DEVELOPMENT OF A FLEXIBLE ANALOG IP LIBRARY

by

Ezgi Kaya

B.S., Electrical and Electronics Engineering, Boğaziçi University, 2014

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
2018

DEVELOPMENT OF A FLEXIBLE ANALOG IP LIBRARY

APPROVED BY:

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

Asst. Prof. Engin Afacan ....

Asst. Prof. İ. Faik Başkaya ....

DATE OF APPROVAL: 05.06.2018

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor, Prof. Günhan Dündar for all of his support and ideas during this work. He introduced me to the research topic, taught how to conduct research and present an academic work, which was an irreplaceable experience for me.

I want to thank Asst. Prof. Engin Afacan for his extensive assistance on this work. Besides providing me technical support during my thesis process, he also taught me how to present an academic work.

Lastly, I should thank to my family, friends and home-mates. Everything would be much harder without them.

I would like to dedicate this thesis to the most wonderful person I have ever known, my mother, who supported me throughout my life. She has always been such a strong women who I have taken as a model.

ABSTRACT

DEVELOPMENT OF A FLEXIBLE ANALOG IP LIBRARY

In this thesis, an analog circuit synthesis and design assistant tool is proposed. The developed tool employs an SPEA2 algorithm as a multi-objective optimization engine to generate Pareto-optimal Fronts (PoF) for a given design problem. An analog library serving as analog IP, was also constructed, which includes pre-optimized PoFs and extracted PoF models for different loading and power limitation conditions. Thus, the user can either generate a new PoF for her/his problem or use the pre-existing PoFs as well as the extracted models without running any optimization step. The developed tool can also be utilized for feasibility checking of a circuit, performance prediction, and topology selection. The tool gives the opportunity of visualization of the design solutions, by allowing the user to verify the Pareto-optimal points in the test benches, to observe the design specifications of a specific design solution. A graphical user interface (GUI) is developed to combine all these utilities. To demonstrate the developed tool, two different OTA topologies and a comparator are examined and all parts of the tool were discussed in detail. Finally, the POFs of the OTA and comparator circuits are composed to obtain the PoF of a higher-level block.

ÖZET

ESNEK ANALOG IP KİTAPLIĞI GELİŞTİRİLMESİ

Bu tezde bir analog devre sentezi ve tasarım asistanı aracı sunulmuştur. Bu araç, belirli bir tasarım problemi için Pareto optimal eğriler (PoF) üretmek için çok amaçlı bir optimizasyon motoru olarak bir SPEA2 algoritmasını kullanmaktadır. Aynı zamanda, önceden optimize edilmiş PoF'leri ve farklı yük ve maksimum güç tüketimi koşulları için çıkarılan PoF modellerini içeren, analog IP olarak görev yapan bir analog kütüphane de inşa edilmiştir. Böylece, kullanıcı ya kendi problemi için yeni bir PoF oluşturabilir ya da herhangi bir optimizasyon adımı çalıştırmadan önceden var olan PoF'leri ve çıkarılan modelleri kullanabilir. Geliştirilen araç ayrıca bir devrenin fizibilite kontrolü ve performans tahminini için, ve topoloji seçimi yapmak amaçlı kullanılabilir. Bu araç, tasarım çözümlerinin görselleştirilmesi ve kullanıcının belirli bir tasarım çözümünün tasarım özelliklerini gözlemlemek için test bench'lerde Pareto-optimal noktaları doğrulamasına imkan verir. Tüm bu yardımcı programları birleştirmek için bir grafik kullanıcı arayüzü (GUI) geliştirilmiştir. Geliştirilen aracı göstermek için iki farklı OTA topolojisi ve bir comparator incelenmiş ve aracın tüm kısımları ayrıntılı olarak tartışılmıştır. Son olarak, OTA ve comparator devrelerinin POF'leri kullanılarak, daha yüksek seviyeli bir bloğun PoF'si elde edilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

D	Density
F	Fitness
i	An individual (solution)
N	Population size
p	Number of design variables
P	Population
P_t	Population at the t^{th} generation
$\bar{P}$	Archive
$\bar{P}_t$	Archive at the t^{th} generation
R	Raw fitness
s	Design variables
S	Strength
t	Current generation number
T	Maximum number of generations
σ_i^k	The distance between individual i and its k^{th} nearest neighbor
$\succ$	Representation of Pareto dominance

LIST OF ACRONYMS/ABBREVIATIONS

EMO	Evolutionary Multi-objective Optimization
EO	Evolutionary Optimization
GBW	Gain x Bandwidth product
IP	Intellectual Property
MOEA	Multi-objective Evalutionary Algorithm
NSGA2	Nondominated Sorting Genetic Algorithm - 2
OTA	Operational Transconductance Amplifier
PAES	Pareto Archived Evolution Strategy
PoF	Pareto-optimal Front
SPEA2	Strength Pareto Evolutionary Algorithm - 2

1. INTRODUCTION

In digital design, automation in both front-end and layout, including verification is in common use [1]. Although there is a lot of room for improvement in the automation of digital design, the analog world lacks much more of automation [2]. One major reason is that in analog design, there are many kinds of parameters to be handled and verified in the test benches, while in digital design there are fewer parameters such as frequency, power and area. Although the circuit complexity and the number of components are generally not very large in analog circuits, many trade-offs among circuit specifications are very hard to be managed. Therefore, beside electronic automation tools, design and reuse has become popular in recent years [3].

