NEW APPROACHES TO OPTIMIZATION ALGORITHMS USED IN CIRCUIT AND SYSTEM LEVEL OF ANALOG DESIGN AUTOMATION

by

Simge AY

B.S., Electronics Engineering, Boğaziçi University, 2010

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering
Boğaziçi University
2012

NEW APPROACHES TO OPTIMIZATION ALGORITHMS USED IN CIRCUIT AND SYSTEM LEVEL OF ANALOG DESIGN AUTOMATION

APPROVED BY:

Prof. Günhan Dündar .....
(Thesis Supervisor)

Prof. H. Levent Akın .....

Assist. Prof. İ. Faik Başkaya .....

DATE OF APPROVAL: 13.09.2012

ACKNOWLEDGEMENTS

I would like to express my profound sense of gratitude to my thesis supervisor Prof. Günhan Dündar for his continuous support, patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I would like to thank Assist. Prof. I. Faik Başkaya and Prof. H. Levent Akin for sharing their knowledge and taking part in my thesis jury.

It is my pleasure to thank my colleagues in both school and office, especially Selin Tolunay, İsmail Kara, Ali Murat Gök, Can Doğa Kırbaç, Berk Omuz, Fehime Bıyıklıoğlu, Mehtap Karamanlı and Mehmet Ali Etişkol for their support, friendship, and inspiring ideas. I am forever grateful to my friends Betül Küçükakarsu Usta, Aylin Aydoğdu, Gamze Güzel and Yigit Demir for their constant support and encouragement throughout my research work.

Finally, I take this opportunity to express the profound gratitude from my deep heart to my beloved parents and my sibling for their love and continuous support both spiritually and materially.

ABSTRACT

NEW APPROACHES TO OPTIMIZATION ALGORITHMS USED IN CIRCUIT AND SYSTEM LEVEL OF ANALOG DESIGN AUTOMATION

This thesis presents different simulation-based analog circuit synthesis methodologies, and synthesis examples that were performed to validate the usefulness of the methodologies. It also presents the integration of circuit synthesizer with system level. Simulation-based approach is preferred so that the synthesizer, SACSES, is implemented with HSPICE. Two different circuit synthesis methodologies that manipulate device size indirectly by using modified ES algorithm are proposed. The first one is based on the inversion coefficient (IC) as a key design parameter and the equation of the EKV MOSFET Model. The second one is based on DC operating points and lookup-tables used for translating DC operating points into transistor dimensions. A hierarchical synthesis structure is proposed for integrating SACSES with system level synthesis. The hierarchical scheme eliminates the need for extra tools to link levels.

ÖZET

ANALOG TASARIM OTOMASYONUNUN DEVRE VE SISTEM SEVİYELERİNDE KULLANILAN OPTIMIZASYON ALGORİTMALARINA YENİ YAKLAŞIMLAR

Bu tezde, benzetimli devre sentezi için benzetim-tabanlı yöntem dizileri ve bu yöntem dizilerinin faydalılığını onaylamak amacıyla gerçekleştirilmiş olan sentez sonuçları sunulmaktadır. Ayrıca, devre sentezleyicinin sistem seviyesi ile birleştirilmesi de sunulmaktadır. Benzetim-tabanlı yaklaşım tercih edilmiş sonucunda sentezleyici, SACSES, HSPICE ile gerçekleştirilmiştir. Transistor boyutlarını ES algoritması kullanarak dolaylı olarak değiştiren iki farklı devre sentez yöntemi önerilmiştir. İlk yöntem tasarım parametresi olarak evirici katsayıya ve EKV MOSFET Model'in denklemlerine dayanır. İkinci yöntem doğru akım çalışma noktalarına ve bu noktaları transistor boyutlarına çeviren başvuru çizelgelerine dayanır. SACSES'i sistem düzeyindeki sentez araçlarıyla birleştirmek için sıradüzenli bir sentez yapısı önerilmiştir. Sıradüzenli yapı, devre ve sentez düzeylerini birleştirmek için ek araçlara olan gereksinimi ortadan kaldırmaktadır.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZET v
LIST OF FIGURES vii
LIST OF TABLES ix
LIST OF SYMBOLS x
LIST OF ACRONYMS/ABBREVIATIONS xi
1. INTRODUCTION 1
1.1. Overview of Analog Design Automation 1
1.2. Overview of Hierarchical Genetic Algorithms 3
1.3. Main Contributions and Outline 4
2. BACKGROUND 5
3. A SINGLE OBJECTIVE OPTIMIZATION APPROACH FOR ANALOG INTEGRATED CIRCUITS 8
3.1. ES Algorithm 9
3.2. Implementation of SACSES in MATLAB 12
3.3. Circuit Synthesis Examples 15
4. OPTIMIZATION OF VOLTAGES AND INVERSION COEFFICIENT . . . 19
4.1. IC Based Approach to Circuit Level Design Automation 19
4.2. Voltage Based Approach to Circuit Level Design Automation 21
4.3. Circuit Synthesis Example for Voltage and IC Based Approach 23
5. HIERARCHICAL OPTIMIZATION OF ANALOG INTEGRATED CIRCUITS 31
5.1. Cost Function of the System Level 34
5.2. HGA Alternative 1 and HGA Alternative 2 35
5.3. HGA Alternative 3 37
5.4. Implementation of Hierarchical Genetic Algorithms 40
6. CONCLUSION 44
REFERENCES 46

LIST OF FIGURES

Figure 1.1. Analog design flow. 2

Figure 3.1. ES algorithm 10

Figure 3.2. Circuit synthesis flow diagram. 13

Figure 3.3. WL formulation for sizing. 15

Figure 3.4. The circuit schematics of BTS. 16

Figure 3.5. The circuit schematics of FC OPAMP. 18

Figure 4.1. IC formulation for sizing. 21

Figure 4.2. Voltage formulation for sizing. 22

Figure 4.3. BTS OPAMP circuit schematic with node voltages. 24

Figure 4.4. Cost function of IC optimization. 26

Figure 4.5. Cost function of V optimization. 26

Figure 4.6. Cost function of WL optimization. 27

Figure 4.7. V5 and V12 of all optimizations. 28

Figure 4.8. V6 and V7 of all optimizations. 29

Figure 4.9. Gain and rout of all optimizations. 30

Figure 5.1. 3rd order low pass Butterworth filter with Sallen-Key topology. . . 32

Figure 5.2. Non ideal OPAMP model used in the synthesis example. 33

Figure 5.3. HGA alternative 1 and 2 flow diagram. 36

Figure 5.4. HGA alternative 3 flow diagram. 39

Figure 5.5. Cost function of HGA alternative 1. 40

Figure 5.6. Cost function of HGA alternative 2. 40

Figure 5.7. Cost function of HGA alternative 3. 41

Figure 5.8. Bandwidth of HGA alternative 3. 41

Figure 5.9. Gain of HGA alternative 3. 42

Figure 5.10. Rout of HGA alternative 3. 42

LIST OF TABLES

Table 3.1. Specifications of the synthesized BTS OPAMP. 17

Table 3.2. Specifications of the synthesized FC OPAMP. 17

Table 4.1. An example of the look-up table corresponding to Vgs=0.4V. . . . 23

Table 4.2. Comparison of the synthesized BTS OPAMP for WL, IC and V optimization. 29

Table 5.1. Synthesis results for integration with three different approaches. . . 43

LIST OF SYMBOLS

V_{ds}	Drain to source voltage
V_{gs}	Gate to source voltage
V_{th}	Threshold voltage
V_{tn}	Threshold voltage of NMOS
V_{tp}	Threshold voltage of PMOS
T	Absolute temperature
α	Cooling rate
λ	The number of offsprings
μ	The number of parents

LIST OF ACRONYMS/ABBREVIATIONS

BTS	Basic Two Stage
CMOS	Complementary Metal Oxide Semiconductor
CMRR	Common Mode Rejection Ratio
DC	Direct Current
GA	Genetic Algorithm
EA	Evolutionary Algorithm
EKV	Enz-Krummenacher-Vittoz
ES	Evolutionary Strategies
HGA	Hierarchical Genetic Algorithm
IC	Inversion Coefficient
MOSFET	Metal Oxide Semiconductor Field-Effect Transistor
NMOS	N Type Metal Oxide Semiconductor
OPAMP	Operational Amplifier
PM	Phase Margin
PMOS	P Type Metal Oxide Semiconductor
SA	Simulated Annealing
SACSES	Simulation-Based Analog Circuit Synthesizer using Evolutionary Strategies
SOC	System on Chip
SR	Slew Rate
V	Voltage
WL	Width-Length

1. INTRODUCTION

1.1. Overview of Analog Design Automation

With the advance in semiconductor technologies and the rapid growth of the markets for computer, communication, and consumer electronics, more and more analog and mixed-signal circuits are integrated with digital units to realize systems-on-chip (SOC). Actually, comprehensive analysis and complicated trade-offs among various aspects of performances are required for designing analog circuits. Especially, in advanced process technologies, numerous high-order non-idealities should be taken into account which further complicates the analysis and design processes. As a result, it usually requires a large portion of development time to design analog functional blocks when implementing a SOC. How to shorten the development time of analog circuits, and consequently shorten the time to market, is a pressing subject for a designer.

The design of analog circuits starts with a high-level behavioural description of the desired circuit. Because of the complexity in analog design, it has not been automated like digital design to a great extent. Analog design can be decomposed into three levels which are system, circuit and layout levels, as shown in Figure 1.1.

The system level design process is responsible for converting high level specifications to lower level block specifications. For this process, system level must have some knowledge about building blocks in order to converge to a feasible solution. This knowledge can be obtained by a performance estimator or by communicating with the circuit level synthesizer.

The circuit level design process is responsible for adjusting the circuit parameters for a design specification, known as the circuit sizing. Circuit sizing is at the center of the analog design and is responsible for designing the circuit by adjusting the transistor widths and lengths, reference currents and voltages, resistors and capacitors. Circuit sizing process should be independent of topology and fabrication process. When an in-

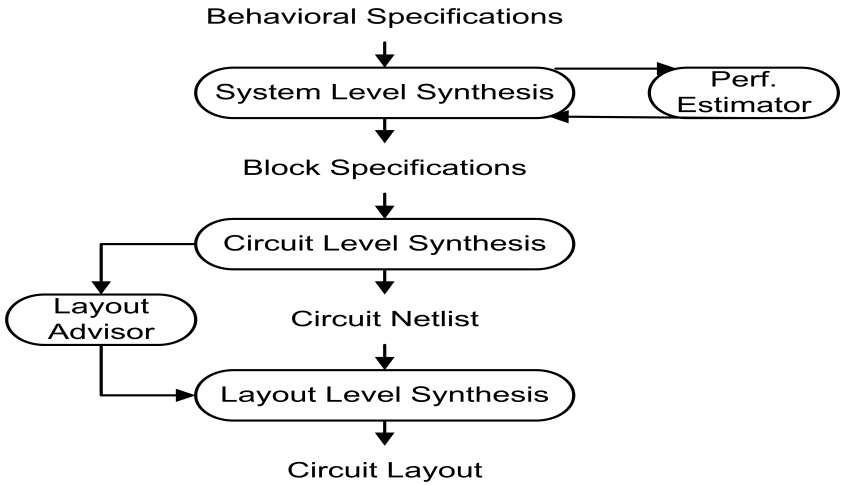

Figure 1.1. Analog design flow.

tegrated circuit is fabricated the performance may be different from simulation results. This may occur due to several reasons, especially global process variations and the mismatch. Normally process parameters are assumed to be similar for all the devices on the same wafer, but may vary from wafer to wafer. Mismatch is caused by the local changes on the same chip. The analog circuits rely on the close matching of a set of devices, and the changes will degrade the performance of the circuit. Because of process variations and the mismatch, the circuit may fail to meet the desired specifications. Furthermore, because circuit properties are correlated with each other, adjusting one parameter may change many performances, which requires simultaneous optimization between all specifications. Therefore, manual circuit sizing is a difficult and time-consuming process. In order to shorten the design time, analog design automation becomes essential.

