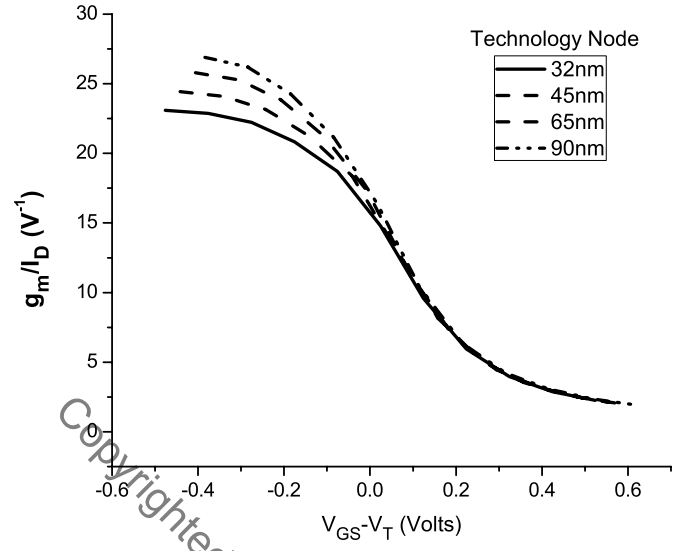
as to meet specifications such as gain-bandwidth while optimizing attributes like low power and small area. The sizing method takes advantage of the g_m/I_D ratio of a MOS transistor and makes use of a set of look-up tables. These tables are derived from physical measurements carried out on real transistors or advanced compact models such as BSIM4.

5.7.1 Study of the g_m/I_D and f_T Parameters for Analog Design

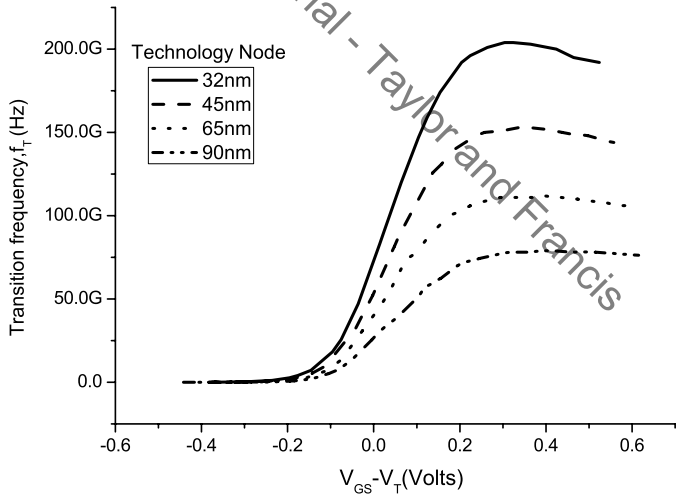
An important challenge in analog design is to achieve a good balance between the bandwidth and power efficiency of a circuit [133]. The two parameters which appear to be very much significant to the analog designers are (1) g_m/I_D and (2) $f_T = g_m/2(\pi C_{gg})$. The former signifies the amount of current to be used per transconductance and the later parameter signifies how much total gate capacitance C_{gg} must be driven at the controlling node per desired transconductance. The values of these quantities are found to be dependent on the region of operations of a MOS transistor. This is demonstrated in Fig. 5.11(a) and 5.11(b). It is observed that the g_m/I_D parameter has the maximum value in the weak inversion region and the value decreases as the operating point moves toward the strong inversion region. On the other hand, the f_T parameters has minimum value in the weak inversion region and the value increases as the operating point moves toward the strong inversion region. In addition, it is observed that the g_m/I_D parameter is not very sensitive to technology scaling. On the other hand, the values of f_T increase significantly with technology scaling.