An Intellectual Property (IP) is a reusable unit of logic, cell or chip layout design. The reuse of IPs is an important issue since it allows the designers design chips faster. Among the digital IPs, standard cell libraries, CPU cores, and memory block generators are in common use [4]. However, there are very few studies for analog IP integration and reuse in the literature [5], [6]. Since there are many interacting performance specifications in an analog design, engineers generally design analog circuits by hand, one device at a time. This issue gave rise to research on analog design automation.

In the past couple of decades, most approaches to analog sizing have been built around optimization-based approaches, that is, some iterative optimization algorithm with a performance evaluator in the loop. The sizing problem has usually been formulated as:

$$\min\ f(x) \text{ subject to } g(x) \geq 0,\ X_L < x < X_H \tag{1.1}$$

where $f(x)$ is the set of objective functions to be minimized. Vector x represents the design variables with X_L and X_H being their lower and upper bounds, respectively. The vector $g(x)$ represents the design constraints. Recently, a wave of innovation has arisen with the development and application of multi-objective optimization algorithms to this kind of problems. Several algorithms have been reported and successfully applied to design automation problems [7]. The outcome of the application of those algorithms to problems with several mutually conflicting design objectives is a Pareto-optimal performance front, which shows the best trade-offs among the objective functions.

Figure 1.1 shows a conventional analog circuit design flow. The first step is to select an appropriate topology that meets the design specifications. Once the topology is selected, the second task is to determine the device sizes, bias voltages, and currents. After circuit sizing is completed, it is verified in the test benches. Then, the layout is constructed and verified in the test benches again. Depending on the post-layout simulations, the sizing of the circuit may be re-adjusted. It may also be concluded that it is not possible to meet the given design specifications using the selected topology. In this case, the design flow starts over by selecting another topology.

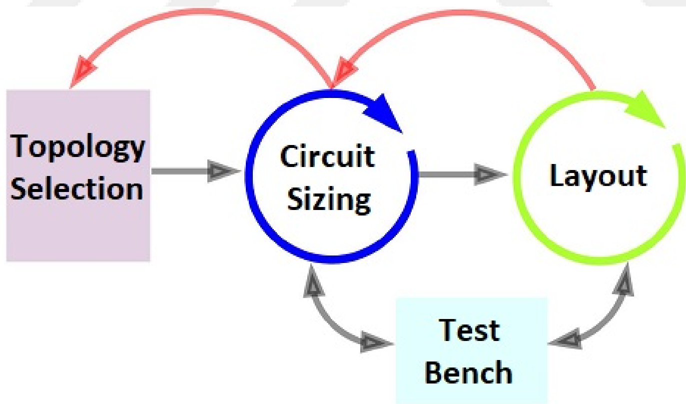

Figure 1.1. Analog design flow.

1.1. Contributions

In this work, an analog circuit synthesis and design assistant tool is proposed. The developed tool employs an SPEA-2 algorithm as a multi-objective optimization engine, to generate Pareto-optimal fronts (PoF) for a given design problem. An analog library serving as analog IP is also constructed, which includes the pre-optimized PoFs and the extracted PoF models for different loading and power limitation conditions, which can be used many times by choosing the design solution that is proper for the performance requirements in each case. As the PoFs reveal the topology limits, they can also be used to make an appropriate topology selection at the beginning of the design. The Pareto-optimal points (PoP) on the PoFs can be further verified in the test benches using the developed tool. The simulations are done in HSPICE and the waveforms can be viewed using CosmosScope.

It is a well-known fact that despite the availability of improved design automation tools, it is still hard to re-synthesize a circuit every time the design requirements change. Therefore, the developed tool aims to save the designers a lot of time. More analog blocks such as oscillators and phase locked loops (PLL) can be optimized and added to the analog library in the future. The developed tool can be improved by adding a layout synthesis feature to complete the design flow.

This thesis is organized as follows: In Chapter 2, Evolutionary Multi-objective Optimization is defined and several Multi-objective Optimization Algorithms are described. In Chapter 3, the developed tool is represented and in Chapter 4, the analysis of different circuit topologies is performed. Finally, this work concludes in Chapter 5.

2. EVOLUTIONARY MULTI-OBJECTIVE OPTIMIZATION

Evolutionary multi-objective optimization (EMO) has become a popular area of research over the last 30 years. Optimization means finding the best possible solution to a problem. If the problem has a single objective to be optimized, then it is aimed to find the best possible solution, which is called the global optimum. However, optimization problems generally have more than one objective to be optimized, which are most likely to be in conflict with each other (otherwise only a single solution exists for that problem). In other words, the optimal solution for one objective is different than the optimal solution for the other objective. For instance, when designing an OTA, the designer may want to maximize the GBW while minimizing the power consumption, which has a trade-off in between. These optimization problems with two or more objectives to be optimized are called multi-objective optimization problems.

In order to take an optimal decision when there is a trade-off between the design objectives, an old notion of optimality is adopted. This notion of optimality was first presented by Francis Ysidro Edgeworth in 1881 [8]. Then it was generalized by economist Vilfredo Pareto in 1896 [9].

2.1. Pareto Principle

For a multi-objective optimization problem, there does not exist a single solution that optimizes all of the objective functions at the same time, but there exists a number of Pareto-optimal solutions, which constitute the so called Pareto curve. A solution is regarded as Pareto-optimal if there does not exist another solution that performs better in terms of at least one objective, without performing worse in terms of any other objectives. In other words, a solution can be Pareto-improved if there exists another solution that has at least one objective value better off and none of the objective values worse off. All the Pareto-optimal solutions in a Pareto curve are considered equally good. The selection among these Pareto-optimal solutions is done by the designer according to her/his additional subjective preference information.