In the layout level, the sized schematic is converted into a physical representation which is ready for manufacturing. Then, physical verification is done through Design Rule Check (DRC) and the Layout Versus Schematic (LVS) check. Finally, in the

extraction, parasitics elements like resistances and capacitances are extracted from the layout.

1.2. Overview of Hierarchical Genetic Algorithms

Genetic algorithms (GAs) have become increasingly important to solving difficult problems because they can provide feasible solutions. However, in some applications, especially in large scale problems, adding specific improvements and tactics is required in genetic algorithms when the time consumption is considerable. A type of a genetic algorithm has started to be used which is called Hierarchical Genetic Algorithm (HGA). For example, it is used in silhouette matching in gait recognition approach [1], T–S fuzzy systems [2], and a fuzzy supervisory PI controller [3].The flexibility and modularity of HGA provides that large-scale problems are divided into sub-problems by using parallel processed Genetic Algorithms. In this way, the efficiency is increased and total process time is diminished.

HGA may have two layers, top layer and the low layer or may be multi-layered: one top level and several lower layers according to the structure of the application. The top layer or higher sub-populations generally search a large space with lower resolution, opposite to this lower-layer or lower levels search smaller space with higher resolution. Communications among the populations located different layers are provided by migration of individuals with different strategies. In the design of the HGA, the structure of the hierarchy and topology strategies like individual migrations, coordination among the top layer and bottom layers is important. Constructing an efficient coordination and load sharing in HGA allows us to accelerate the convergence speed of the algorithm to the optimum, and to diminish the total process time.

As a consequence, using Hierarchical Genetic Algorithm with different strategies and models is required for solving complex problems with the same quality in GA but faster than GA.

1.3. Main Contributions and Outline

This thesis has three main objectives. The first one is to integrate SACSES [4] with an existing simulator, HSPICE. The second objective is to develop simulation-based analog circuit synthesis methodologies that manipulate device size indirectly by using ES algorithm. The third objective is to develop an infrastructure to communicate with system level and circuit level using HGA. Main contributions of this thesis are listed below.

- SACSES is integrated with HSPICE because of accurate circuit simulation and successful design tape-outs.

- In order to shorten the synthesis time and the search space of ES algorithm, IC optimization based on EKV MOSFET model is developed.

- Voltage optimization based on bias voltages and currents is developed. It provides that the search space of ES algorithm can be restricted by the circuit constraints. Also, it guarantees that all transistors are operated in saturation region.

- A new synthesis methodology, hierarchical synthesis, that combines successive levels of abstraction, is developed together with the study of [5] and [4]. The method is applied to the system and circuit levels and the need for a performance estimator is eliminated.

The thesis is organized such that the implementation of SACSES in MATLAB and the integration of it with HSPICE are given in detail in Chapter 2. Chapter 3 gives two different approaches about circuit optimization. HGA alternatives used in the communication between circuit and system level are explained in Chapter 4. The final chapter states the conclusions.

2. BACKGROUND

Analog circuit synthesis is essential in various levels at analogue and digital design processes by adjusting transistor sizing, calculating the passive component values, and adjusting bias voltages and currents. In addition to this, it is a difficult and very time consuming process. In order to shorten the design time, analog design automation has become a hot search topic in recent years. In the literature, analog design automation approaches can be roughly classified into three categories: knowledge-based approaches, simulation-based approaches and equation-based approaches.

In knowledge based analogue synthesis computer programs; OASYS [6], BLADES [7] and IDAC [8], optimization depends on designers' background knowledge and intuitions. Knowledge-based generalizes designers' experience and obtains a design solution based on some pre-defined design flows and database. However, the results of these knowledge based approaches are inaccurate.

In addition to, equation-based analogue synthesis techniques have also been used. Some examples of these approaches are OPASYN [9], OPTIMAN [10] and AMGIE [11]. These techniques are quite fast due to using analytical equations for circuit evaluation, if the terms in transfer function are not complex. As the equations get more complicated, this model loses its efficiency. In order to improve efficiency while searching the optimality, the approaches based on geometric programming (GP) [12, 13] are getting popular in last decade. The optimal solution can be obtained by solving equations using mathematical solvers with the posynomial models of circuit properties and transistor parameters. Posynomial models have been developed for CMOS (Complementary Metal Oxide Semiconductor) OPAMPs [12, 13], multi-stage amplifiers [14] and oscillators [15]. However, large prediction errors often occur in some transistor parameters because of the short channel effect of CMOS technologies. In order to diminish errors, the this approach requires several iteration. In addition to this, the time required for model development and topology dependence limit the usefulness of this approach.

In simulation-based approaches, the circuit performances are evaluated via s SPICE-class simulator. Most of the simulation-based tools use commercially available. For example, the tool presented in [16] is integrated with the CADENCE Design Framework, GBOPCAD [17] uses HSPICE, [18] utilizes Eldo and its Levenberg-Marquardt algorithm-based optimizer. There are also some tools like MAELSTROM [19] and [20] that have simulator-encapsulating mechanisms in order to be more generic. Writing simulator requires extra work and time and the possibility of programming errors for written simulators is higher than commercial simulators. In this thesis, a commercial simulator, HSPICE is used because of accurate circuit simulation and successful design tape-outs. It is the industry's trusted and comprehensive circuit simulator.

The simulation based approach needs a SPICE netlist including all transistor dimensions and device values. The general solution for the sizing problem is adopting circuit simulators in the sizing iteration until finding the relatively better design [4,20,21]. Conventionally, the design variables of the sizing process are transistor dimensions W and L [4]. However, the free space of the design variables is huge because the dimension range is very large in a given process. Another useful set of design variables is DC operating point of the circuit: bias voltages and bias currents [10, 21]. In [10] All transistor dimensions are calculated from DC operating point using a general transistor model which may be first-order or more sophisticated model. In [21], transistor dimensions are obtained from DC operating point through look-up tables.

Another part of a circuit synthesizer is the search algorithm. A synthesizer, on the other hand, starts from a randomly sized circuit and needs global search methods. Among various global search techniques, simulated annealing (SA) and evolutionary algorithms (EA) like genetic algorithms (GA) and evolutionary strategies (ES), have been widely used for the synthesis of analog circuits. In [21], SA is adopted to speed up the convergence of the sizing iteration. In DARWIN [22], genetic algorithm (GA) is used to perform simultaneous topology selection and circuit sizing. In [23], evolutionary algorithms are applied for the design of arithmetic circuits. In this thesis, the combination of ES and SA algorithms [5, 24–26] is used. In the selection step of ES, the Boltzmann selection mechanism of SA is used [4]. In this way, the scope of the

search is adjusted during synthesis

Genetic algorithms are used hierarchical structure in order to solve complex problems and decrease the total process time. HGA can address different hierarchical aspects, including the use of a fitness-based hierarchy of populations [27], problem-specific subdivision of an algorithm into multiple levels [28], and the use of a hierarchical representation by using control genes that regulate other genes [29].

3. A SINGLE OBJECTIVE OPTIMIZATION APPROACH FOR ANALOG INTEGRATED CIRCUITS

Many real-world decision making problems need to achieve several objectives: minimize risks, maximize reliability, minimize deviations from desired levels, minimize cost, etc. The main goal of single-objective optimization is to find the best solution, which corresponds to the minimum or maximum value of a single objective function that lumps all different objectives into one. In this thesis, the simulation-based analog synthesis tool, SACSES [4] is implemented in MATLAB. In SACSES, simulation based analog integrated circuits synthesis is translated into single-objective optimization to obtain desired gain, bandwidth, phase, power, area, etc. We gather all of these objectives into a single cost function:

$$C = \mathbf{C}_{perf} + \mathbf{w}_{penalty}\mathbf{C}_{penalty} \tag{3.1}$$

$\mathbf{C}_{penalty}$ is calculated according to the operation point of all transistors. It is proportional to each transistor distance from the constraint-satisfying point. The performance related part, $\mathbf{C}_{perf}$ is a weighted sum square of normalized values of objectives:

$$\mathbf{C}_{perf} = \sum_{i=1}^{n} \mathbf{w}_i \hat{\mathbf{f}}_{\mathbf{i}}^2 \tag{3.2}$$

where n is the number of performance specifications. With the presence of the weight coefficient of every sub-objective, this function is not considered as a simple single objective function rather a kind of multi objective function.

As a method of simulation-based circuit level design automation, global search methods like genetic algorithms (GA), simulated annealing (SA), and evolutionary strategies (ES) are widely used because of their simplicity and ability to converge more quickly in practice. The circuit synthesizer presented in this thesis uses ES for the optimization algorithm because ES algorithm tries to avoid get stuck at a local minimum due to its selection mechanism.

This chapter starts by explaining the general structure of the ES algorithm. After mentioned operations of ES algorithm, the implementation of circuit synthesis in MATLAB is described. The chapter ends with the presentation of synthesis examples for BTS OPAMP and Folded Cascode (FC) OPAMP.

3.1. ES Algorithm

ES is a global search algorithm that has self-adaptation capability for the optimization parameters. It does not get stuck at a local minimum because it uses the Metropolis criterion of SA as the selection mechanism.

The algorithm is represented as (μ+λ) ES. This means that μ is the number of parents and λ is the number of offsprings. The outline of the algorithm is given in Figure 3.1.

Each individual chromosome is composed of circuit variables such as transistor widths, lengths, resistor, capacitor, inductor, biasing current and voltage values, and strategy parameters such as mutation step size, crossover coefficient, and cost function weights. The ES algorithm can be summarized by the following steps:

- In initialization μ individuals are generated randomly according to the given variable ranges.

- In recombination, two parents are randomly selected to generate two offsprings by crossing over using the recombination coefficient.

$g \leftarrow 0$

$P_\mu \leftarrow P_{\mu 0}$

while *convergence not reached* **do**

  for $i = 1$ to $\lambda/2$ **do**

    $[I_{parent1}\ I_{parent2}] \leftarrow choose + (P_\mu, 2)$

    $[I_{\mu+i}\ I_{\mu+i+1}]_s \leftarrow recombine_s(I_{parent1},\ I_{parent2})$

    $[I_{\mu+i}\ I_{\mu+i+1}]_x \leftarrow recombine_x(I_{parent1},\ I_{parent2})$

  end for

  for $i = 1$ to $\mu + \lambda$ **do**

    if I_i *is selected for mutation* **then**

      $I_{is} \leftarrow mutate_s(I_i)$

      $I_{ix} \leftarrow mutate_x(I_i)$

    end if

    $I_{icost} \leftarrow evaluate(I_i)$

  end for

  $P_\mu^{g+1} \leftarrow select(P_{\mu+\lambda}^g, \mu)$

  $g \leftarrow g + 1$

  $T \leftarrow updatetemperature()$

end while

$output \leftarrow bestsolution$

Figure 3.1. ES algorithm.