It is interesting for the analog designers to study the variations of the product of g_m/I_D and f_T with the region of operation. This is shown in Fig. 5.12. This helps the designers to determine the overdrive voltage, i.e., $(V_{GS} - V_T)$ such that the bandwidth objective is met while operating at the corresponding maximum possible g_m/I_D (lowest power). It is observed that for a given technology node, the product quantity exhibits a "sweet spot" around a gate overdrive of 100 mV, which is a commonly found bias condition in many of today's moderate-to-high speed designs [133]. On the other hand, working with high g_m/I_D greatly helps in reducing power consumption for applications that do not demand an extremely high bandwidth. For such applications, the MOS transistors can be biased in the weak inversion region. However, operating in the weak inversion region with high g_m/I_D comes at the cost of degraded linearity performance of the transistor. This is illustrated in Fig. 5.13 , which shows the linearity performance of various technologies versus g_m/I_D. The linearity is characterized through the parameter VIP3 which represents the extrapolated gate-voltage amplitudes, at which the third-order harmonics become equal to the fundamental tone in the drain current I_D. Mathematically, this is expressed as [202]

$$\text{VIP3} = \sqrt{24\frac{g_m}{g_{m3}}} \tag{5.26}$$



(a) variation of g_m/I_D



(b) variation of f_T

FIGURE 5.11
Simulation results for the variations of g_m/I_D and f_T with the region of operation and technology nodes.

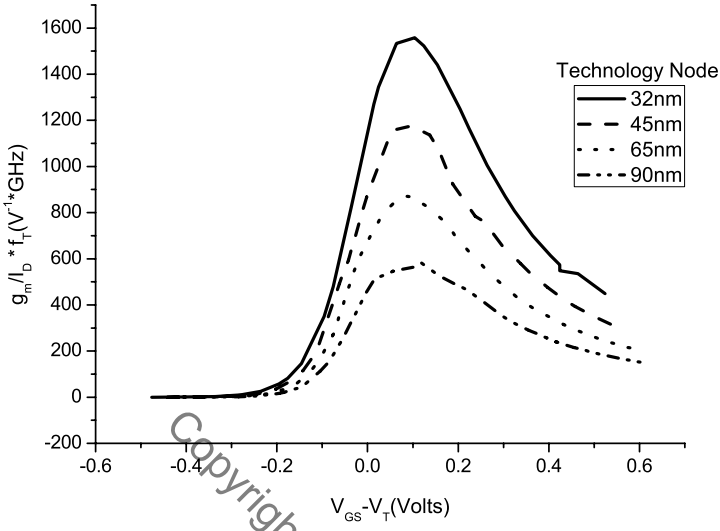


FIGURE 5.12
Simulation results showing the variations of the product of g_m/I_D and f_T with the region of operation and technology node.

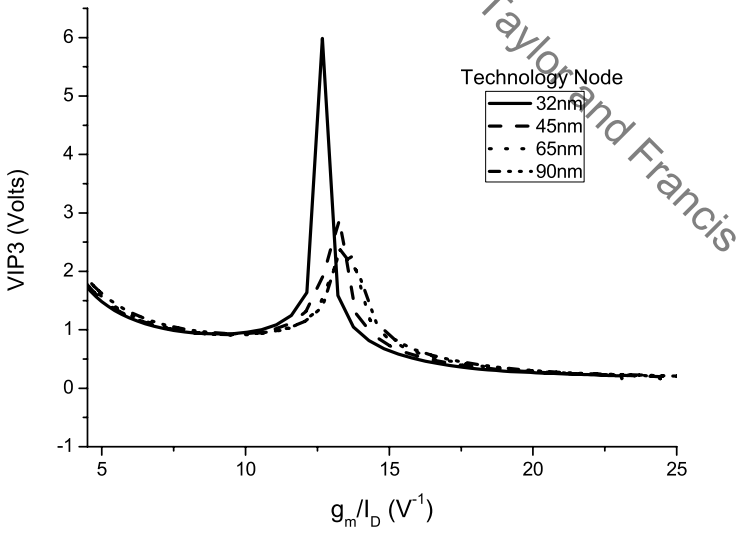


FIGURE 5.13
Simulation results showing the variations of the transconductance linearity with the region of operation and technology node.

where $g_{m3} = \dfrac{\partial^3 I_D}{\partial V_{GS}^3}$. The VIP3 peak, which is shown in Fig. 5.13, is because of the second-order-interaction effect and can be explained as a cancellation of the third-order nonlinearity coefficient by device internal feedback around a second-order nonlinearity. In practice, it is very hard to utilize this extremely linear point. It is observed that the linearity is degraded as the g_m/I_D ratio is increased.

