DESIGN OF TWO-STAGE OPERATIONAL AMPLIFIER BY USING ARTIFICAL BEE COLONY ALGORITHM

E. Srinivas and G. Sairam
Department of ECE, Anurag University, Hyderabad, India
E-Mail: edem.srinivas@gmail.com

ABSTRACT

A two-stage operational amplifier is designed in this paper by using artificial bee colony algorithm. The aim is to minimize the MOS transistor size by using artificial bee colony (ABC) algorithm to allow automated synthesis of analog systems. This proposed methodology is used to find the optimal dimensional parameters (length and width) in order to obtain operational amplifier performances for analog CMOS (complementary metal oxide semiconductor) based circuit applications. A few parameters are considered in this study, direct current (DC), unity-gain bandwidth (GBW), phase margin (PM), power consumption (P), slew rate (SR), and area (A) using the MATLAB optimization toolbox to implement the program. Also, by using variables obtained from ABC algorithm, the two-stage operational amplifier is simulated by using cadence virtuoso spectra circuit simulator in standard GPDK 180nm CMOS technology. A good agreement is observed between the program optimization and electric simulation.

Keywords: two-stage operational amplifier, particle swarm optimization, genetic algorithm and artificial bee colony algorithm.

INTRODUCTION

Designing an analog circuit in VLSI domain will plays a crucial part. Implementation and design a two-stage CMOS operational amplifier are more difficult to get considerable DC gain with high unity gain frequency. The advantages of two-stage CMOS operational amplifier have good gain, high output swing, low noise and good bandwidth over folded cascode. And it needs compensation, low PSRR value compared to folded cascode. The designed circuit is to meet the required specifications is showninfig1.

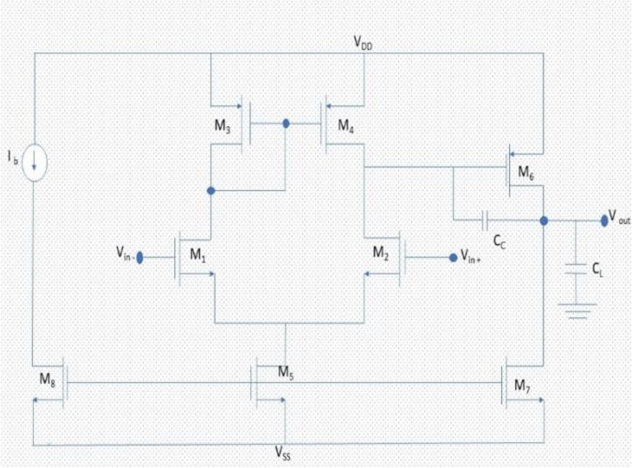

Figure-1. Two-stage CMOS operational amplifier.

The method uses for program based on multi objective optimization using ABC algorithm to calculate the optimal transistors dimensions, length, and width of an operational amplifier in Figure-2 which is used as part of an electronic front-end for signal shaping stage. The method which handles a wide variety of specifications and constraints is extremely fast and results in globally optimal designs.

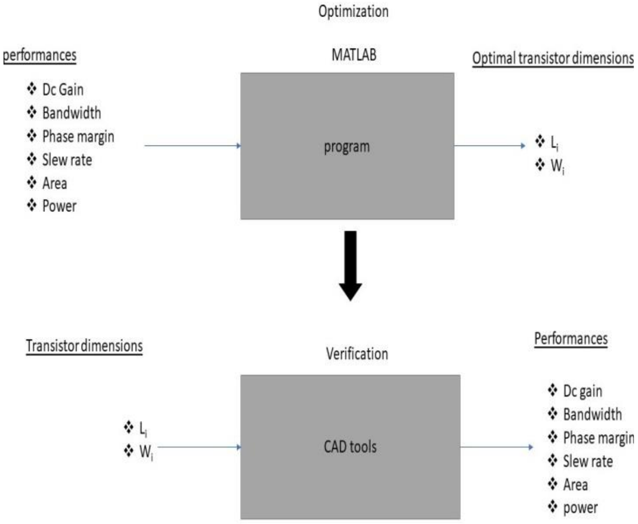

Figure-2. Operational amplifier design flow.

Design issues of CMOS op-amp are estimated by Convex Optimization ref. (1). PSO is used to design the analog VLSI circuits in ref. (2). Optimization of radio frequency (RF) circuit parameters utilizing RPSO is proposed in ref. (3). To obtain the maximum gain and UGB, folded cascode op-amp circuit is designed by PSO method in ref. (4). Simplex-PSO is utilized for the optimal selection of component values of analog active filers in ref. (5). Optimal CMOS transistor sizing for minimum area-oriented optimization, which is only a part of a complete analog circuit CAD tool remains between topology selection and actual circuit layout in ref. (6). The ant colony algorithm (ACO) is inspired from the foraging behavior of the ant colonies in ref. (7). The ABC algorithm simulates the foraging behavior of the honey bees in ref. (8). ABC algorithm is used to design active analog filters and its performance is compared with GA and PSO algorithms in ref. (9). The Genetic Algorithm (GA) is based on Darwian law of survival of fittest in ref.