- In the mutation, each individual has its own standard deviation for each search variable and strategy parameters. During mutation, if an individual is selected for mutation, these standard deviations are first updated, then strategy parameters and search variables are updated.

- In cost function evaluation, performance related cost, $\mathbf{C}_{perf}$, and constraint related cost, $\mathbf{C}_{penalty}$, are evaluated because the cost function is composed of them as seen in the Equation 3.1. The performance related part, $\mathbf{C}_{perf}$, is the weighted sum of squared normalized distances from the target point seen in the Equation 3.2. In this equation, n is the number of performance specifications and fi the normalized values calculated:

$$\hat{\mathbf{f}}_i = \frac{\mathbf{g}_i - \mathbf{f}_i}{\mathbf{g}_i - \mathbf{b}_i}, \hat{\mathbf{f}}_{i,min} = 0 \tag{3.3}$$

where $\mathbf{g}_i$, good limit and $\mathbf{b}_i$, and bad limit of the performance specifications. The constraint related part is calculated according to the distance of each transistor from the constraint satisfying point:

$$p_{cut-off} = \sum_{i=1}^{m} p_{cut_i} = \sum_{i=1}^{m} \frac{k_{th}V_{th_i} - V_{gs_i}}{V_{th_i}}, p_{cut_i,min} = 0 \tag{3.4}$$

$$p_{triode} = \sum_{i=1}^{m} p_{triode_i} = \sum_{i=1}^{m} \frac{k_{sat}V_{dsat_i} - V_{ds_i}}{V_{dsat_i}}, p_{triode_i,min} = 0 \tag{3.5}$$

$$C_{penalty} = p_{cut-off} + p_{triode} \tag{3.6}$$

where m is the number of biasing constraints and the factors k_{th} and k_{sat} are utilized to introduce a safety margin for biasing. In this study, both k_{th} and k_{sat} are equal to unity.

- In selection, Metropolis criterion is used in a different manner. When selecting parents of the next generation among the current population, a randomly chosen individual competes with a pseudo individual having the population average cost $\mathbf{C}_{av}$ and wins with probability:

$$p(\mathbf{I}_i) = 1/e^{\mathbf{C}_i - \mathbf{C}_{av}/T} \tag{3.7}$$

and the process is repeated until the number of individuals wining against average cost is equal the number of parents. In this selection mechanism, population temperature is introduced. Population temperature can be viewed as a measure for the maturity of the evolution process, with lower temperature meaning later stages of evolution.

3.2. Implementation of SACSES in MATLAB

With the increase in computer performances, optimization-based synthesis has become the more preferred alternative which transforms the problem into a function minimization problem and exploits search algorithms. In this thesis, SACSES, an optimization-based synthesis using ES as a search algorithm, is implemented in MATLAB and cost function evaluation is made with the application of HSPICE. Flow diagram of the concept is shown in Figure 3.2.

While reading circuit topology and specifications, the number of circuit specifications and good and bad limits for them are set in MATLAB. Moreover, the netlist file name of the circuit run with HSPICE is set in MATLAB.

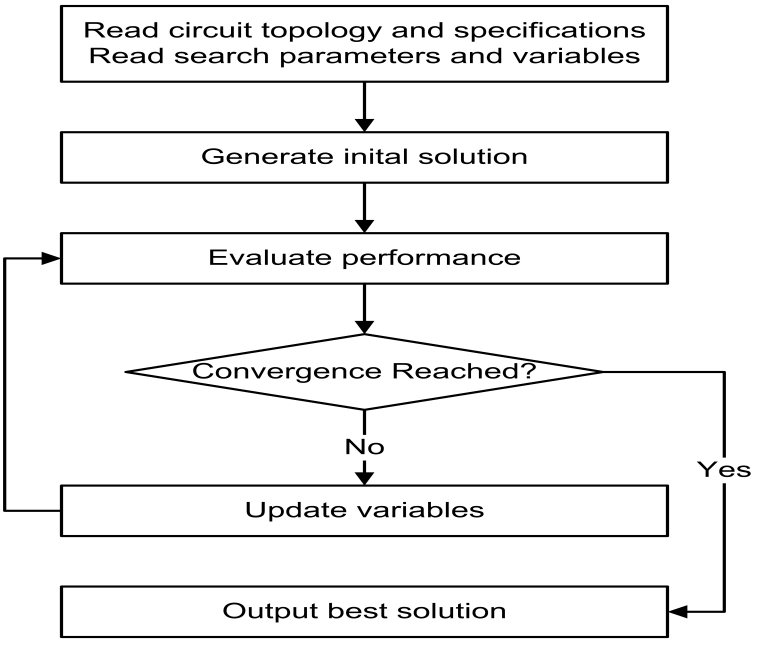

Figure 3.2. Circuit synthesis flow diagram.

In reading search parameters and variables part, the upper and lower bound of search parameters and the number of search parameters are entered in MATLAB in order to construct the initial population of the ES algorithm. In addition to this, the population size and initial temperature needed by the ES algorithm is specified in MATLAB. After the determination of initial values for circuit synthesis, the script in MATLAB is ready to run the ES algorithm.

For evaluating performance part, HSPICE is used. After the population is constructed, individuals are evaluated respectively. In evaluation of one individual, its chromosomes are written into a file .param which sets the values of variables for the netlist after which HSPICE is called for the simulation. After the simulation of the netlist, the output files of HSPICE are read to obtain the result of circuit specification as well as drain, gate and source voltages of the transistors. According to these voltages, we determine whether the transistor operates in saturation region or not and calculate the $C_{penalty}$. C_{per} is also computed according to simulation results. With the presence of $C_{penalty}$ and C_{per}, the cost function of one individual is calculated. This process is repeated until all individuals of the population are evaluated.

In convergence controlling block, the decision is based on checking if maximum number of generations is reached or if the population is frozen. If convergence is reached, the script stops. If not, the population is updated.

In update variables part, new parents of the population are constructed according to the selection part of the ES algorithm. Then, the evaluation of the new population is restarted. These processes continue until the convergence is reached.

The majority of computer aided design (CAD) tools depend on W-L based approach which sets the dimension of the width (W) and length (L) of the transistors, along with setting the supply voltages and bias currents. This approach is more flexible because it does not need any pre-calculation steps. In addition to this, it can obtain more accurate results, but often takes plenty of time. Actually, the design space is large and it is directly proportional to the number of the components.

In this approach, each design variable is composed of W's and L's. Its performance is found by directly inserting the W's and L's into the netlist, then simulating with a SPICE. It is shown in Figure 3.3.

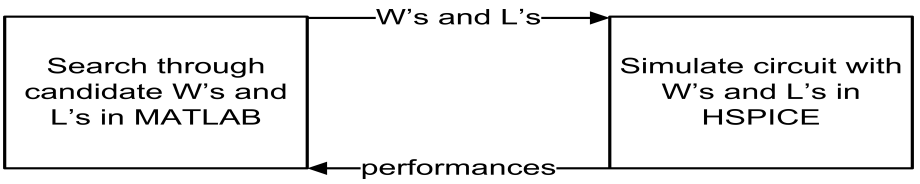

Figure 3.3. WL formulation for sizing.

When it is considered in terms of ES algorithm, the circuit variables composed of the chromosomes of the individual are W's and L's of the transistors. So, the search space of the ES is set according to these variables.

3.3. Circuit Synthesis Examples

The performance of the proposed MATLAB-HSPICE circuit synthesizer has been tested by using a BTS OPAMP topology as well as on FC OPAMP topology. Target technology is a standard 0.18 μm CMOS technology with 1.8 V supply voltage. However, standard 0.35 μm CMOS technology with 3.3 V supply voltage are also used in order to compare the performances of proposed MATLAB-HSPICE circuit synthesizer and SACSES.

In the examples, a (30+20) ES scheme was used and synthesis runs are performed on a dual core 3.0 GHz PC.

The circuit schematic of the synthesized BTS OPAMP is given in Figure 3.4. It is comprised of two subsections of circuits, namely bias stage and gain stage. It was found that this topology was able to successfully meet all of the design specifications.

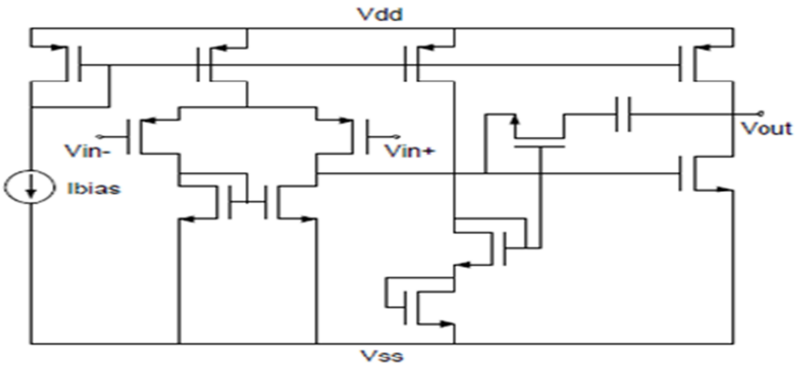

Figure 3.4. The circuit schematics of BTS.

Table 3.1 shows the target and attained performance values with a 1 pF load capacitor. The optimization problem contains 22 independent variables. Table 3.1 shows the best results from proposed synthesis (MATLAB+HSPICE). Both, the MATLAB+HSPICE and SACSES meet the design specifications at nearly same values. It can be seen that the MATLAB+HSPICE 0.18μm mechanism hardly obtain the bandwidth specification and gets considerably low power and rout. Notice that the execution time of the MATLAB+HSPICE is nearly same with SACSES although it requires less iteration. The reason is that MATLAB+HSPICE cannot be utilize acceleration mechanisms of SPASE.

The other synthesized circuit schematic, Folded Cascode (FC) OPAMP is given in Figure 3.5. The basic idea of the FC OPAMP is to apply cascode opamp transistors to the input differential pair.

Table 3.2 shows the target and attained performance values with a 1 pF load capacitor. The optimization problem contains 15 independent variables. Table 3.2 shows the best results from proposed synthesis (MATLAB+HSPICE). It can be seen that the both MATLAB+HSPICE 0.35μm and 0.18μm achieve all specifications nearly

Table 3.1. Specifications of the synthesized BTS OPAMP.

Specification	Target	SACSES 0.35μm	MATLAB 0.35μm	MATLAB 0.18μm
BW(kHz)	> 10	14.4	15	10.4
A0(dB)	> 75	75.5	76.8	75.2
Rout(kΩ)	< 50	28.5	20.8	3.5
Power(mW)	< 5	1.9	1.6	0.02
Area(μm^2)	< 30000	21600	27400	23500
Iteration No		60($\pm$10)	33	34
Time(m)		18($\pm$3)	16	16

same with SACSES. In addition, the execution time of population for the MATLAB+HSPICE was higher than SACSES but it requires less iteration. Notice that area values of two different technologies are similar each other.