It is also important to study the variations of the intrinsic capacitances of MOS transistors as functions of the g_m/I_D parameter. These are shown in Fig. 5.14(a) and 5.14(b). It is observed that the values of the both the capacitors are low, when g_m/I_D is high.

5.7.2 g_m/I_D Based Sizing Methodology

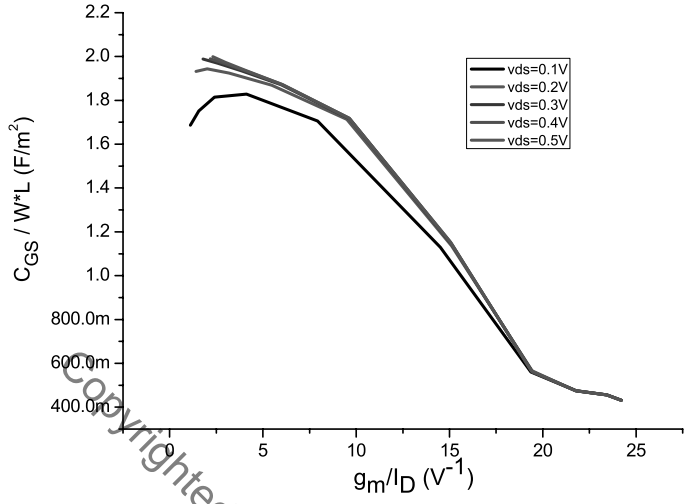
The g_m/I_D based circuit sizing procedure is based on the relation between the ratio of the transconductance over DC current g_m/I_D and the normalized current $I_N = I_D/(W/L)$[173, 34]. The selection of the g_m/I_D as the key parameter is due to three reasons. First, this parameter is strongly related to the analog performances. Second, it gives an indication of the operating region of a MOS transistor. Third, it provides a tool for calculating the transistor dimensions. This parameter is considered to be a universal characteristic of the transistors in the same process technology. The relation between the g_m/I_D parameter with the operating region of the transistor may be written as follows

$$\frac{g_m}{I_D} = \frac{1}{I_D}\frac{\partial I_D}{\partial V_{GS}} = \frac{\partial\left(\ln I_D\right)}{\partial V_{GS}} = \frac{\partial\left\{\ln\left[\dfrac{I_{DS}}{\left(\dfrac{W}{L}\right)}\right]\right\}}{\partial V_{GS}} \tag{5.27}$$

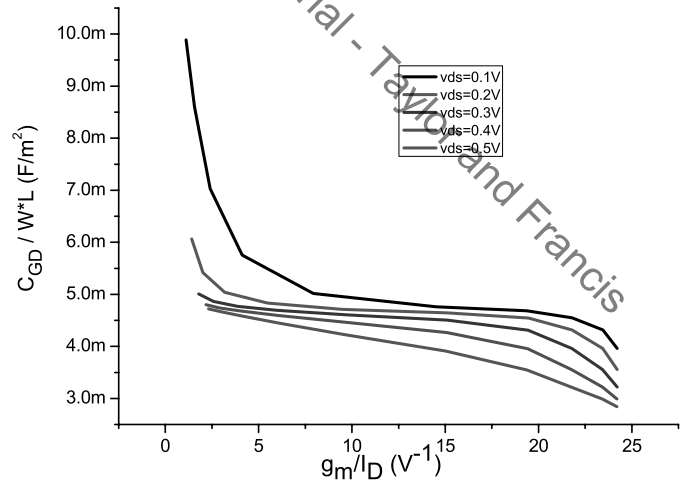
As it has been demonstrated in Fig. 5.11(a) the maximum value of the g_m/I_D ratio lies in the weak inversion region and the value decreases as the operating point moves toward strong inversion when I_D or V_{GS} are increased. It may be noted that the g_m/I_D ratio is independent of the transistor sizes. The normalized current I_N is also independent of the transistor sizes. Therefore, the relationship between the g_m/I_D and the normalized current is a unique characteristic for all transistors of the same type (n-channel MOS or p-channel MOS) in a given batch. This relationship is shown in Fig. 5.15(a) and Fig. 5.15(b). This universal characteristic of the g_m/I_D versus I_N curve is used to determine the aspect ratio of a transistor, which is then subsequently used to determine the channel width, assuming a fixed value of the channel length. The corresponding simulation graph is shown in Fig. 5.16.