Figure 2.1 illustrates the Pareto-dominance and the Pareto-optimal front concepts for a two-dimensional performance space. In this example, there are two objective functions to be maximized. The two solutions x_1 and x_2 are non-dominated. They do not dominate each other; because even though one solution performs better than the other solution in terms of one of the objective functions, it performs worse in terms of the other objective function. However, the solution x_3 is dominated because it performs worse than the other solutions, in terms of both objective functions.

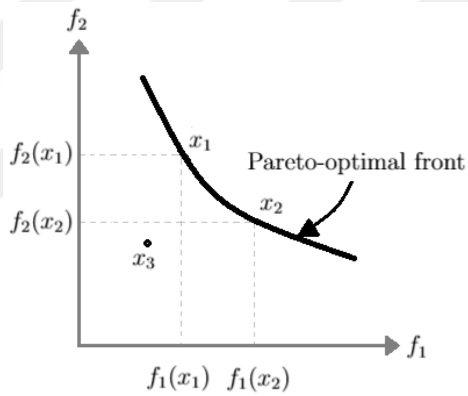

Figure 2.1. An illustration of the Pareto-dominance and the Pareto-optimal front concepts.

In order to generate Pareto fronts, multi-objective evolutionary algorithms are used. Multi-objective evolutionary algorithms use a population based approach in which a set of candidate solutions (so called individuals) evolves along a number of iterations (so called generations). Therefore, the optimization algorithm requires the performance evaluator to execute a number of times (typically in the order of thousands) to achieve the optimal set of solutions. Reducing the required number of iterations to achieve the Pareto-optimal front and decreasing the evaluation time are important objectives in the implementation of evolutionary algorithms. The generation of Pareto-optimal fronts may be an expensive process in terms of the computation time but once they are generated, they can be used many times by choosing the design solution that is proper for the performance requirements in each case.

2.2. Multi-objective Evolutionary Algorithms

The direct integration of Pareto optimality concept into an evolutionary algorithm was first done by David E. Goldberg [10]. In his book, he suggested the use of non-dominated ranking in order to direct a population of solutions to the Pareto-optimal front in a multi-objective optimization problem. The aim is basically to find the solutions in a population that Pareto-dominate the other solutions in the population. These Pareto-non dominated solutions are assigned the highest rank. Then, from the remaining population, another set of Pareto-non dominated solutions are selected and are assigned the next highest rank, until all the population members are assigned a rank. He also suggested the use of a niching technique such as fitness sharing, to prevent the algorithm from converging to a single point on the Pareto-front. Although Goldberg did not exhibit an implementation of his procedure, all the multi-objective evolutionary algorithms developed after Goldberg's publication were influenced by his ideas.

Since evolutionary optimization (EO) algorithms have become a popular method in solving multi-objective optimization problems, several successful multi-objective evolutionary algorithms (MOEA) has been proposed such as NSGA2, SPEA2 and PAES. For this thesis, the SPEA2 is utilized, since it yields promising results in comparison with the other methods, in terms of converging near the ideal Pareto-optimal front [11].

In evolutionary algorithms, many solutions which constitute a population, undergo iterations and in each iteration, a new population is generated. Evolutionary algorithms have become popular because they are relatively simple to implement, they do not require any derivative information, and they can be used in many areas. When there is a single-objective optimization problem, using a population of solutions to find a single optimal solution may be unnecessary. But when there is a multi-objective optimization problem, using evolutionary optimization algorithms is a good choice [12]. The most representative multi-objective evolutionary algorithms are the following.

2.2.1. Non-dominated Sorting Genetic Algorithm - 2 (NSGA2)

NSGA2 was proposed [13] as an improved version of the NSGA [14]. In NSGA2, a fast non dominated sorting approach is utilized. In this context, first, two entities are calculated for each individual i in the population P :

- The number of individuals that dominate i, n_i
- The set of individuals that are dominated by i, S_i

If $n_i = 0$, i.e. no individuals dominate i, then i belongs to the first non-dominated front F_1 whose rank is 1. For each solution i with $n_i = 0$, each individual j in its set S_i is visited and its domination count n_j is reduced by 1. If the domination count of any individual j becomes 0, then it is taken into a separate list Q which constitute the second non-dominated front F_2 whose rank is 2. This procedure continues for each individual in Q and as a result, a third non-dominated front F_3 with rank 3 is obtained. This process continues until all fronts are identified. After assigning the ranks, the crowding distance in each front is calculated. Crowding distance is an estimate of density of solutions around a particular solution in the population, which is computed by averaging the distance of two points on either side of this point along each of the objectives. Afterwards, the environmental selection is done such that if the size of F_1 is smaller than the population size N, all the individuals in F_1 is chosen for the new population P_{t+1} and the remaining members of P_{t+1} are chosen from the subsequent fronts in the order of their ranking. In the selection of members of the last front F_i (which can be F_1 as well), the crowding distances of the members are taken into account, such that the less crowded individuals are chosen for P_{t+1}. In other words, the non-dominated solutions are preferred to the dominated solutions, but the decision between two individuals of the same non-domination rank is done according to their crowding distances such that the less crowded individual is preferred. The population P_{t+1} is then used to create a new population, after undergoing selection, crossover and mutation operations. Individuals in the first non-dominated front are more likely to reproduce.

NSGA2 does not make use of an external memory as in some other multi-objective evolutionary algorithms such as SPEA which was the first multi-objective evolutionary algorithm to incorporate external populations. Alternatively, it adopts an elitist mechanism where the best parents and the best off-springs are combined. NSGA2 is much more efficient than its predecessor and its performance has proved to be good, which is the reason that it has become popular in the last decade.