(10). The particle swarm optimization (PSO) algorithm simulates the behavior of birds flocking in search of food in ref. (11).

The resting part of this paper follows the section ii design methods, section iii specifications, section iv optimization techniques, section v results and discussion, section vi conclusion and section vii references.

DESIGN METHODS

Two-stage operational amplifier consists a finding variable set$x = \{x1, x2, \ldots\ldots, xn\}$ that optimizes the performance functions such as offset voltage, gain, maximum operating frequency and signal to noise ratio. Vector x may encompass biases, length (L), widths (W), of the MOS transistors and component values.

Specifications

In this paper, two-stage operational amplifier is a fundamental circuit used for the design analog IC. Design variables are the size of the transistor (length and width). The value of the passive components is (capacitors and resistors), and the value of bias currents and bias voltages. For this particular two-stage operational amplifier, there are 14 design variables. Design specifications for two-stage operational amplifier are open loop DC gain (Av), unity gain bandwidth (GBW), phase margin (Pm), slew rate (SR), power consumption (P), and area (A).

Open loop Dc gain (Av): for two-stage operational amplifier, the open loop voltage gain is given by

$$\mathrm{Av} = \frac{\mathrm{gm1}}{\mathrm{gds2+gds4}} + \frac{\mathrm{gm6}}{\mathrm{gds7+gds6}} \ \ldots \qquad (1)$$

Av = open loop DC gain
gm = Transconductance
gds = Output conductance

Unity gain bandwidth (GBW):

$$\text{Gain bandwidth (GBW)} = \frac{\mathrm{gm1}}{\mathrm{Cc}} \ldots\ldots \qquad (2)$$

Cc = compensation capacitance.

Phase margin (Pm): The phase margin of a operational amplifier depends on the sum of phase shifts at the unity gain frequency. Contributed by the non-dominant poles (P1 and P2) and zeros (Z).

$$\mathrm{Pm} = \pm 180 - \tan^{-1}\left(\frac{\mathrm{GBW}}{\mathrm{P1}}\right) - \tan^{-1}\left(\frac{\mathrm{GBW}}{\mathrm{P2}}\right) - \tan^{-1}\left(\frac{\mathrm{GBW}}{\mathrm{Z}}\right) \ldots\ldots.. \quad (3)$$

Pm =Phase margin
GBW =Gain bandwidth
$P1$ =First pole
$P2$ =Second pole
Z = zero

Power consumption (P):

$$\mathrm{P} = (\mathrm{Vdd} - \mathrm{Vss}) + (\mathrm{I5} + 2\ \mathrm{I7}) \ldots\ldots \qquad (4)$$

Vdd = Supply voltage
Vss = Voltage source supply
P = Power consumption
$I5\ is$ Current flows through the $M5$ transistor.

Slew rate (Sr): slew rate for operational amplifier is

$$\mathrm{Sr} = \frac{\mathrm{I5}}{\mathrm{Cc}} \ldots\ldots \qquad (5)$$

4.6 Area (A): The area of the operational amplifier is given by the sum of the transistors and capacitor areas.

$$\text{Area (A)} = \sum_{i=1}^{k} \mathrm{Wi} * \mathrm{Li} \ldots\ldots \qquad (6)$$

OPTIMIZATION TECHNIQUES

Genetic Algorithm

In the early 1970s Jhon Holland introduced the concept of genetic algorithm and his aim was to make computers do what nature does, Holland was concerned with algorithms that manipulates string of binary digits. Each artificial “chromosome” consists of a number of “genes” and each gene is represented by 0 or 1
1011010000010101

Nature has an ability to adapt and learn without being told what to do. In other words, nature finds good chromosome blindly, genetic algorithm do the same. Two mechanisms link a GA to the problem, it solving ENCODING and EVALUTION.

The GA uses a measure of fitness of individual chromosomes to carry out reproduction. As reproduction takes place, the crossover operator exchanges parts pf two single chromosomes and the mutation operator changes the gene value in some randomly chosen location of the chromosome.

BASIC GENETIC ALGORITHMS

Represents the problem variable domain as a chromosome of a fixed length, choose the size of a chromosome population N, the crossover probability Pc and the mutation probability Pm. Define the fitness function to measure the performance, or fitness, of an individual chromosome in the problem domain. The fitness function establishes the basis for selecting chromosomes that will be mated during reproduction. Randomly generate the initial population of chromosome of size N:

x1, x2, ……., xN
Calculate the fitness of each individual chromosome:
f(x1), f(x2), ……., f(xN)

Select a pair of chromosomes for mating from the current population parent chromosomes are selected with a probability related to their fitness. Create a pair of offspring chromosomes by applying the genetic operators - CROSSOVER and MUTATION. Place the created offspring chromosomes in the new population. Repeat until the size of the new chromosome population becomes equal to the size of the initial population, N. Replace the initial (parent) chromosome population with the new (offspring) population. repeat the process until the termination criterion is satisfied.