For a MOS transistor, the magnitude of the intrinsic voltage gain is given by

$$A_v = g_m r_0 = \left(\frac{g_m}{I_D}\right)(I_D r_0) = \left(\frac{g_m}{I_D}\right) V_A \tag{5.28}$$

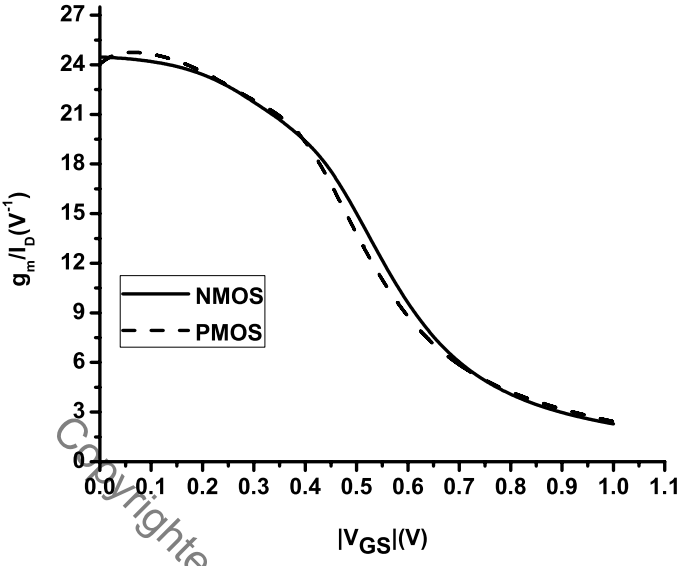


(a) variation of C_{GS}

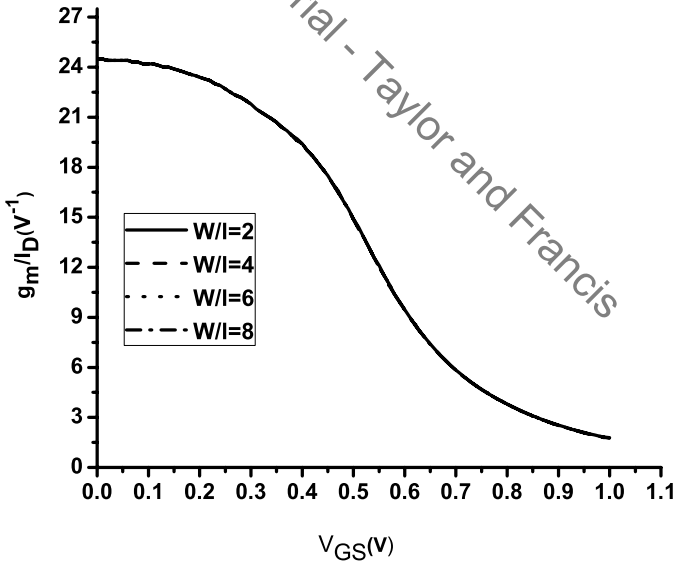


(b) variation of C_{GD}

FIGURE 5.14
Simulation results for the variations of C_{GS} and C_{GD} with the g_m/I_D.



(a) g_m/I_D for n-channel and p-channel MOS transistor



(b) g_m/I_D graph for different aspect ratios

FIGURE 5.15
Simulation results for the g_m/I_D variations.