2.2.2. Pareto Archived Evolution Strategy (PAES)

The PAES algorithm [15] adopts an evolution strategy where one single parent produces one single off-spring. In this algorithm, the non-dominated solutions of the previous generation are stored in an external memory, which constitutes a reference set by which each mutated individual is compared. The PAES algorithm utilizes a crowding technique where the objective space is divided iteratively, in order to maintain diversity. Each individual is positioned in a certain grid location according to its objective values, which constitutes that individual's coordinates. This way, a map of such grid is obtained, indicating the number of solutions in each grid. No additional parameters except the number of divisions of the objective space is required by this adaptive grid procedure. This procedure has been utilized by several modern multi-objective evolutionary algorithms.

2.2.3. Strength Pareto Evolutionary Algorithm - 2 (SPEA2)

SPEA2 is an improved version of SPEA which was published by Zitzler and become a landmark in the area, introducing the elitism concept in a multi-objective evolutionary algorithm. After the introduction of SPEA, it has become very popular to use external populations in multi-objective evolutionary algorithms. As a matter of fact, the use of elitism is a theoretical requirement to ensure convergence of the algorithm.

Elitism usually refers to the usage of an external population, which is called the archive, to maintain the non-dominated individuals found along the evolutionary process. The idea behind using an archive is that a non-dominated solution in a population may not be non-dominated by the individuals in other populations. However, an individual is desired to be non-dominated with respect to all the individuals that the algorithm has generated. Thus, the non-dominated individuals are stored in an archive and an individual has to dominate all the archive members to be able to enter the archive. Similarly, if an archive member is dominated by an individual, it is removed from the archive.

SPEA2 comprises a fine-grained fitness assignment technique that for each individual takes into account the number of individuals it dominates and the number of individuals it is dominated by. The fitness assignment is done by considering both closeness to the ideal Pareto-front and even distribution of solutions at the same time. It uses a nearest neighbor density estimation technique to ensure that the solutions are properly distributed along the Pareto-front. SPEA2 also has an improved archive truncation method that guarantees the preservation of the boundary solutions.

2.2.4. Comparison of MOEAs

Table 2.1 summarizes the comparison of the well known MOEAs described in Section 2.2.1 - 2.2.3 [16]. SPEA-2 is a very good example for the use of external populations. The use of an external population provides the storage of the non-dominated solutions found so far in the search. One other advantage of SPEA-2 is the use of a truncation method which makes sure that the extreme points are preserved. These features allow the algorithm have an advantage in terms of converging near the ideal PoF. However, SPEA-2 is a computationally expensive algorithm in comparison with the other MOEAs, such as NSGA-2. NSGA-2 adopts an elitism approach without using a secondary external population. Differently from SPEA-2, NSGA-2 utilizes a crowding distance approach to obtain a uniform spread of solutions. The main advantage of the crowding approach is that a measure of the population density around an individual is calculated without the need of a user defined parameter such as σ_{share} and the k^{th} nearest neighbor. A disadvantage of the crowding distance technique is that it works in objective space only. These advantages and disadvantages of the algorithms should be taken into account while determining the proper algorithm for an application.

Table 2.1. Comparison of MOEAs.

Algorithm	**Fitness assignment**	**Diversity mechanism**	**Elitism**	**External population**	**Advantages**	**Disadvantages**
PAES	Pareto dominance is used to replace a parent if offspring dominates	Cell-based density as tie breaker between offspring and parent	Yes	Yes	Random mutation hillclimbing strategy Easy to implement Computationally efficient	Not a population based approach Performance depends on cell sizes
NSGA	Ranking based on non-domination sorting	Fitness sharing by niching	No	No	Fast convergence	Problems related to niche size parameter
NSGA-2	Ranking based on non-domination sorting	Crowding distance	Yes	No	Single parameter (N) Well tested Efficient	Crowding distance works in objective space only
SPEA	Ranking based on the external archive of non-dominated solutions	Clustering to truncate external population	Yes	Yes	Well tested No parameter for clustering	Complex clustering algorithm
SPEA-2	Strength of dominators	Density based on the k-th nearest neighbor	Yes	Yes	Improved SPEA Make sure extreme points are preserved	Computationally expensive fitness and density calculation

3. THE PROPOSED TOOL

In this chapter, the features of the developed tool are proposed. The flowchart of the developed tool, named DATA-IP, can be seen in Figure 3.1, and the graphical user interface (GUI) of DATA-IP can be seen in Figure 3.2.

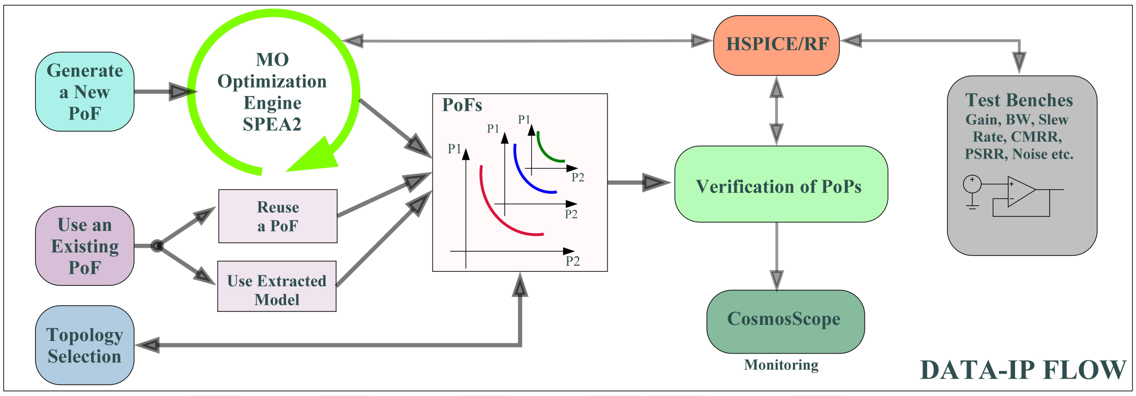

Figure 3.1. Flowchart of DATA-IP.