Genetic Algorithms

GA represents an iterative process. Each iterative process called GENERATION. A typical number of generations for a simple GA can range from 50 to over 500. The entire set of generations is called RUN. Because GAs use a stochastic search method, the fitness of a population may remain stable for a number of generations before a superior chromosome appears. A common practice is to terminate a GA after a specified number of generations and then examine the best chromosome in the population. If no satisfactory solution is found, the GA is restarted.

Flow Chart for Genetic Algorithm

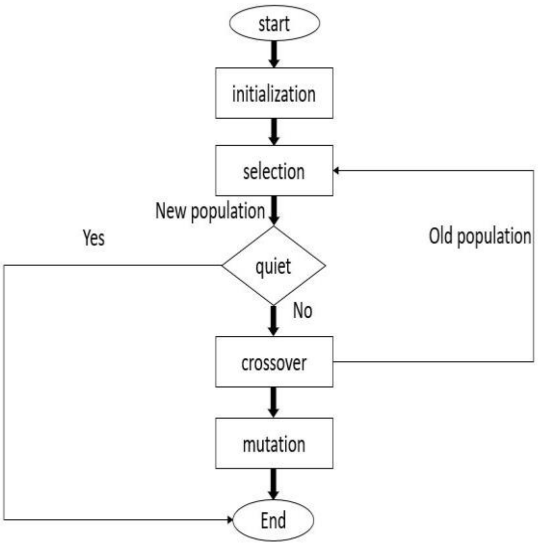

Figure-3. Flow chart for genetic algorithm.

Partical Swarm Optimization (PSO)

Particle swarm optimization (PSO) is an evolutionary computation technique developed by Kennedy and Eberhart. It exhibits common evolutionary computation attributes including initialization with a population of random solutions and searching for optima by updating generations.

A simulation of a simplified social system and original intent was to graphically simulate the graceful but unpredictable choreography of a bird flock. Each partical keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far.

PSO is initialized with a group of random particle solutions and then searches for optima by updating generations. Potential solutions, called particles, are then “flown” through the problem space by following the current optimum particles. Each particle keeps track of its coordinates in the problem space, which are associated with the best solution (fitness) it has achieved so far, this value is called “pbest”. Another “best” value that is tracked by the particle swarm optimizer is the best value obtained so far by any particle in the population and the second-best value is a global best and called as “gbest”. And the PSO concept consists of, at each step, changing the velocity (i.e. accelerating) of each partical toward its “pbest” and “gbest” locations (global versions of PSO).

Initialize the PSO parameter which are necessary for the algorithm. Population size which indicates the number of individuals. Number of generations necessary for the termination criterion. Cognitive constant, social constant, variation of the inertia weight, maximum velocity, number of design variables and respective ranges for the design variables. Generate random population equal to the population size specified. Each population member contains the value of all the design variables is randomly generated in between the design variable range specified Population means the group of birds (particles) which represents the set of solutions. Obtain the values of the objective function for the all the population members. For the first iteration, value of objective function indicates the pBest for the respective particle in the solution. Identify the particle with the best objective function value which identifies as gbest. If the problem is a constrained optimization problem, then a specific approach such as static penalty, dynamic penalty and adaptive penalty is used to convert the constrained optimization problem into the unconstrained optimization problem. Update the velocity of each particle and check for the maximum velocity. If the velocity obtained exceeds the maximum velocity, then reduce the existing velocity to the maximum velocity. Update the position of the particles and check all the design variables for the upper and lower limits. Obtain the value of the objective function for all the particles. The new solution replaces the pbest if it has better function value. Identify the gbest from the population. Update the value of inertia weight if it required. Best obtained results are saved using `elitism. All elite members are not modified using crossover and mutation operators but can be replaced if better solutions are obtained in any iteration. Repeat the steps until the specified number of generations or termination. The method easily suffers from the partial optimism, which causes the less exact at the regulation of its speed and the direction.