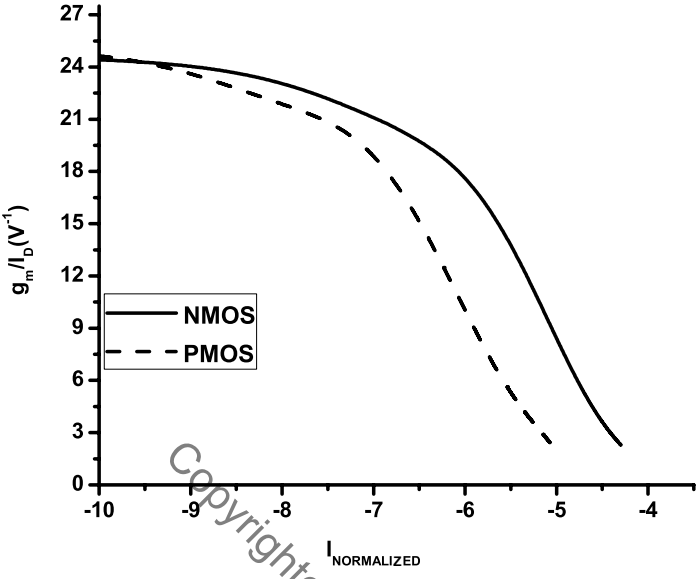


FIGURE 5.16
Simulation results showing the variations of the g_m/I_D with the normalized current I_N.

where V_A is referred to as the early voltage of the transistor. Assuming V_A to be constant for a particular channel length of a transistor, the intrinsic gain is determined by the g_m/I_D ratio. Therefore, the intrinsic gain of a MOS transistor is maximum in the weak inversion region and reduces as the operating point moves toward the strong inversion region. This is shown in Fig. 5.17. Therefore, an important guideline to get high gain for a MOS transistor, is to bias the transistor in the weak inversion region with as low V_{GS} as possible. An interesting thing observed from the curve is that under weak inversion regions very small amounts of drain current flows, which implies a small amount of power dissipation. Therefore, by biasing the MOS transistor in the weak inversion region, it is possible to obtain high gain with very small power dissipation.

The procedure for determining the aspect ratio through the g_m/I_D methodology is explained by a simple example. Let the current flow through the transistor be $I_D = 100nA$. The transistor is biased in the weak inversion region with $g_m/I_D = 21.8V^{-1}$ at $V_{GS} = 0.3V$. From the normalized current plot, it is observed that the corresponding $I_N = 41.64nA$. Therefore, the aspect ratio is found to be $W/L = 2.4$. Therefore, by assuming $L = 100nm$, the channel width is found to be $W = 0.24\mu m$. The power dissipation corresponding to a supply voltage of 0.5V will be 50nW.

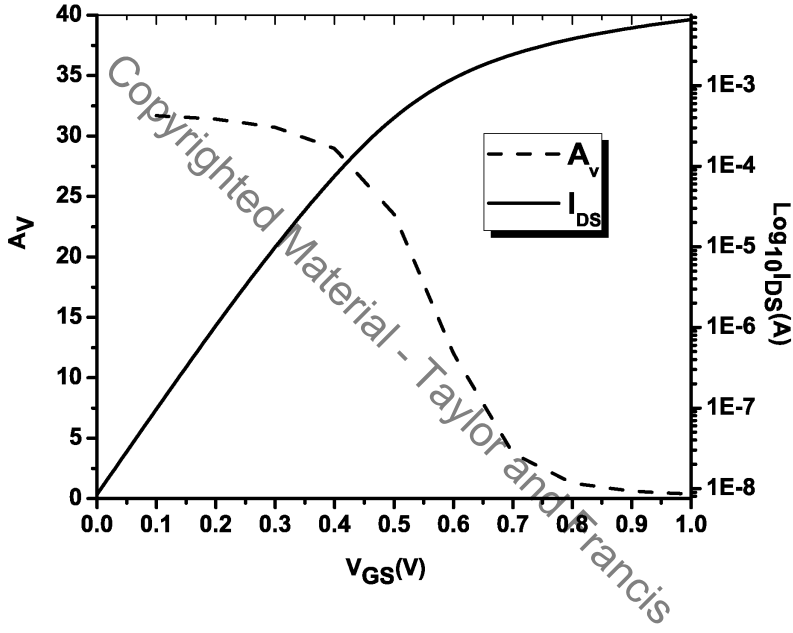


FIGURE 5.17
Simulation results showing the variations of the intrinsic gain and drain current with the region of operation.