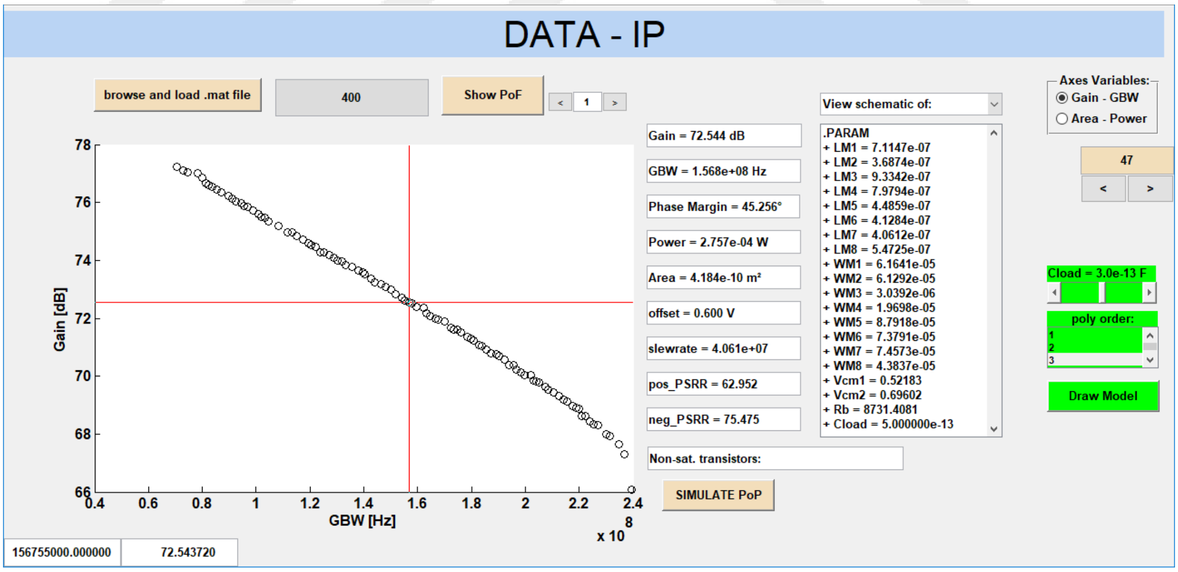

Figure 3.2. GUI of DATA-IP.

3.1. Multi Objective Optimization Algorithm: SPEA2

In this section, the developed multi objective optimization algorithm which utilizes the SPEA2 approach is discussed in detail. The algorithm starts with an initial population and an empty archive.

- Population initialization : An initial population P_0 is generated using the quasi-monte carlo method, which utilizes a low-discrepancy sequence called the sobol sequence. N individuals are initially created using this method, at the beginning of the optimization. The empty archive $\bar{P}_0$ is also created.
- Simulation : All the individuals are then simulated in HSPICE. The individuals who does not meet the design constraints are punished with a high fitness value. If any one of the transistors are not in saturation region, that individual is also punished.
- Fitness Calculation : Fitness, $F(i)$, calculation is done following the below steps:
  (i) Each individual i in the population and the archive is assigned a strength value $S(i)$, which is equal to the number of population and archive members that are dominated by i in terms of the objective values

$$S(i) = |\{j | j \in P_t + \bar{P}_t \wedge i \succ j\}| \tag{3.1}$$

  where " $\succ$ " represents the Pareto-dominance.

  (ii) Then the raw fitness $R(i)$ is calculated by summing up the strengths of the individual i's dominators in both the population and the archive. It must be noted that the lower the raw fitness value, the better, i.e. if an individual has a raw fitness value of zero ($R(i) = 0$), that means the individual i is non-dominated. If $R(i)$ is a high value on the other hand, that means the individual i is dominated by many other individuals.

$$R(i) = \sum_{j \in P_t + \bar{P}_t, j > i} S(j) \tag{3.2}$$

  (iii) The final component of the fitness value is the density, $D(i)$. Density information provides additional selection criterion when many of the individuals do not dominate each other, i.e. have the same raw fitness value. To estimate the density in an individual's neighborhood, the k^{th} nearest neighbor technique is utilized [17]. In this technique, the distances between the individuals in the population and the archive are calculated and sorted in an increasing order. The k^{th} element of this list gives the distance σ_i^k, where k is the square root of the sum of the number of individuals in the population and the archive ($k = \sqrt{2N}$). Then the density is calculated as

$$D(i) = \frac{1}{\sigma_i^k + 2} \tag{3.3}$$

It must be noted that 2 is added to the denominator to prevent the denominator from being equal to zero and also to make $D(i) < 1$. Finally, the sum of density and raw fitness gives the fitness value.

$$F(i) = D(i) + R(i) \tag{3.4}$$

- Environmental selection : Environmental selection is basically the determination of which individuals to keep during the evaluation process. After the fitness values of all the members in the population and the archive are calculated, an environmental selection is done to update the archive. Firstly, the non-dominated individuals (i.e. individuals with fitness values smaller than 1) in both the population and the archive are stored in a temporary list, L.

$$L = \{i | i \in P_t + \bar{P}_t \wedge F(i) < 1\} \tag{3.5}$$

If the number of individuals in L is equal to the archive size N ($|L| = N$), then the individuals in L are copied to $\bar{P}_{t+1}$ and the environmental selection is completed without any future effort. If the number of individuals in L is smaller than the archive size N ($|L| < N$), then the individuals in P_t and $\bar{P}_t$ with lower fitness values are chosen to fill the archive, $\bar{P}_{t+1}$. This is done by sorting the individuals in P_t and $\bar{P}_t$ according to their fitness values and copying the first $N - |L|$ individuals with fitness value greater than 1 to $\bar{P}_{t+1}$. Else if the number of individuals in the list is larger than the archive size N ($|L| > N$), then truncation is done, where the individuals in L with the smallest distance to its nearest neighbor are eliminated [18], until the size of the list reduces to the size of the archive, N, and those remaining individuals are then copied to $\bar{P}_{t+1}$.