Flow Chart for PSO

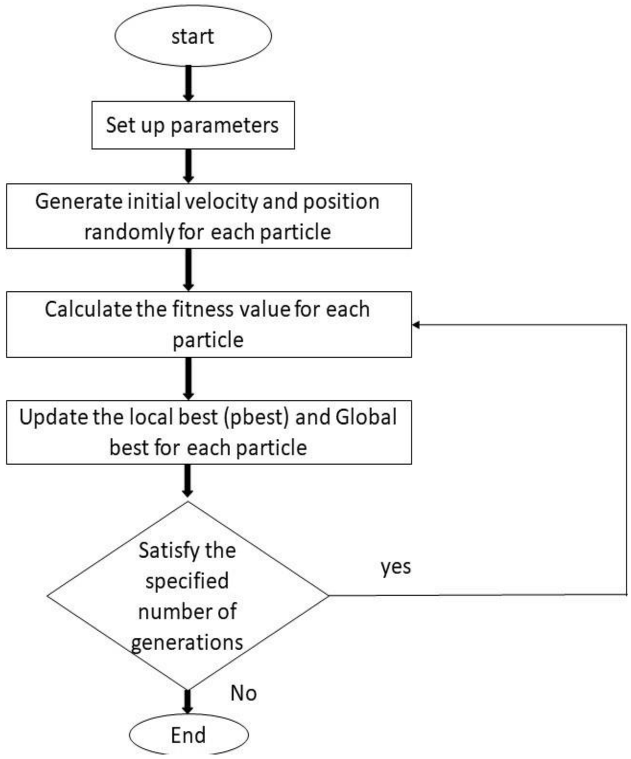

Figure-4. Flow chart for PSO algorithm.

Artificial Bee Colony (ABC)

It is a swarm intelligence algorithm. It was first proposed by DERVIS KARABOGA in 2005 and first publication in 2007, in the journal of global optimization. “a powerfull and efficent algorithm for numerical function optimization: abc algorithm”.

Swarm Intelligence

Any attempt to design algorithms or distributed problem-solving devices inspired by the collective behavior of social insect colonies and other animal societies.

Examples

- Bee swarming around their hive.
- Ant colony with ants as individual agents.
- Flocking of birds is a swarm of birds.
- Immune system is a swarm of cells.
- Crowd is a swarm of people.

Properties of Swarm Intelligence Behavior

There are two types of swarm intelligence behavior they are:

- Self-organization
- Division of labor

Self-Organization

Interactions are executed on the basis of purely local information without any relation to the global pattern. Positive feedback, negative feedback, fluctuations and multiple interactions.

5.8 DIVISION LABOR

Tasks performed simultaneously by specialized individuals.

Components of Honey Bee Swarms

- Food sources
- Employed foragers
- Unemployed foragers

Food Sources

Value depends on its proximity, richness, and the ease of extraction. It can be represented with a single quantity “profitability”.

Employed Forgares (BEES)

Currently exploiting a food-sources. It contains information on distance profitability and direction from the next. It shares the information with a certain probability. It takes nectar to the hive and unloads, abounded food-source becomes an uncommitted follower. It continues to forage at the food-source.

Unemployed Forages (BEES)

In these unemployed forages there are two types of bee are present they are:

- On looker bees
- Scout bees

On Looker Bees

In this case, watch the waggle dancers to become a recircuit and start searching for food sources.

Scout Bees

In this case, starts searching around the nest spontaneously.

ABC Consists of Three Phases

- Employed bee phase
- Onlooker bee phase
- Scout bee phase

Employed Bee Phase

Employed bee phase try to identify better food source then the one associated with it. Generate a new solution using a partner solution.

On Looker Bee Phase

It selects a food source with a probability related to nectar amount. Generate a new solution using partner solution.

Scout Bee Phase

Exhausted food sources are abandoned, it discards and generate new solution.

Greedy Selection

Accepting new solution if it is better than the current solution is called a greedy selection. In this ABC optimization fitness value is related to objective function. Fitness solution is evaluated as:

Fit = 1/1+f if f>=0
1+[f] if f<0

	O	F
S1	10	10
S2	5	5

Figure-5. Table for greedy selection.

If we are solving a minimization problem then solution S2 is better than S1. Fitness is directly corresponding to objective function.

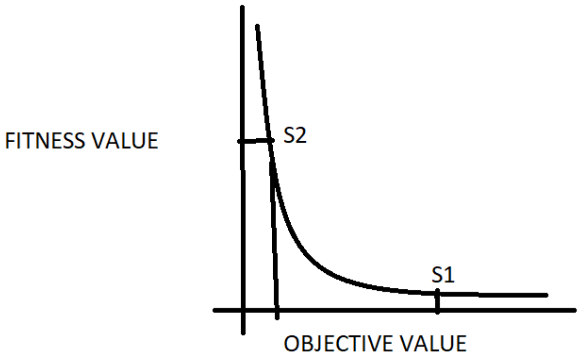

Figure-6. Graph between fitness value and objective value.

Taking two solutions, it shows the if the fitness value increases then the objective function value decreases. If we are solving a minimization problem, we have to take S2 as a better solution because it having a lower objective function value. Since we are working with the fitness value which is inversely related, we are supposed to take a solution which as maximum fitness value. For taking two solutions we have to find a greedy selection it would to take a higher fitness value because in the ABC algorithm fitness value is related to objective function value. Where as in another algorithm fitness value should take a lower fitness value, because the fitness value is directly corresponded to the objective function. If I solving a minimization problem, I will take a higher fitness value for greedy selection. Greedy selection to update the solutions.