The primary advantage of the g_m/I_D methodology over other methodologies is that this methodology uses a set of look-up tables as the main design tool, which is itself constructed from SPICE simulations. Therefore, the predicted results, as obtained after the circuit sizing process, appear to be quite close to the actual SPICE results. The same can be achieved by using ANN/LS-SVM based models. However, construction of look-up tables is much easier and less time consuming. Therefore, the g_m/I_D methodology is gaining importance day by day for nano-scale analog circuit design. In addition, the g_m/I_D methodology gives the designers the flexibility to operate the transistors in any region of operation. It may be noted that by using compact models, it is possible to compute the g_m/I_D graph analytically.

5.7.3 Case Study 4: Sizing of Low-Power Nano-Scale Miller OTA Using the g_m/I_D Methodology

The sizing methodology of a Miller OTA circuit, shown in Fig.5.6 , is based on what is presented in [173, 34]and is outlined below.

1. From the power consumption requirement, the total current I_T flowing through the circuit is calculated.
2. The compensation capacitor C_c is calculated from the 60° phase margin requirement, $C_c > 0.22C_L$.
3. The bias current is determined from the slew rate requirement. $I_b = I_5 = SR.C_c$
4. The second-stage branch current is calculated as $I_8 = I_7 = I_T - 2I_b$
5. From the gain-bandwidth requirement, the transconductance of M1 transistor is calculated as $GBW = \frac{g_{m1}}{2\pi C_c}$
6. Fix $\left(\frac{g_m}{I_D}\right)_1$ to operate the M1 transistor in weak inversion.
7. $\left(\frac{g_m}{I_D}\right)_1 = \left(\frac{g_m}{I_D}\right)_2$
8. The transistors M3 and M4 are operated in the weak inversion region since the current flowing through these transistors is small. Select the $\left(\frac{g_m}{I_D}\right)_3$ and $\left(\frac{g_m}{I_D}\right)_4$ sufficiently high.
9. The transistors M5, M6 and M8 are similarly made to operate in the weak inversion region and the $\left(\frac{g_m}{I_D}\right)$ ratios are determined accordingly.
10. From the relation $g_{m7} >> 10.g_{m1}$, find out g_{m7} and hence $\left(\frac{g_m}{I_D}\right)_7$

TABLE 5.9
Aspect Ratios and g_m/I_D Ratio of Each Transistor of Case Study 4

Transistor	W/L	g_m/I_D
M1	230	24.7
M2	230	24.7
M3	4	23.43
M4	4	23.43
M5	459	24.7
M6	459	24.7
M7	14	22.73
M8	2526	24.7

TABLE 5.10
Comparison between Analytical and Simulation Results for Case Study 4

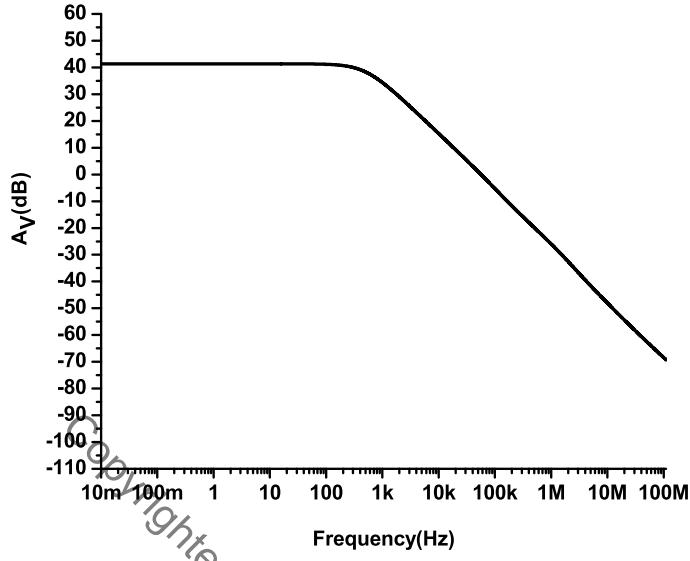
Performances	Analytical	Simulation
Gain (db)	60	41.4
GBW (KHz)	50	56
SR (V/μs)	0.025	0.02
CMRR(db)		71.9
ICMR(V)		(0.065 to 0.9)
Total current (nA)	300	299

11. Once the $\left(\frac{g_m}{I_D}\right)$ of all transistors and the corresponding drain currents are known, the normalized current are determined from the g_m/I_D vs I_N graph and hence the $\left(\frac{W}{L}\right)$ ratios for each transistor are determined from the corresponding normalized currents and drain currents.