Table 3.2. Specifications of the synthesized FC OPAMP.

Specification	Target	SACSES 0.35μm	MATLAB 0.35μm	MATLAB 0.18μm
BW(kHz)	> 10	14.4	13.3	13.2
A0(dB)	> 75	75.8	75.9	74.9
CMRR(dB)	> 65	69	69.9	68.9
SR(V/μs)	> 10	12.2	14.0	14.0
Area(μm^2)	< 10000	8430	8515	8638
Iteration No		64($\pm$10)	33	31
Time(m)		21($\pm$3)	16	15

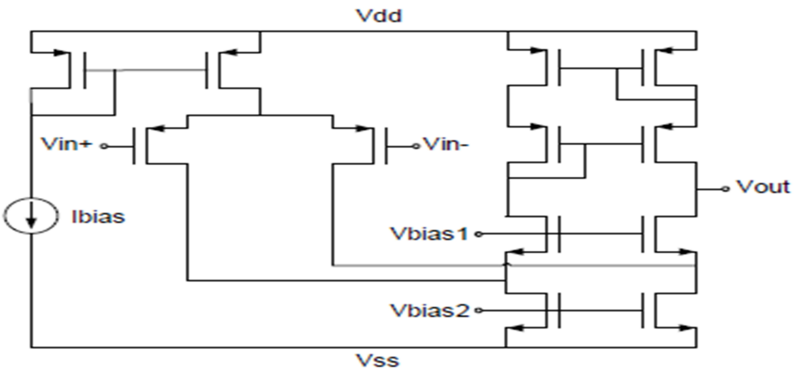

Figure 3.5. The circuit schematics of FC OPAMP.

4. OPTIMIZATION OF VOLTAGES AND INVERSION COEFFICIENT

In simulation-based approaches, the simulation needs a netlist including all transistor dimensions and device values. Using device variables of the sizing process as transistor dimensions makes the search space huge. The optimization process may require many iterations to converge. The solution of this problem is to reduce search space. However, it is not easy to limit the dimension range of transistors from the circuit specifications. In order to overcome this problem, we introduced two different approaches which are voltage-based approach and inversion coefficient (IC) approach.

Both of these approaches reduce the search space through using different circuit variables whose range is limited when compared with the dimension of transistors. Then, they convert these variables into the transistor sizes by a specified conversion mechanism.

4.1. IC Based Approach to Circuit Level Design Automation

IC based approach to circuit level design automation is based on inversion coefficients and the equations of EKV MOSFET Model [30]. It is a fully analytical model dedicated to the design and analysis of low-voltage, low-current analog currents. In this model all variables are continuous in weak, moderate and strong inversion regions. IC equals to the ratio of the drain current over the normalization current. IC values changes due to regimes which is shown here:

$$IC > 10 \qquad : \; strong\, inversion \tag{4.1}$$

$$0.1 < IC < 10 \qquad : \ moderate\ inversion \tag{4.2}$$

$$0.1 > IC \qquad : \ weak\ inversion \tag{4.3}$$

In this approach, ICs and lengths of the transistors are used as design variables. The search is relatively narrow in comparison with WL based approach. In this work, it is assumed that all transistors are in moderate region. The range of IC can be limited with the Equation 4.2. Actually, moderate inversion is highly important for analog/RF IC design due to the following advantages:

- Good trade-off among gain, speed, linearity, noise, matching
- Low-medium saturation voltage, series resistance effect negligible
- Reduced impact of mobility effects (vertical field) and velocity saturation.

However, user can select any region which is more suitable for his design. Then, IC ranges are adjusted according to this choice. Using the IC coefficient, we can easily obtained the width of transistor given its lengths from the following formula:

$$W = |\mathbf{I}_D L|(2N\mu\mathbf{C}_{ox}IC\mathbf{U}_t^2) \tag{4.4}$$

where $\mathbf{I}_D$ is the drain current, N is the slope factor, μ and $\mathbf{C}_{ox}$ are the transconductance model-related parameters, while W and L are the width and length of transistor respectively.

With this formula, a mechanism which converts the inversion coefficients to transistor dimensions is constructed. It is shown in Figure 4.1.

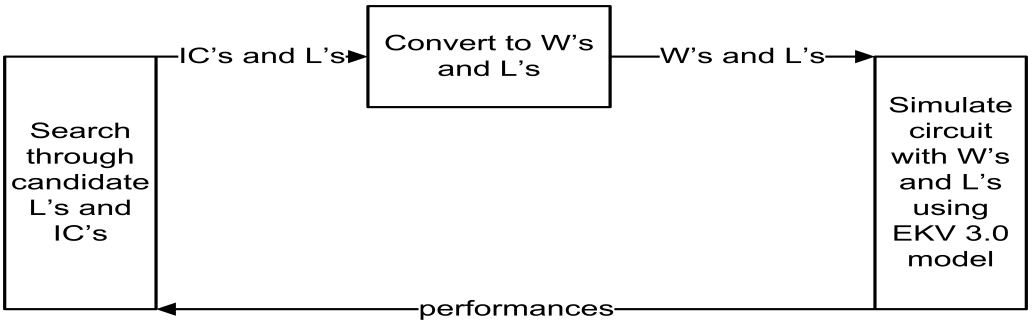

Figure 4.1. IC formulation for sizing.

When it is considered in terms of the ES algorithm, the circuit variables composing of individuals' chromosomes are inversion coefficients and lengths of transistors. So, the search space of the ES is set according to these variables. The other steps of ES algorithm are same.

4.2. Voltage Based Approach to Circuit Level Design Automation

In this approach, the design variables are DC operating points of the circuit: bias voltages and bias currents. The search space can be restricted by the circuit constraints. Range of node voltages are calculated through some inequalities because all transistors are operated in the saturation region. For this reason, this approach depends on circuit topology and requires pre-calculation step. In addition to, it needs a mechanism which converts the bias points to transistors dimensions because the netlist of the circuit needs transistors dimensions. It is shown in Figure 4.2.

Simple look-up tables are used in this conversion. The tables contain the information which provides accurate mapping from bias points to transistor size. The

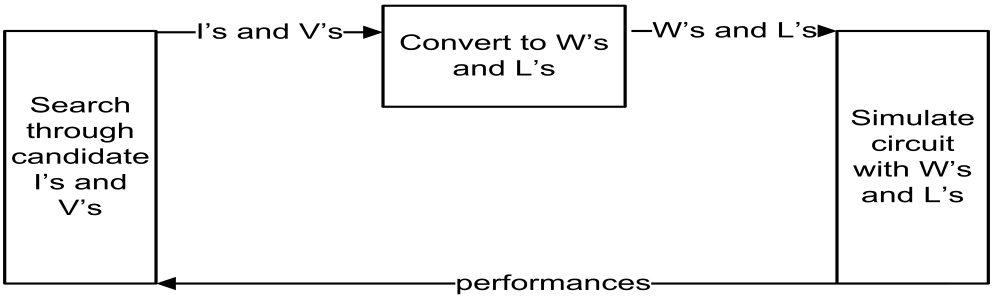

Figure 4.2. Voltage formulation for sizing.

information can be generated for both PMOS and NMOS by a circuit simulator. However, it is difficult to generate a table containing all combinations of bias points and transistor sizes. Therefore, we generate limited look-up tables for the 0.18μm technology. The look-up tables are constructed with respect to the gate to source voltage (Vgs). Vgs is taken 0.4 then it is increased by 0.2 up to 1.8. In this way, 8 tables are generated for NMOS and 8 tables are generated for PMOS. Rows of tables for NMOS shows drain voltage (Vd) and their columns shows source voltage (Vs). In PMOS, rows of tables shows Vs and their columns shows Vd. Both Vd and Vs can range 0 V to 1.8 V. They contain combination of bias voltages providing that all transistors are operated in saturation region and W/L ratio that equals 1. One example of the look-up tables is presented in the Table 4.1.

When bias voltages are given, the related current can be obtained from the look-up table. Then, we sweep a small reference range of transistor size to obtain the bias current from the related current. In this way, the transistors size can be determined according to bias voltages and bias current.

Table 4.1. An example of the look-up table corresponding to Vgs=0.4V.

Vgs = 0.4 V		Vd (V)				
		0∼0.2	**0.2∼0.4**	**.....**	**1.4∼1.6**	**1.6∼1.8**
Vs (V)	**0.4∼0.6**	$\mathbf{I}_{ref}$(1,1)	$\mathbf{I}_{ref}$(1,2)	.....	$\mathbf{I}_{ref}$(1,18)	$\mathbf{I}_{ref}$(1,19)
	0.6∼0.8	$\mathbf{I}_{ref}$(2,1)	$\mathbf{I}_{ref}$(2,2)	.....	$\mathbf{I}_{ref}$(2,18)	$\mathbf{I}_{ref}$(2,19)
	.....	.....	.....	.....	.....	.....
	1.4∼1.6	$\mathbf{I}_{ref}$(16,1)	$\mathbf{I}_{ref}$(16,2)	.....	$\mathbf{I}_{ref}$(16,18)	$\mathbf{I}_{ref}$(16,19)
	1.6∼1.8	$\mathbf{I}_{ref}$(17,1)	$\mathbf{I}_{ref}$(17,2)	.....	$\mathbf{I}_{ref}$(17,18)	$\mathbf{I}_{ref}$(17,19)

When it is considered in terms of ES algorithm, the circuit variables composed of individuals' chromosomes are node voltages and bias current. So, the search space of the ES is set according to these variables. The other steps of ES algorithm are the same.

4.3. Circuit Synthesis Example for Voltage and IC Based Approach

In this section, the implementation of these two approaches and a comparison are performed by using the BTS OPAMP topology. Initially, the implementation of optimization for inversion coefficient is described. The optimization problem contains 22 independent variables. The number is the same for both the WL-based and IC-based approach. However, the search space is different because ranges of IC in moderate region is narrower than range of width. The population is constructed according to boundaries of independent variables. For the simulation, these variables are transformed into the transistor dimensions by using Equation 4.4. The slope factor, N is set at 1.2. The other model related parameters is used with EKV 3.0 parameter set to obtain better results.