X = current solution
X new = newly generated solution
F = objective function value of a solution
F new = objective function value of new solution
Fit = fitness of a solution
Fit new = fitness of a new solution

X= X new
f= f new
X and f remain the same.
If Fit new < fit

Flow Chart for Artificial Bee Colony Algorithm

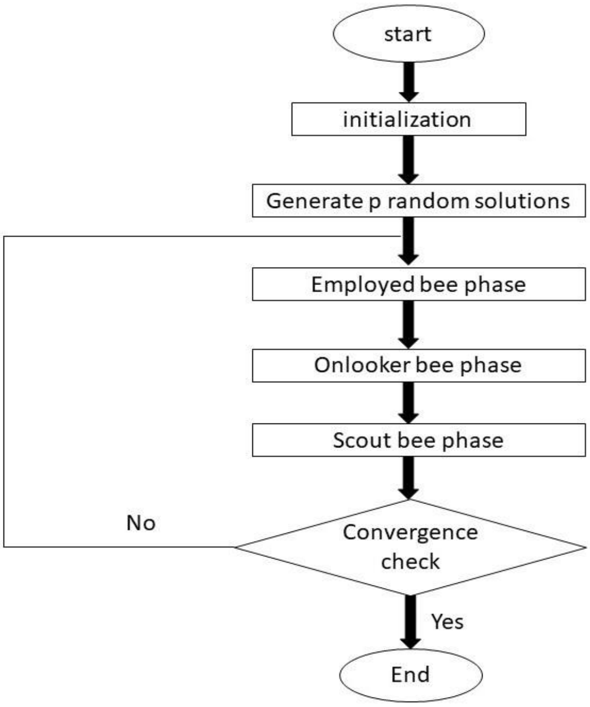

Figure-7. Flow chart for ABC algorithm.

RESULTS AND DISCUSSIONS

Schematic and Simulation Results of Two Stage Operational Amplifier

A two-stage operational amplifier is implemented in 180nm CMOS process technology. For designing the op-amp first we have implemented the MATLAB code and by running the code we achieved the W/L values. Simultaneously, the same process is repeated for the evolutionary algorithms. With the help of these values, we achieved the gain as 80.32dB. The power consumption achieved is 144.311µW under the 1V supply voltage. The simulation results and outputs are,

Equation based Two Stage OP-AMP

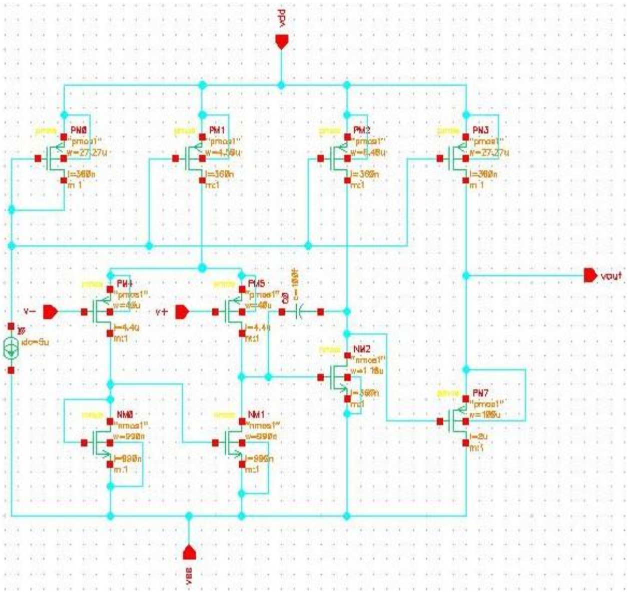

Figure-8. Schematic of a two stage OP-AMP.

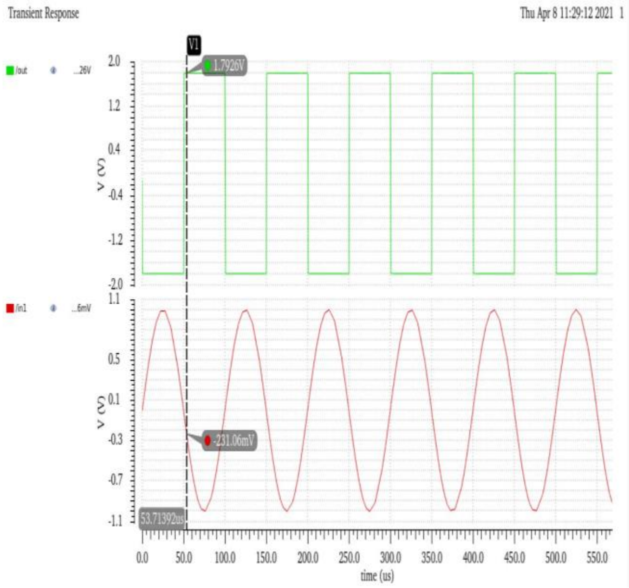

Figure-9. Transient analysis of a two-stage op-amp.