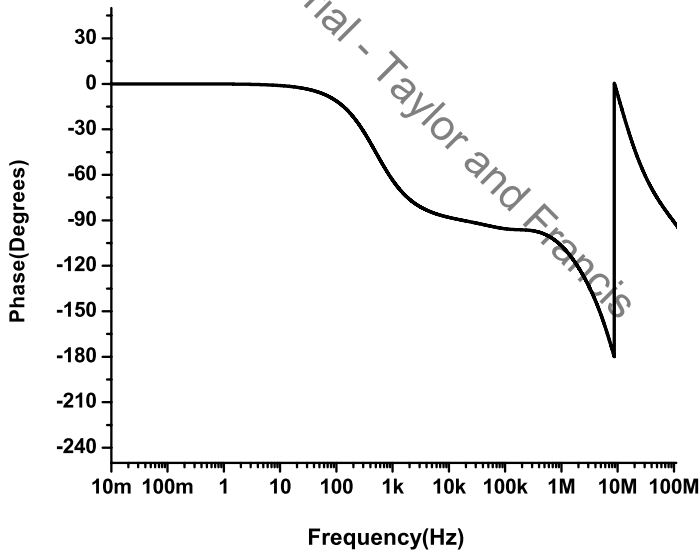
It may be noted that an important issue related to the operation of MOS transistors in a weak inversion region is that under this condition, the drain current mismatch due to threshold voltage mismatch is maximum. Therefore, often the transistors involved in the current mirror circuit are not operated in the weak inversion region. Therefore, the g_m/I_D ratios of such transistors are to be selected accordingly.

With the present methodology it is attempted to design a two-stage Miller OTA with gain $A_v > 40dB$, gain bandwidth product $GBW \geq 40KHz$, phase margin $PM > 60^0$, slew rate $SR = 25V/ms$ and power dissipation ¡350nW.

The transistor length is considered to be $0.1\mu m$. The length can be increased, if higher gain is required. The aspect ratios as well as the (g_m/I_D) ratio of each transistor are tabulated in Table. 5.9. The circuit is simulated using 45nm,1V CMOS technology using HSPICE simulation tool. Table 5.10