Then, optimization of voltages is implemented. In this optimization, to construct initial population with search variables, bias voltages and currents requires the

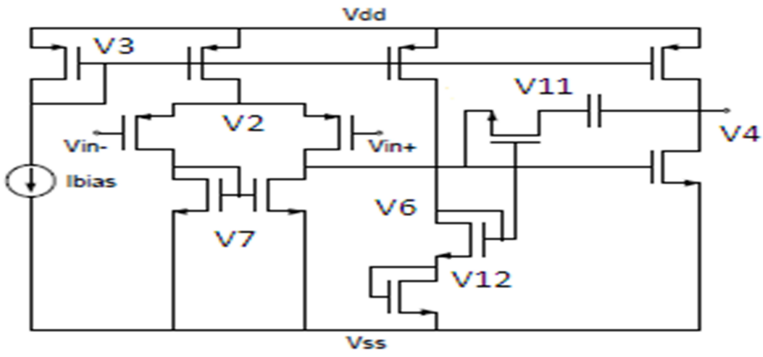

Figure 4.3. BTS OPAMP circuit schematic with node voltages.

specification of their ranges. Their ranges are dependent on circuit topology. It requires pre-calculation step for determination the ranges of search variables. For BTS OPAMP topology shown in Figure 4.3, the design variables would be V2, V3, V4, V6, V7, V11, V12 and Ibias. Because all transistors are operated in the saturation region, the variables can be limited by the following inequalities:

$$0 < V3 < Vdd - Vtp \tag{4.5}$$

$$Vtp < V2 < Vtp + V3 \tag{4.6}$$

$$Vtp < V7 < Vtn + Vss \tag{4.7}$$

$$Vtn + V7 < V6 < Vtp - V3 \tag{4.8}$$

$$Vtn + V3 < V4 < Vtp - V7 \tag{4.9}$$

$$Vtn + Vss < V12 < V6 - Vtn \tag{4.10}$$

$$Vtn + V6 < V11 < V4 \tag{4.11}$$

Vtp and Vtn represent the threshold voltages of PMOS and NMOS transistors respectively. Vdd and Vss are the supply voltages. By these inequalities, the free space of the variables is limited by circuit specifications. The population is constructed according to these circuit specifications. For the simulation, bias voltages and currents are transformed into the transistor dimensions by using look-up tables which were previously constructed for a 0.18μm technology for NMOS and PMOS.

The performance of three different optimization approaches has been tested by using BTS OPAMP. Figures 4.4 - 4.9 show the best results from all proposed optimization. The number of generations means the number of parents for them. They are compared with each other in terms of cost function, the number of iterations, the execution time, some node voltages and meeting design specifications.

When we compare their cost functions, WL optimization has the highest cost, more iterations and requires more time. Although it meets design specifications, the number of transistors operated in the saturation region is fewer than the others. It has relatively higher search space so the number of iteration as well as the execution time increases. On the other hand, V optimization has the lowest cost although it cannot meet all design specifications. The reason of the it lowest cost is that it guarantees that all transistors are operated in the saturation region. It also has lowest iteration number and the execution time because of its limited search space. However, it requires some a prior effort for the construction of look-up tables as well as limit the search space according to the circuit topology. IC optimization seems the best. The cost function,

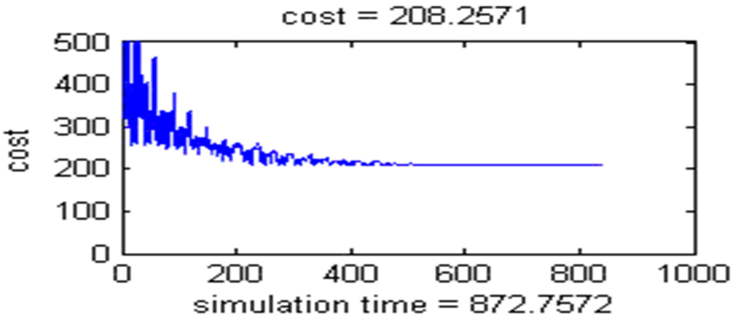

Figure 4.4. Cost function of IC optimization.

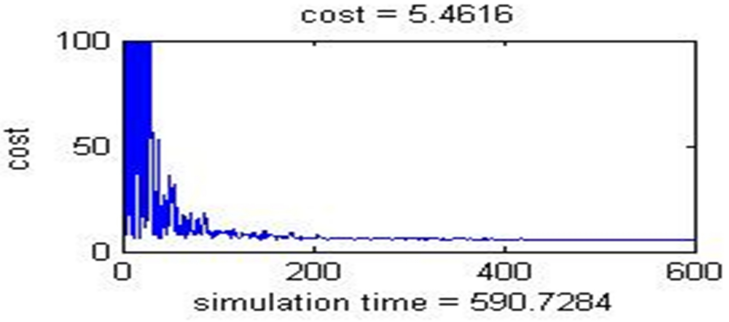

Figure 4.5. Cost function of V optimization.

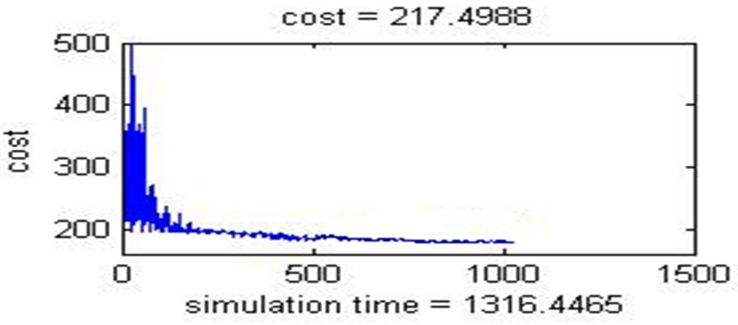

Figure 4.6. Cost function of WL optimization.

iteration number and execution time is lower than WL optimization, as well as not requiring any initial effort before the running. It can compensate the benefits of V optimization with its flexible structure. The only drawback of it is that it may not give good results another technology because of analytical equation of EKV MOSFET model.

Figures 4.7 and 4.8 show the some node voltages of three optimization approaches, V5, V6, V7 and V12 shown in Figure 4.3. It can be seen that these node voltages converge to nearly the same point in WL and IC approaches. However, V optimization is different from them slightly. This difference can be seen clearly in node voltage V6. Although the ranges of node voltages for V optimization are defined with some inequalities, the final values of the node voltages are similar for all of them.

Figure 4.9 shows two design specifications, gain and rout values for all approaches during the optimization. The aim is to obtain higher gain and lower rout in terms of these specifications. It can be seen that all approaches reach the higher gain when they keep their rout lower.

Table 4.2 shows the design specification values for all three approaches. IC optimization has the highest gain and lowest rout. WL optimization provides the all design

specification in desired range. However, its bandwidth value is near the its bad limit. V optimization can also meet all design specification but its rout value is the highest. In Figure 4.9 for V, it can be seen that the values of gain for IC and V optimization are nearly the same.

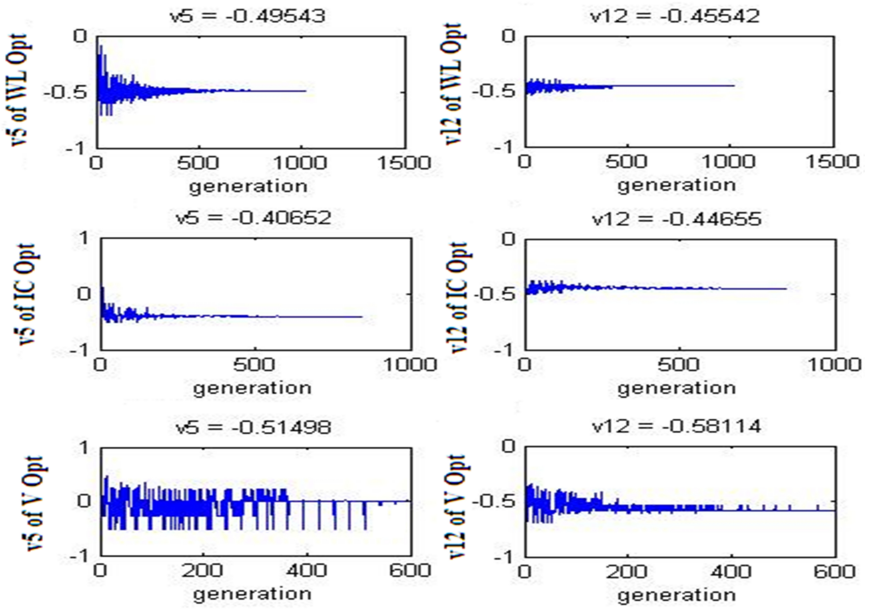

Figure 4.7. V5 and V12 of all optimizations.

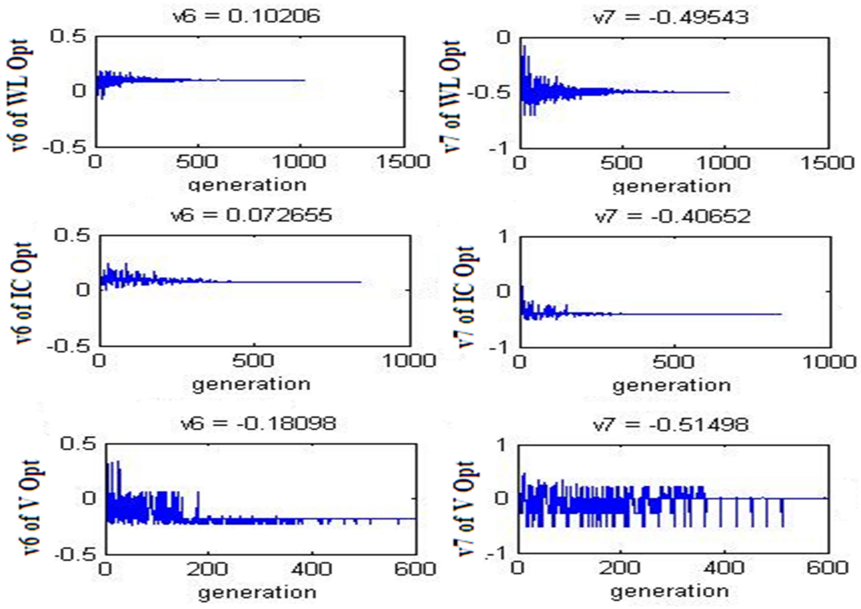

Figure 4.8. V6 and V7 of all optimizations.

Table 4.2. Comparison of the synthesized BTS OPAMP for WL, IC and V optimization.

Specification	Target	WL	IC	V
BW(kHz)	> 10	10.4	36.3	17.9
A0(dB)	> 75	75.2	82.4	82.2
Rout(kΩ)	< 50	3.9	2.5	28.7
Power(mW)	< 5	0.2	0.5	0.1
Area(μm^2)	< 30000	23500	18600	11704
$C_{perform}$		5.346	4.735	5.461
Iteration No		34	28	20
Time(m)		16	15	10

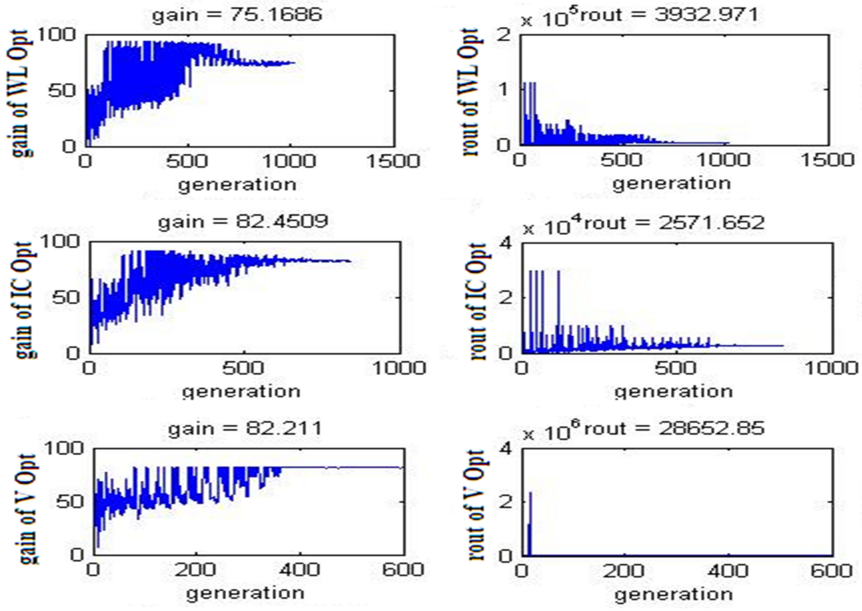

Figure 4.9. Gain and rout of all optimizations.