- Mating selection : Mating selection is basically the determination of which individuals to produce off-springs. In order to determine the new population, first, a mating pool consisting of $(1.5) \times (N/2)$ individuals is created by using a binary tournament method. In this method, two individuals from the archive are randomly selected and the one with the lower fitness value is copied to the mating pool. This process continues until the mating pool is filled.
- Recombination : New individuals are created by randomly choosing two individuals from the mating pool and multiplying their design variable values ($s_{parent1}$ and $s_{parent2}$) with a recombination coefficient, r, such that

$$s_{child} = r \times s_{parent1} + (1 - r) \times s_{parent2} \tag{3.6}$$

where, $r = 0.6 + (0.2) \times rand()$. The recombination coefficient is determined such that it gives off-springs closer to either of the parents. It was observed that a recombination coefficient of $r = 0.6$ - 0.8 is a better choice, which is also supported by other studies [19].

- Mutation : In favor of diversity, some individuals of the new population undergo a mutation operation. Mutation is done by randomly selecting one of the design variables of that individual and replacing it with a random value between the lower and upper limits defined for that variable. The determination of the individuals to be mutated is a random selection process, which is controlled to some extent, with a pre-defined mutation step size operator. A random number between 0 and 1, $rand()$, and a $mutationStepSize$ operator are generated for each individual and if $rand() < mutationStepSize = 0.2 + (0.1) \times rand()$, then that individual undergoes mutation.
- Termination : The maximum number of generations, T, must be chosen such that the iterations continue until the Pareto front takes its final shape and no longer improves with more iterations. When T is reached, the algorithm stops and the Pareto-optimal front is obtained.

A summary of the developed algorithm flow is given in Figure 3.3 and it is illustrated in Figure 3.4.

The algorithm flow :

input: N (size of the population and the archive)

T (number of generations)

p (number of design variables)

output: A (non dominated set)

Step-1: Population initialization: Generate the initial population P_0 and the empty archive $\bar{P}_0 = \emptyset$. Set t=0.

Step-2: Fitness Assignment: Calculate the fitness values of the individuals in both the population P_t and the archive $\bar{P}_t$.

Step-3: Environmental Selection: Copy all non-dominated individuals in P_t and $\bar{P}_t$ to $\bar{P}_{t+1}$. If the size of $\bar{P}_{t+1}$ exceeds N, reduce $\bar{P}_{t+1}$ by doing truncation. If the size of $\bar{P}_{t+1}$ is less than N, then fill $\bar{P}_{t+1}$ with the individuals in P_t and $\bar{P}_t$, who has lower fitness values.

Step-4: Mating selection: Perform binary tournament among the individuals in $\bar{P}_{t+1}$, in order to determine the mating pool.

Step-5: Recombination and mutation: Apply recombination and mutation operations to the mating pool, to create the new population P_{t+1}. Increase the generation count by 1, $t = t + 1$, and go to Step-2. Continue iterations until t reaches T.

Step-6: Termination: When $t = T$ is reached, set A to $\bar{P}_{t+1}$ and stop.

Figure 3.3. The algorithm flow.