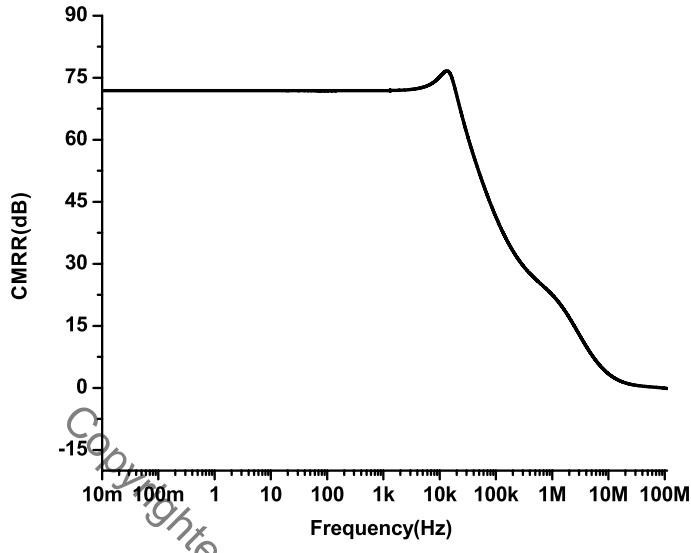


(a) Gain vs. frequency plot of the synthesized OTA

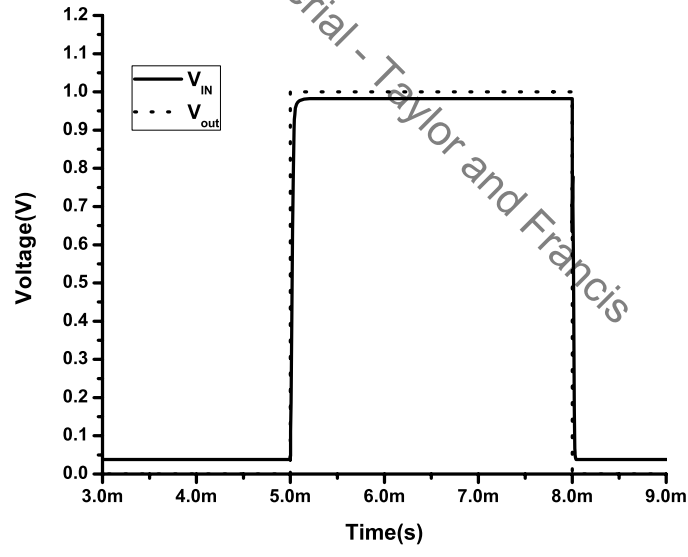


(b) Phase vs. frequency plot of the synthesized OTA

FIGURE 5.18
AC simulation results of the synthesized OTA in Case Study 4.



(a) CMRR



(b) Slew rate

FIGURE 5.19
CMRR and slew rate of the synthesized OTA in Case Study 4.

shows the comparison between the performances as calculated analytically and as obtained from SPICE simulation results. The AC simulation results illustrating the variations of the gain and phase with frequency for the synthesized OTA are shown in Fig. 5.18(a) and Fig. 5.18(b). The CMRR and the slew rate is obtained from Fig. 5.19(a) and Fig. 5.19(b) respectively. The difference between the analytical and simulation results occurs primarily because of the fact that the variations of the early voltage with drain bias are not taken into consideration in this method. This is an important issue in nano-scale design and needs to be taken care of judiciously.

5.8 High-Level Specification Translation

At the architectural level of abstraction of a hierarchical analog design methodology, the overall architecture of the system is first decomposed into several component blocks. The specifications of these component blocks are then derived from the specifications of the complete system so that they can be designed separately. This is referred to as the process of high-level specification translation [139]. For example, if the system to be considered is a $\Sigma-\Delta$ ADC, then at the architecture level, the component blocks are the integrator, comparator, etc. The specifications of these component blocks need to be derived such that the system specifications are optimally satisfied. On the other hand, at the same time, it has to be ensured that the specifications of the component blocks can actually be realized in practice, finally when the various component blocks are to be implemented using transistor-level circuits. Therefore, the task of high-level specification translation is a challenge to the analog designers.

The task of construction of feasible design space as an intersection of an application bounded space (constructed by top-down procedure through interval analysis technique) and circuit realizable space (constructed by bottom procedure through actual circuit simulation) and subsequent identification through LS-SVM classifier method has been discussed in [139]. This has been described in detail in Chapter 4 of this text. The identified feasible design space needs to be explored through any design space exploration algorithm such as the particle swarm optimization algorithm or genetic algorithm etc., in order to find out the various values of the design parameters. Recently a geometric programming-based methodology has been used for high-level specification translation [38].

5.9 Summary and Conclusion

This chapter discusses in detail the task of automated sizing of analog circuits. The particle swarm optimization algorithm has been described as a popular design space exploration algorithm. This is also demonstrated with the case study of synthesizing an OTA circuit. The cost functions which are to be computed by the design space exploration algorithm during the sizing procedure are often made simple, based on the square law model of MOS transistors. As a result, often the synthesized results are found to deviate greatly when the designs are actually simulated through SPICE. This is sometimes judiciously avoided by considering a large guard band for the specifications. The alternative is to embed accurate models such as ANN/SVM-based learning models. These are also discussed in the present text and demonstrated through case studies. Finally, this chapter presents a look-up table based approach, based on the g_m/I_D methodology. This approach is found to be simple and suited for nano-scale analog circuit sizing, however with a scope of improvement.