5. HIERARCHICAL OPTIMIZATION OF ANALOG INTEGRATED CIRCUITS

The structure of Hierarchical Genetic Algorithm (HGA) is more flexible and modular than the conventional genetic algorithm. Multi-layered hierarchical topology of HGA provides to divide large-scale problems into sub-problems by using parallel processed Genetic Algorithm. In this way, the efficiency of the optimization search is increased and the total process time is diminished.

In this study, a two-layered hierarchical genetic algorithm is used to optimize a complex MOS integrated circuit. A 3rd order Butterworth low pass filter is selected for a simulation example. The Proposed HGA is formed of two layers. A master population or first layer which is called system level runs with ES to optimize the values of its own individuals. These individuals are formed by external capacitances, external resistances, and cut-off frequency of filter. The second layer which is called circuit also uses a different ES algorithm to optimize its own individuals. In this layer, transistor based OPAMP circuits are calculated and optimized with SPICE.

In the example, the system level optimizes a 3rd order active Butterworth low pass filter with non ideal OPAMPs. The circuit level optimizes the bandwidth, output resistance and gain of the OPAMPs with SPICE parameters by optimizing the transistor based circuit. After optimizing, the circuit level sends the chip layout areas and chip power consumptions of OPAMPs. This process continues until getting a satisfactory individual. We implemented this idea in three different manners: HGA1, HGA2 and HGA3. HGA alternative 1 [4] and HGA alternative 2 [4] depends on BTS OPAMP macro-model parameters and while the HGA alternative 3 does not and uses ES algorithm hierarchically. All implementations use transistor-level simulation for circuit level performance evaluation.

A 3rd order Butterworth low pass filter shown in Figure 5.1 with the Sallen-Key topology is selected for the simulation and its transfer function with ideal OPAMP's is used:

$$H(s) = \frac{V_0(s)}{V_i(s)} = \frac{{w_c}^3}{s^3 + 2w_c s^2 + 2{w_c}^2 s + {w_c}^3} \tag{5.1}$$

Then, from the circuit transfer function;

$$w_c = \frac{2}{C_1R_1 + C_2R_2 + C_3R_3} \tag{5.2}$$

Input resistance of the OPAMP is not included, due to infinite DC input impedance

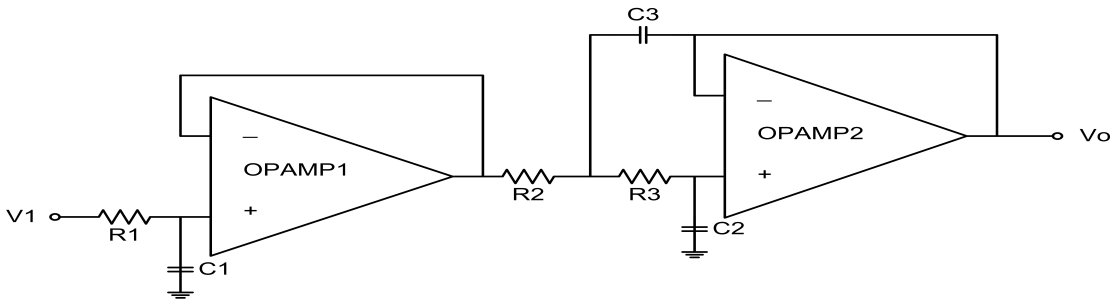

Figure 5.1. 3rd order low pass Butterworth filter with Sallen-Key topology.

of MOSFETs and finite gains and output impedances of the OPAMPs are accounted for by using the OPAMP model of Figure 5.2.

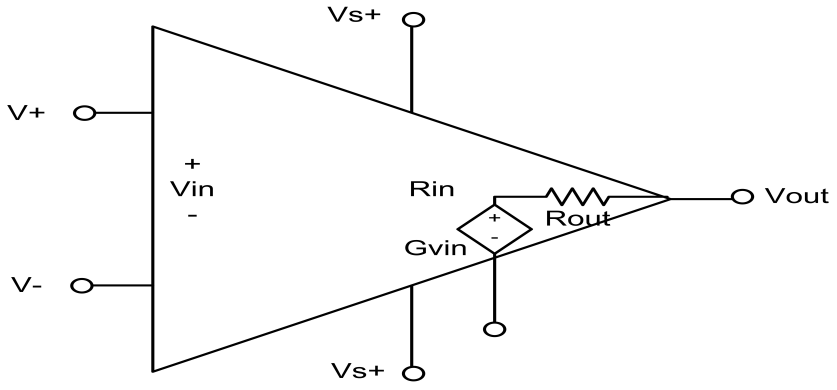

Figure 5.2. Non ideal OPAMP model used in the synthesis example.

Using this non-ideal model, the transfer function takes the form in Equation 5.3.

$$H(s) = \frac{V_0(s)}{V_i(s)} = \frac{K_3 Z_2 s^2 + K_3 Z_1 s + K_3 Z_0}{s^3 + \frac{K_2}{K_3} s^2 + \frac{K_1}{K_3} s + \frac{K_0}{K_3}} \tag{5.3}$$

where the coefficients K_i depend on the external resistor and capacitor values as well as finite gains and output resistances of OPAMPs. K_i are calculated according to [5]. With the non-ideal model, parasitic zeroes are added and the cutoff frequency, f_c, can be obtained in three ways:

$$f_{c1} = \frac{K_2}{4\pi K_3} \tag{5.4}$$

$$f_{c2} = \frac{1}{2\pi}\sqrt{\frac{K_1}{2K_3}} \tag{5.5}$$

$$f_{c3} = \frac{1}{2\pi}\sqrt[3]{\frac{K_0}{K_3}} \tag{5.6}$$

These three cutoff frequencies should be equal because of the ideal Butterworth characteristic.

5.1. Cost Function of the System Level

The cost function [4] is formed from cutoff frequency values, max/min external resistor value, max/min external capacitor value, power consumption value of the OPAMP, and chip layout area value of the OPAMP. When we start to construct the cost function, first, each center frequency is normalized shown in the Equation 5.7.

$$k_{fi} = \frac{f_{ci} - f_{c,target}}{f_{c,max} - f_{c,min}} \tag{5.7}$$

where i = 1,2,3, $f_{c,max}$ and $f_{c,min}$ are the minimum and maximum acceptable values for the center frequency calculated from the user given target frequency, $f_{c,target}$ and tolerance value.

In addition to this, normalization values of ratio for the max resistance to min resistance and the ratio for the max capacitance to min capacitance are added into the cost function. These proportions are vital to make a Butterworth characterization. After adding normalization values of OPAMP power consumption and OPAMP chip layout area obtained from circuit level, the cost function is completed. The normalization of these performance metrics are calculated from:

$$k_{P_j} = \frac{|P_j - P_{j,good}|}{|P_{j,good} - P_{j,bad}|} \tag{5.8}$$

Then, the overall cost function becomes:

$$C_{system} = w_{fc}(k_{f1}^2 + k_{f2}^2 + k_{f3}^2) + w_p k_p + w_a k_a + w_R k_R + w_C k_C \tag{5.9}$$

where w_{fc}, w_p, w_a, w_R, and w_C are the weights for the cutoff frequency, power, area, resistor and capacitor ratios, respectively.

This cost function was embedded in a standard genetic algorithm to form the system level synthesizer [5]. However, we implemented this cost function in ES algorithm. Three different possibilities, HGA alternative 1, 2, and 3 for integration with the circuit level were tested.

5.2. HGA Alternative 1 and HGA Alternative 2

These two approaches are based on macro models and equations and the idea behind them is similar. Their flow diagram is shown in Figure 5.3. The synthesis flow of the first approach can be summarized in the following steps:

- The synthesis starts at the system level. It sends block specifications, gain, bandwidth and rout of individuals to the circuit level with respect to input specifications of the system level.

- Then, individuals are ordered according to system cost values and circuit level synthesis starts from the individual having lowest cost.

- After individuals are optimized in the circuit level entirely with respect to block specifications, the power and area values of OPAMP are sent to the system level.

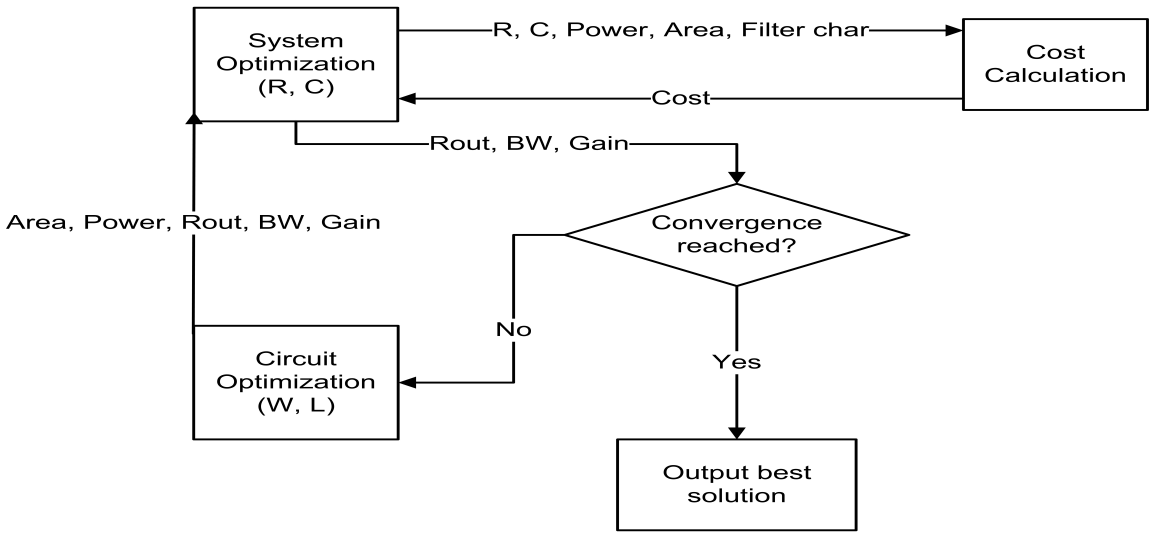

Figure 5.3. HGA alternative 1 and 2 flow diagram.

- System level optimization restarts with external resistors and capacitors. It uses power and area values and the amplifier characteristics for the cost calculation during the optimization process. It is truncated every 20 generations.

- This process continues until both circuit level cost and system level cost drop at desired point.

The synthesis flow of the second approach can be summarized with the following steps:

- The synthesis starts at the system level. It sends block specifications, gain, bandwidth and rout of individuals to the circuit level with respect to input specifications of the system level.