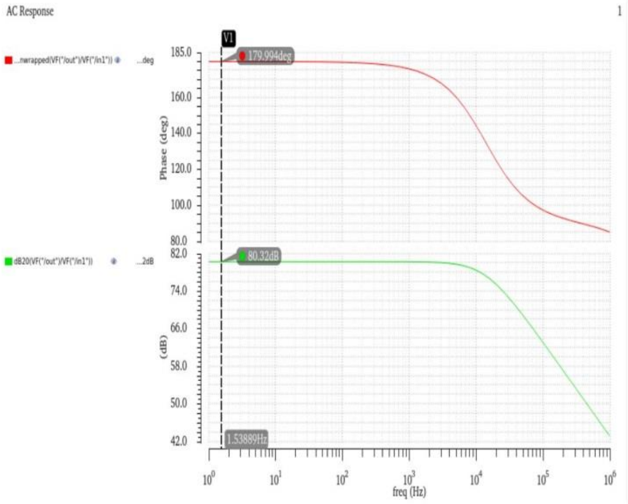

Figure-10. AC response of an op-amp
GAIN of an OP-AMP =80.32dB

Simulation Results of Two-stage OP-AMP by Using Genetic Algorithm

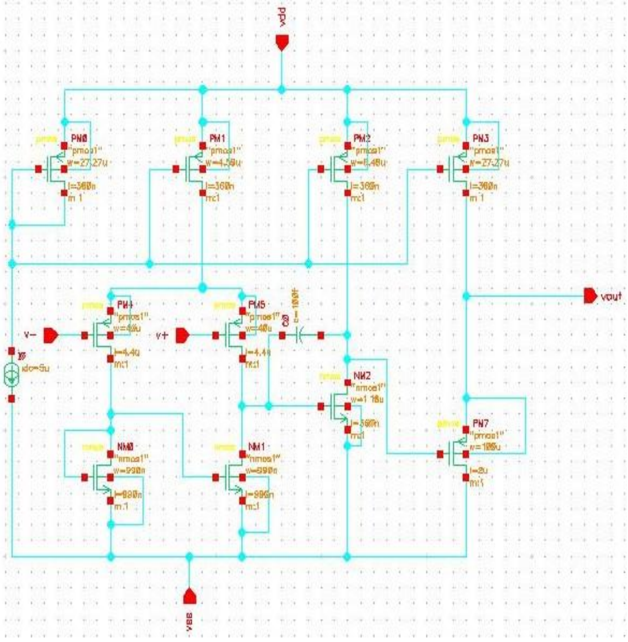

Figure-11. Schematic of two-stage op-amp by using genetic algorithm.

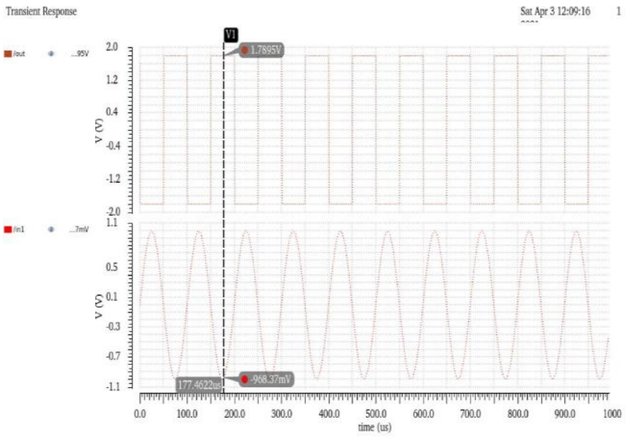

Figure-12. Transient response of two-stage op-amp by using genetic algorithm.

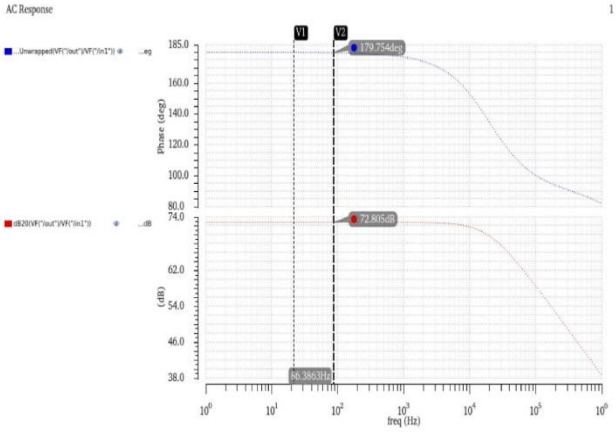

Figure-13. AC response of two-stage op-amp by using genetic algorithm GAIN =72.80dB