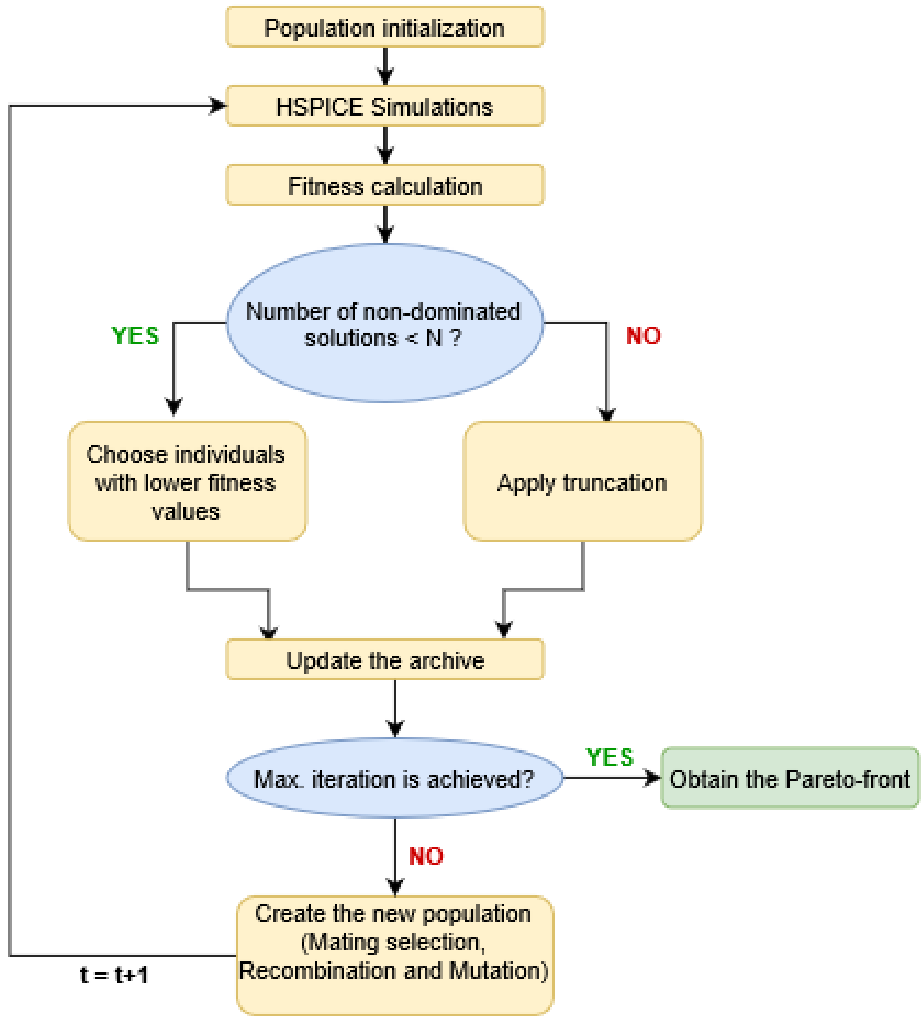

Figure 3.4. The flow chart of the developed algorithm.

3.2. Modeling of the Pareto Optimal Fronts

The generation of the PoFs is a computationally expensive process. However, once they are generated, they can be re-used many times by selecting the design solution with the appropriate performance trade-off in each case. At this point, a problem arises due to the fact that some circuit performances do not depend only on the circuit itself but also on the other circuitry to which it is connected. The conclusions drawn from the performance front obtained for some given conditions cannot be thus directly extrapolated to some other conditions. In the optimization process of analog circuits, some of the performances (e.g. bandwidth and phase margin) depend on the load conditions. However, it is not possible to know the load conditions until any other circuitry around the designed circuit is known. A solution to this problem is to create a model of the circuit, which defines its performance depending on the load conditions, so that when the load conditions are known, the performance of the circuit can easily be determined using this model. In this study, a model that predicts the circuit performance depending on the load capacitance, C_L, is created by using the following method. A number of PoFs are generated for different C_L values. Using the curve fitting property of MATLAB, an n^{th} order polynomial is obtained for each PoF such that

$$f(x) = p_1 \times x^n + p_2 \times x^{n-1} + \cdots + p_{n+1} \times x^0 \tag{3.7}$$

where x is the normalized GBW (i.e. GBW/10^8) and $f(x)$ is gain.

Then using the curve fitting property, each coefficient p is written in terms of the C_{load} value such that

$$p_i(x) = a_1 \times x^n + a_2 \times x^{n-1} + \cdots + a_{n+1} \times x^0 \tag{3.8}$$

where x is the C_{load} value. Finally, the p_i values in Equation 3.8 are substituted in Equation 3.7 so that $PoF(gain, GBW) = f(C_{load})$ is obtained. The user can select a C_L value of interest and draw the PoF for that C_L value. It must be noted that the PoF model can only be used for the C_L values between the minimum and maximum C_L values that the circuit has been evaluated.

Using the same method, a model that shows the circuit performance depending on the power limitation is also created. A number of PoFs are generated for different power limitations. Using the curve fitting property of MATLAB, an n^{th} order polynomial is obtained for each PoF such that

$$f(x) = p_1 \times x^n + p_2 \times x^{n-1} + \cdots + p_{n+1} \times x^0 \tag{3.9}$$

where x is the normalized GBW (i.e. GBW/10^8) and $f(x)$ is gain.

Then using the curve fitting property, each coefficient p is written in terms of the $PowerLimit$ value such that

$$p_i(x) = a_1 \times x^n + a_2 \times x^{n-1} + \cdots + a_{n+1} \times x^0 \tag{3.10}$$

where x is the $PowerLimit$. Finally, the p_i values in Equation 3.10 are substituted in Equation 3.9 so that $PoF(gain, GBW) = f(PowerLimit)$ is obtained. This allows the user to see the maximum achievable performance of a selected circuit topology in terms of the objective functions, for a given amount of power consumption limitation. Again, it must be noted that the PoF model can only be used for the $PowerLimit$ values between the minimum and maximum $PowerLimit$ values that the circuit has been evaluated.

3.3. Verification of the Pareto Optimal Points

On a PoF, each point represents a sized circuit showing the best trade-off between the two design objectives. The user is allowed to choose a specific PoP by clicking on it and simulate that individual in the test benches to see its design specifications and the transistors that are not operating in the saturation region, if there are any. The simulations are performed using HSPICE and the waveforms are viewed using CosmosScope. The user can also see the design variable values for that individual, which are generally the W and L values of the transistors, bias voltages, and bias currents.

3.4. Topology Selection

Topology selection is an important application of the PoFs. PoFs can be used to predict the performance of different circuit topologies for a user given design specification and the best topology that gives the optimal performance can be chosen.

In this study, two different OTA topologies which are Folded Cascode OTA and two-stage Miller OTA are analyzed. The design objectives are chosen as gain and GBW, while the design constraints are chosen as power consumption and phase margin. The user is allowed to compare the circuit performances in terms of the design objectives, for different design constraints of interest. As it will be discussed in detail in Chapter 4, an OTA topology may reveal more optimum results in terms of the design objectives, for a given range of design constraints; while it may perform worse for another range of design constraints. Therefore, once the design specifications are determined, the developed tool allows the user to select the best topology at the beginning of the design.