- Then, individuals are ordered according to system cost values and circuit level synthesis starts from the individual having lowest cost.

- After ordered individuals are obtained from the system level, the circuit level optimize them respectively until their cost drops to a predetermined desired level. In this way, overall synthesis time is reduced.

- Then, system level optimization restarts. If its cost drops at desired level, optimization will be ended. If its cost drops at desired level but convergence not, complete synthesis of 20 individuals is performed by circuit level and system level optimization restarts.

- Again, this process stops when both circuit level cost and system level cost drop at desired point.

5.3. HGA Alternative 3

The HGA methods presented in the previous sub-section has high synthesis speed because they rely on macro-models at the system level. Macro-models require considerable amount of initial effort and are topology dependent. In order to overcome this problem, a generalized version of the proposed HGA alternative 3 is implemented. In this approach, three ES algorithm run hierarchically, one for the circuit level and one for the system level. The synthesis flow can be summarized with the following steps:

- The first step is to initialize the variables of the circuit level and system level. In initialization, individuals are generated randomly according to the given gain bandwidth and rout ranges for circuit level design specifications. In addition to this, both system level and circuit level initialize their own search variables.

- Then, circuit level makes optimization with respect to individuals of the population consisting circuit level specifications.

- After circuit synthesis, optimized OPAMPs are inserted into the filter, in other words system level.

- With these OPAMPs, system level optimizes its own search space, external resistors and capacitors, and sends the cost function for each of OPAMPs to the top level.

- If convergence is reached, the iteration will stop. If not, the top level update its search space with selection and recombination operators of ES algorithm. With new population of top level, iteration restarts.

In this approach, there are three populations. The individuals of the first one includes the block specifications of circuit. The second one is for circuit sizing and its individuals are composed of circuit variables. The last one's individuals are external resistors and capacitors of the filter. Circuit level is optimized according to the individuals of the first population. After second population individuals are sent to system level, the system level optimizes the third population and constructs system cost function. According to this cost function, the new parents of the first population is selected and the process restarts. This process continues until the satisfactory solution is obtained. In this way, optimization of the system level is made thanks to this hierarchical approach. The flow diagram of this approach is shown in Figure 5.4.

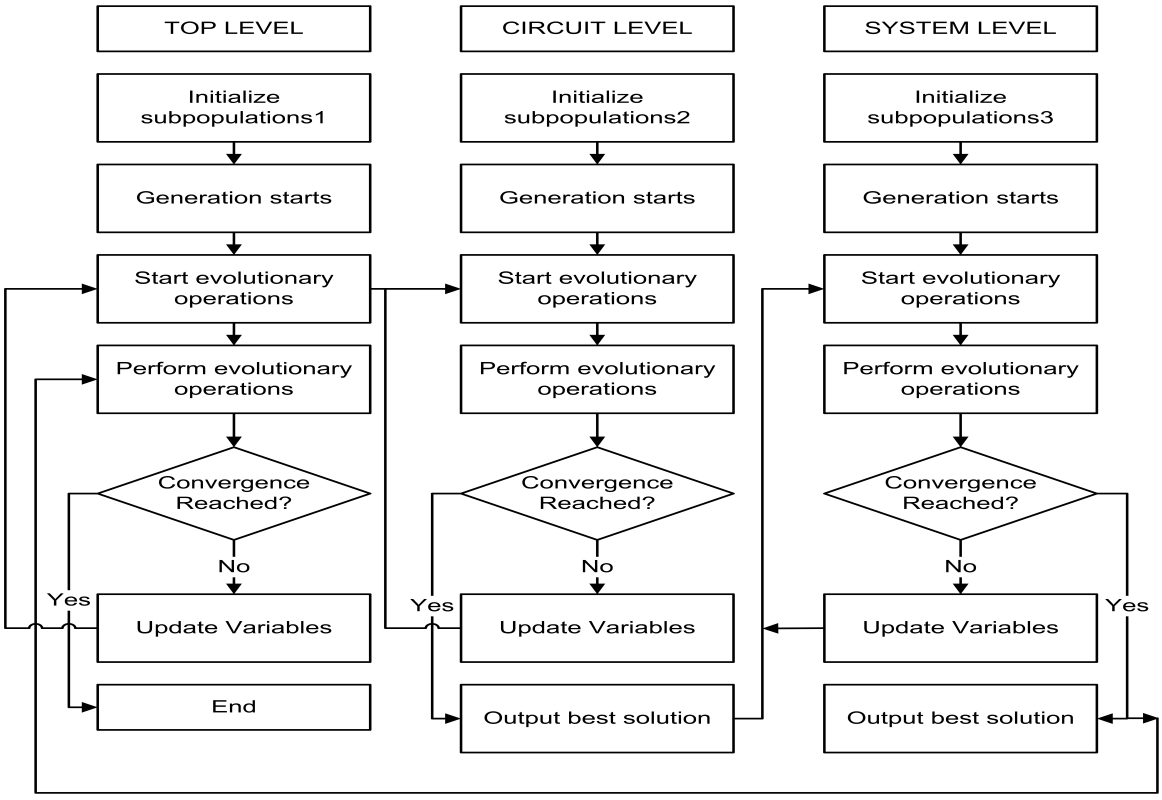

Figure 5.4. HGA alternative 3 flow diagram.

5.4. Implementation of Hierarchical Genetic Algorithms

In this section, the implementation of three HGA approaches are made using 3rd order Butterworth filter as a system example and BTS OPAMP at the circuit level. The target cutoff frequency is set 10 kHz with 0.01 tolerance for the filter. In HGA 1 and 2, the population size for the system level is taken 20 and for the circuit level is 30. In HGA 3, the population size for the top level is 20 and for both circuit level and system level is 30.

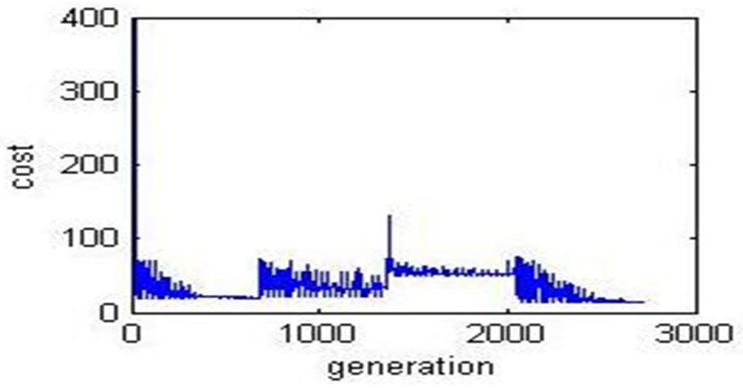

Figure 5.5. Cost function of HGA alternative 1.

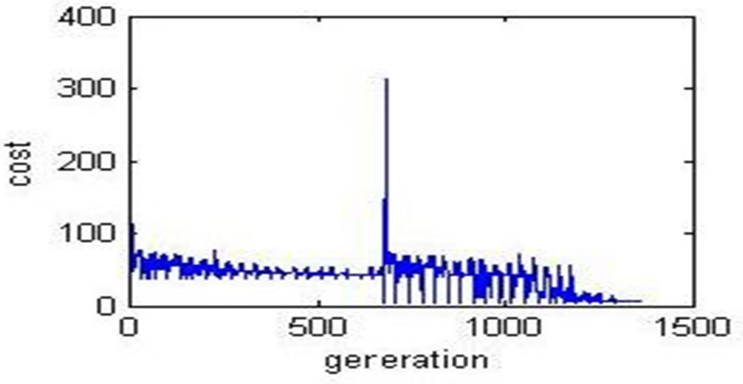

Figure 5.6. Cost function of HGA alternative 2.

Figure 5.5 shows the convergence of the system cost function for the first approach. After the first entire system optimization, the system level cost does not drop below desired point. So, iteration process continues until the cost function reaches the expected point. In this approach, both the system and circuit level are optimized

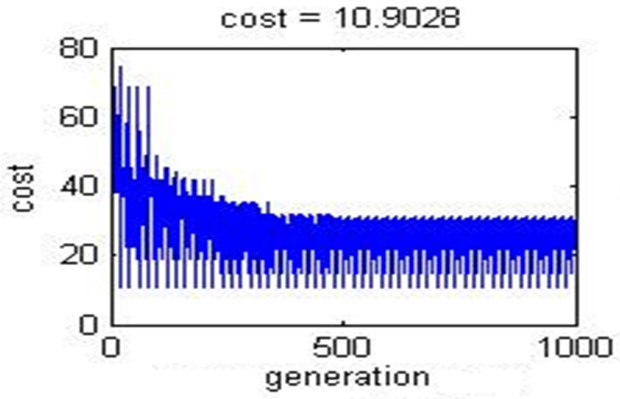

Figure 5.7. Cost function of HGA alternative 3.

entirely in each iteration. For this reason, it takes longer time when it is compared with the second approach.

Figure 5.6 shows the system cost versus number of generation for HGA alternative 2. It can be clearly seen that the it converges more rapidly than the previous approach.

Figure 5.7 shows the convergence of the system cost function for the third approach. It can be clearly seen that the it is similar with Figure 4.4, 4.5 and 4.6 because in the top level of this approach, ES algorithm runs.

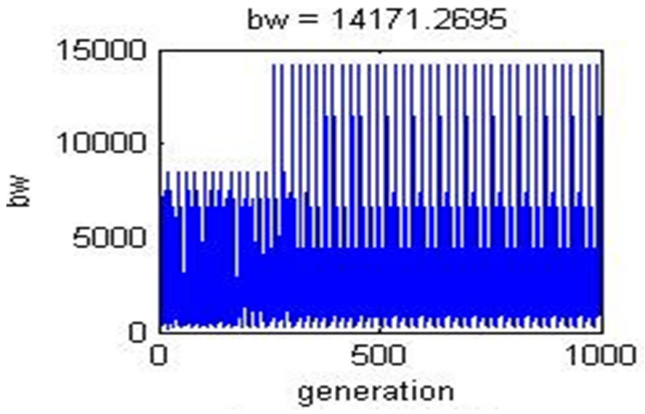

Figure 5.8. Bandwidth of HGA alternative 3.

Figure 5.8, 5.9 and 5.10 shows the bandwidth, gain and rout values of OPAMPS in HGA alternative 3 respectively. The graphs do not converge to one point, indeed they fluctuate. The reason is that the selection mechanism of the algorithm depend

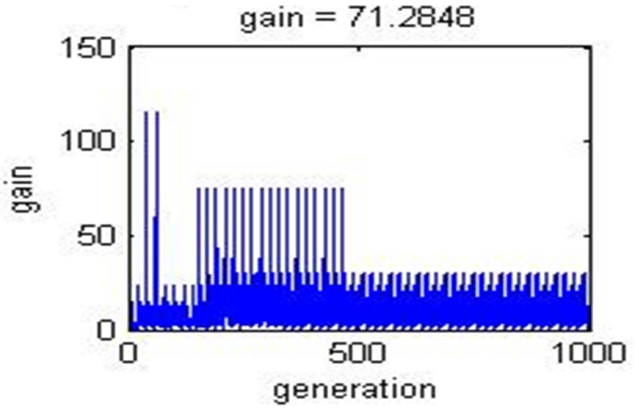

Figure 5.9. Gain of HGA alternative 3.