- **Simulation results of two-stage op-amp by using MPSO algorithm:**

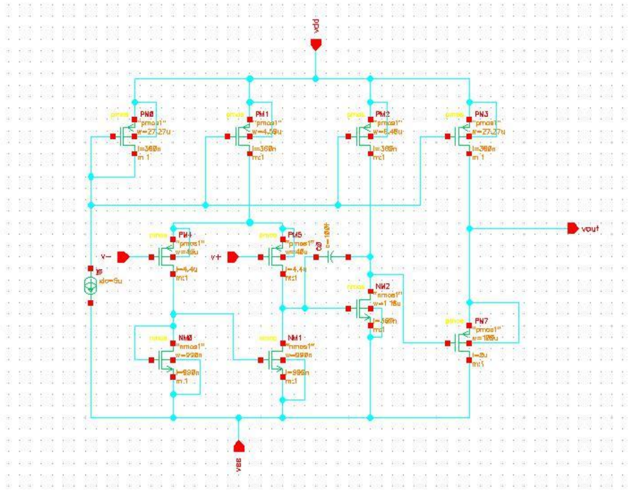

Figure-14. Schematic of two-stage op-amp by using MPSO algorithm.

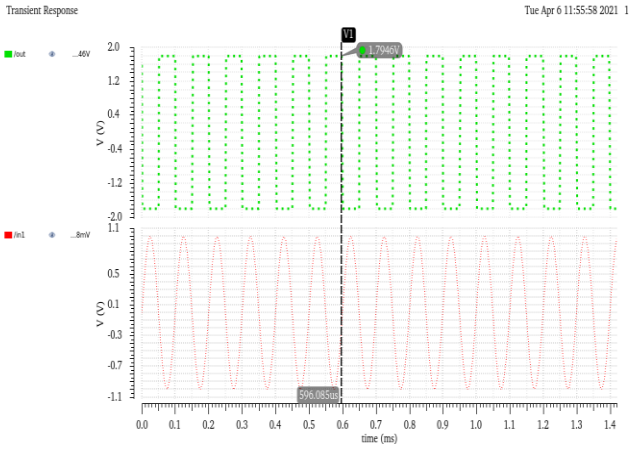

Figure-15. Transient response of two-stage op-amp by using MPSO algorithm.

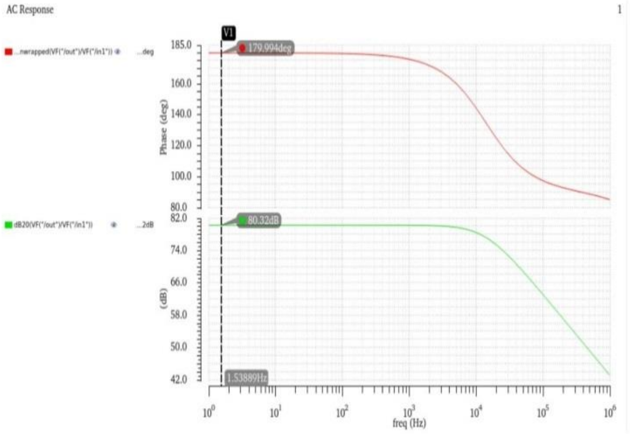

Figure-16. AC response of two-stage op-amp by using MPSO algorithm GAIN =80.32dB

- **Simulation results of two-stage op-amp by using ABC algorithm:**

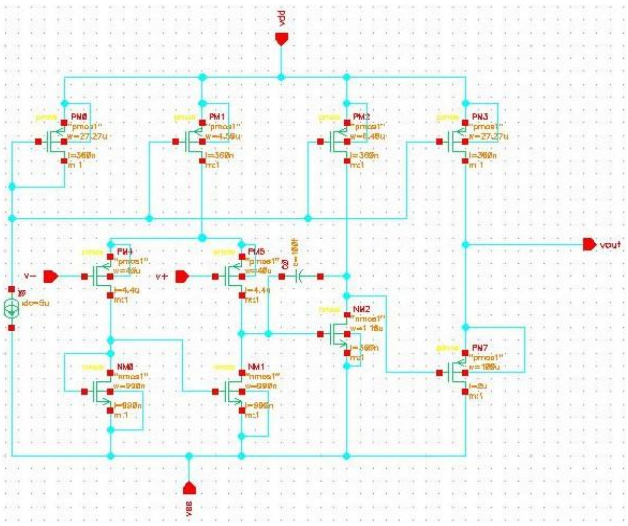

Figure-17. Schematic of two-stage op-amp by using ABC algorithm.

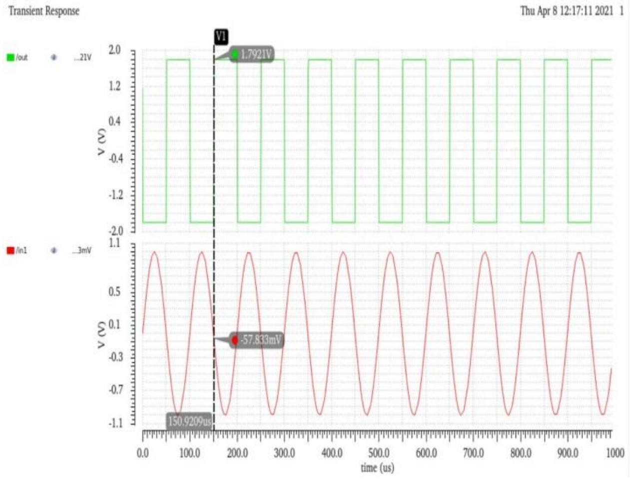

Figure-18. Transient response of two-stage op-amp by using ABC algorithm.

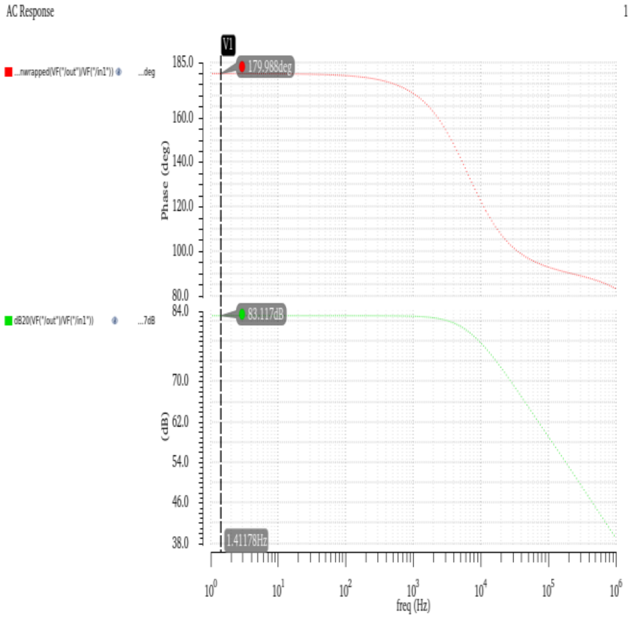

Figure-19. AC response of two-stage op-amp by using ABC algorithm GAIN =83.117dB

Comparison:

Parameters	Equation based two-stage op-amp	Genetic op-amp	Modified PSO	ABC
DC gain(dB)	76.32	72.80	80.32	83.117
Phase Margin(^{0})	58	56	52	63
Slew rate(V/µS)	34.152	32.5196	40.0153	41.024
Power(µW)	147.3119	198.023	144.401	142.204

CONCLUSIONS

In this paper, two-stage CMOS operational amplifier circuit are optimally designed by using ABC algorithm. ABC are able to find out optimal design parameters for the designed circuit. The circuits are re-designed in the MATLAB environment by using ABC algorithm-based design variables. Simulation results that establish that evolutionary technique-based algorithms will meet all the specifications and also minimizes the total MOS transistor area. ABC provides better performances in terms of gain and power consumption when compared to other methods.

REFERENCES

[1] M. Hershenson, S. P. Boyd and T. H. Lee. 2001. Optimal design of a CMOS op-amp via geometric programming. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 20: 1-21.

[2] P. P. Kumar and K. Duraiswamy. 2012. An optimized device sizing of analog circuits using particle swarm optimization. J. Comput. Sci. 8: 930-935.

[3] S. Kamisetty, J. Garg, J. N. Tripathi and J. Mukherjee. Optimization of analog RF circuit parameters using randomness in particle swarm optimization, Proc. IEEE World Congress on Information and Communication Technologies (WICT'11). pp. 274-278.

[4] V. Ceperic, Z. Butkovic, A. Baric. 2006. Design and optimization of self-biased complementary folded cascode. In Proceedings of the IEEE Mediterranean Electrotechnical Conference (MELECON). pp. 145-148.

[5] B. P. De, R. Kar, D. Mandal, S. P. Ghoshal. 2015. Optimal selection of components value for analog active filter design using simplex particle swarm optimization. Int. J. Mach. Learn. Cybern. 6(4): 621-636.

[6] R. A. Vural, T. Yildirim. 2012. Analog Circuit Sizing via Swarm Intelligence. AEU International Journal of Electronics and Communications. 66(9): 732-740.

[7] M. Dorigo and C. Blum. 2005. Ant colony optimization theory: A survey. Theoretical computer science. 344(2): 243-278.

[8] D. Karaboga and B. Basturk. 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. Journal of global optimization. 39(3): 459-471.

[9] R. A. Vural, T. Yildirim, T. Kadioglu, and A. Basargan. 2012. Performance evaluation of evolutionary algorithms for optimal filter design. Evolutionary Computation, IEEE Transactions on. 16(1): 135-147.

[10] S.Sivanandamand S. Deepa, Introduction togenetic algorithms. Springer Science & Business Media, 2007.

[11] J. Kennedy and R. Eberhart. 1995. Particle swarm optimization. In Neural Networks, 1995. Proceedings. IEEE International Conference on. 4: 1942-1948.