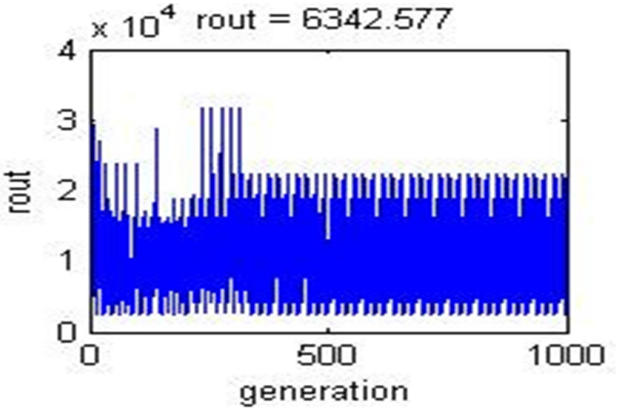

Figure 5.10. Rout of HGA alternative 3.

on system cost function. This means that, it is possible that the cost functions of populations having different values of bandwidth, gain and rout are similar.

Table 5.1 summarizes the results of the three approaches. The target cutoff frequency is set 10 kHz with 0.01 tolerance for the filter. It is clear that the first and second approach have same values of both power and area. The characteristics of the synthesized filters are also similar. However, the total synthesis time for the HGA alternative 2 approach is lower than HGA alternative 1. This means that optimization process makes faster without sacrificing circuit performance in HGA alternative 1 when compared with HGA alternative 2. In HGA alternative 3, the characteristics of the synthesized filter is also similar with HGA alternative 1 and 2. Compared to the macro-model based approaches, HGA alternative 1 and 2, HGA alternative 3 requires more time. However, the time required for the generation of performance equations and macro-models is not included in the macro-model based synthesis time.

Table 5.1. Synthesis results for integration with three different approaches.

Specification	HGA 1	HGA 2	HGA 3
Filter f_c (kHz)	10.4	10.8	9.7
Filter $R_{max}/R_{min}(\Omega/\Omega)$	412/170	401/90	407/94
Filter C_{max}/C_{min}(nF/nF)	809/187	370/38	616/72
OPAMP A_0(dB)	80.2	83.5	71.2
OPAMP BW(kHz)	14.3	17.2	14.2
OPAMP Rout(kΩ)	1.2	2.3	6.3
OPAMP Power(mW)	2	2	0.3
OPAMP Area(μm^2)	26230	26230	20447
Time(sec)	54924	1585	84517

6. CONCLUSION

A simulation-based analog circuit synthesis methodology and its implementation, SACSES, was implemented in MATLAB. In this way, SACSES can use any desired simulator. This maybe necessary for synthesizing some nonlinear or time-variant circuit topologies. For example, a harmonic balance simulator can be implemented and added to synthesize mixers. Furthermore, it is possible making aging-aware synthesis with using an aging simulator. In this thesis, SACSES uses HSPICE as a circuit simulator.

The different set of search variables for ES algorithm is used. With the use of transistor dimensions as search variables, the search space becomes huge. The number of iteration required in order to find optimal solution increases. Also, in the circuit topologies including more devices, the optimization process cannot work efficiently and takes plenty of time.

Using DC operating points as search variables narrows the search space, the number of iteration for convergence decreases. It also guarantees that all transistors operate in saturation region. However, it requires pre-calculation step based on circuit topologies in order to limit the search space. Also, obtaining transistor dimensions from DC operating points requires look-up tables. The construction of look-up tables depends on the technology and takes much time.

Another useful set of design variables inversion coefficients and transistor lengths are used. In this case, the search space is narrowed and optimization process does not require any pre-calculation step and look-up tables. Therefore, it does not depend on circuit topology. However, because it uses EKV MOSFET Model to obtain transistor dimensions from inversion coefficient, the efficiency of optimization depends on technology as well as EKV MOSFET Model.

A hierarchical genetic algorithm structure was proposed for integrating the system and circuit levels. The method takes advantage of the fast initial convergence of the genetic algorithms. The proposed hierarchical scheme was applied to the synthesis of a 3rd order Butterworth filter with three different ways. Two of them depends on macro-model, the other not. The results of first and second approach are similar. However, the convergence time for the second approach is shorter than the first approach. The third approach eliminates the need for macro-model generation and performance equation derivation with using multi-layered HGA.

REFERENCES

1. Han, J. and B. Bhanu, "Fusion of Color and Infrared Video for Moving Human Detection", *Pattern Recognition*, Vol. 40, No. 6, pp. 1771–1784, 2007.

2. Lo, C., P. Chan, Y. Wong, A. Rad and Cheung, "Fuzzy-Genetic Algorithm for Automatic Fault Detection in Hvac Systems", *Applied Soft Computing*, Vol. 7, No. 2, pp. 554–560, 2007.

3. Worapradya, K. and S. Pratishthananda, "Fuzzy Supervisory PI Controller Using Hierarchical Genetic Algorithms", *Control Conference, 2004. 5th Asian*, pp. 1523–1528, IEEE, 2004.

4. Sönmez, S., *Circuit Level Analog Design Automation*, Ph.D. Thesis, Boğaziçi University, 2010.

5. Kayaaltı, B., *Analog Circuit Optimization with Hierarchical Genetic Algorithms - 3RD Order Low-Pass Butterworth Filter Example*, M.S. Thesis, Boğaziçi University, 2005.

6. Harjani, R., R. Rutenbar and L. R. Carley, "OASYS: A Framework for Analog Circuit Synthesis", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, Vol. 8, No. 12, pp. 1247–1266, 1989.

7. El-Turky, F. and E. Perry, "BLADES: An Artificial Intelligence Approach to Analog Circuit Design", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, Vol. 8, No. 6, pp. 681–692, 1989.

8. Degrauwe, M., O. Nys, E. Dijkstra, J. Rijmenants, S. Bitz, B. Goffart, E. Vittoz, S. Cserveny, C. Meixenberger, G. van der Stappen and H. Oguey, "IDAC: An Interactive Design Tool for Analog CMOS Circuits", *Solid-State Circuits, IEEE Journal of*, Vol. 22, No. 6, pp. 1106–1116, 1987.

9. Koh, H. Y., C. H. Sequin and P. R. Gray, "OPASYN: A Compiler for CMOS Operational Amplifers", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, Vol. 9, No. 2, pp. 113–125, 1990.

10. Gielen, G., H. Walscharts and W. Sansen, "Analog Circuit Design Optimization Based on Symbolic Simulation and Simulated Annealing", *Solid-State Circuits, IEEE Journal of*, Vol. 25, No. 3, pp. 707–713, 1990.

11. Van der Plas, G., G. Debyser, F. Leyn, K. Lampaert, J. Vandenbussche, G. Gielen, W. Sansen, P. Veselinovic and D. Leenarts, "AMGIE-A Synthesis Environment for CMOS Analog Integrated Circuits", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, Vol. 20, No. 9, pp. 1037–1058, 2001.

12. Hershenson, M., S. P. Boyd and T. H. Lee, "Optimal Design of a CMOS Op-amp via Geometric Programming", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, Vol. 20, No. 1, pp. 1–21, Jan 2001.

13. Mandal, P. and V. Visvanathan, "CMOS Op-amp Sizing Using a Geometric Programming Formulation", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, Vol. 20, No. 1, pp. 22–38, 2001.

14. Dawson, J. L., S. P. Boyd, M. del Mar Hershenson and T. H. Lee, "Optimal Allocation of Local Feedback in Multistage Amplifers via Geometric Programming", *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, Vol. 48, No. 1, pp. 1–11, 2001.

15. Xu, Y., K. L. Hsiung, X. Li, L. T. Pileggi and S. P. Boyd, "Regular Analog/RF Integrated Circuits Design Using Optimization with Recourse Including Ellipsoidal Uncertainty", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, Vol. 28, No. 5, pp. 623–637, 2009.

16. Papadopoulos, S., R. J. Mack and R. E. Massara, "A Hybrid Genetic Algorithm Method for Optimizing Analog Circuits", *Circuits and Systems, 2000. Proceedings*

of the 43rd IEEE Midwest Symposium on, pp. 140–143, IEEE, 2000.

17. Yuan, J., N. Farhat and J. VanderSpiegel, "GBOPCAD: A synthesis Tool for High Performance Gain-Boosted Opamp Design", *Circuits and Systems I: Regular Papers, IEEE Transactions on*, Vol. 52, No. 8, pp. 1535–1544, 2005.

18. Vladimirescu, A. and R. Zlatanovici, "Analog Circuit Synthesis Using Standard EDA Tools", *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, pp. 5239–5242, IEEE, 2006.

19. Krasnicki, M., R. Phelps, R. A. Rutenbar and L. R. Carley, "MAELSTROM: Efficient Simulation-Based Synthesis for Custom Analog Cells", *Design Automation Conference, 1999. Proceedings. 36th*, pp. 945–950, IEEE, 1999.

20. Phelps, R., M. Krasnicki, R. A. Rutenbar, L. R. Carley and J. R. Hellums, "Anaconda: Simulation-Based Synthesis of Analog Circuits via Stochastic Pattern Search", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, Vol. 19, No. 6, pp. 703–717, 2000.

21. Lin, C., P. Sue, Y. Shyu and S. Chang, "A Bias-Driven Approach for Automated Design of Operational Amplifiers", *VLSI Design, Automation and Test, 2009. VLSI-DAT '09. International Symposium on*, pp. 118–121, IEEE, 2009.

22. Kruiskamp, W. and D. Leenaerts, "Darwin: CMOS Opamp Synthesis by Means of a Genetic Algorithm", *Design Automation, 1995. DAC '95. 32nd Conference on*, pp. 433–438, IEEE, 1995.

23. Miller, J. F., P. Thompson and T. Fogarty, *Algorithms and Evolution Strategies in Engineering and Computer Science: Recent Advancements and Industrial Applications*, Wiley, 1997.

24. Sönmez, S., *A New Approach to Analog Integrated Circuit Optimization*, Ph.D. Thesis, Boğaziçi University, 1988.

25. Kayaaltı, B., *An Analytical Performance Estimation Tool for Analog Computer Aided Design*, M.S. Thesis, Boğaziçi University, 2000.

26. Sönmez, S., *An Optimization-Based Hierarchical Analog Design Automation System*, M.S. Thesis, Boğaziçi University, 2000.

27. Hu, J. and E. Goodman, "The Hierarchical Fair Competition (HFC) Model for Parallel Volutionary Algorithms", *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*, pp. 49–54, IEEE, 2002.

28. Gulsen, M. and A. Smith, *Evolutionary Algorithms*, Springer, 1999.

29. Tang, K., K. Man and R. Istepanian, "Teleoperation Controller Design Using Hierarical Genetic Algorithms", *Industrial Technology 2000. Proceedings of IEEE International Conference on*, pp. 707–711, IEEE, 2000.

30. Grabinski, W., B. Nauwelaers and D. Schreurs, *Transistor Level Modeling for Analog/RF IC Design*, Springer Netherlands, 2006.