3.5. Axes Variable Selection

In this study, the OTAs are optimized for the deign objectives gain and GBW, and the design constraints phase margin and power consumption. However, the user is also allowed to see the power vs area plots which are obtained by the projection of power-area values of the individuals on a PoF that has been optimized for the design objectives gain and GBW.

4. CASE STUDY

In this chapter, a case study is performed for two different OTA topologies which are Folded Cascode OTA and Two-stage Miller OTA, and a comparator circuit. The OTAs are optimized for the design objectives gain and GBW, using the developed optimization algorithm. Then, PoF models are extracted for different loading and power limitation conditions. In addition to the OTAs, a comparator circuit is also optimized for the design objectives propagation delay (t_p) and power consumption. Finally, the PoFs of the Folded Cascode OTA and the comparator circuit are composed to obtain the PoF of a higher level block, that is, a comparator with a pre-amplifier which is the Folded Cascode OTA. The generated PoFs and the extracted PoF models are gathered to serve as an analog IP library.

4.1. Folded Cascode OTA

In this section, the Folded Cascode OTA shown in Figure 4.1 is analyzed.

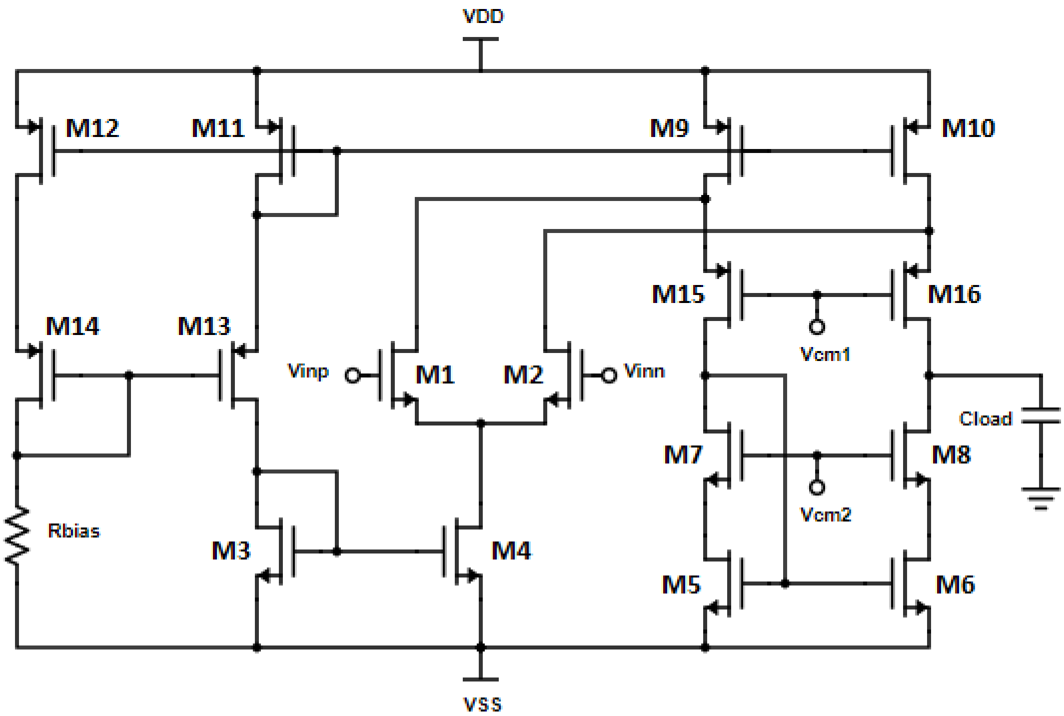

Figure 4.1. The Schematic of the Folded Cascode OTA.

Table 4.1. Lower and Upper limits of the design variables of the Folded Cascode OTA.

	$L_{1-8}[\mu m]$	$W_{1-8}[\mu m]$	$R_{bias}[k\Omega]$	$V_{cm1}[V]$	$V_{cm2}[V]$
Lower limit	0.13	0.65	1	0.2	0.4
Upper Limit	1.30	97.50	100	0.8	1.0

The circuit consists of 16 transistors and a bias resistor. The W and L values of the mirror transistors (M1-M2, M3-M4, M5-M6, M7-M8, M9-M10, M11-M12, M13-M14, and M15-M16) are chosen the same. Therefore, the circuit has 19 design variables which are 8 × W, 8 × L ,R_{bias} , V_{cm1} and V_{cm2}. The algorithm requires the lower and upper limit of the design variables as an input. A rational determination of these limits to the design variable values help the algorithm to find the Pareto-optimal front faster. The lower and upper limits of the design variables of the Folded Cascode OTA are shown in Table 4.1. The user is then asked the design objectives and the design constraints if there are any. Finally, the user provides the size of the population and the archive, N, and the maximum number of generations, T, and runs the algorithm.

4.1.1. Gain-GBW optimization of the Folded Cascode OTA

The Folded Cascode OTA is optimized for the design objectives gain and GBW. The design constraints are chosen as phase margin and power consumption such that:

$$\begin{aligned} &45^\circ < phase\ margin < 90^\circ \\ &power\ consumption < power\ limit = 500\mu W \end{aligned} \tag{4.1}$$

The population size and the maximum number of generations are set to 100 and 400, respectively. The load capacitance is taken as C_{load} = 500fF. The resulting PoF can be seen in Figure 4.2. Each point represents a sized circuit showing a best trade-off between the two design objectives. To illustrate the evolution of the PoF, Figure 4.3 shows the fronts after 25, 50, 100, 200, and 400 generations.

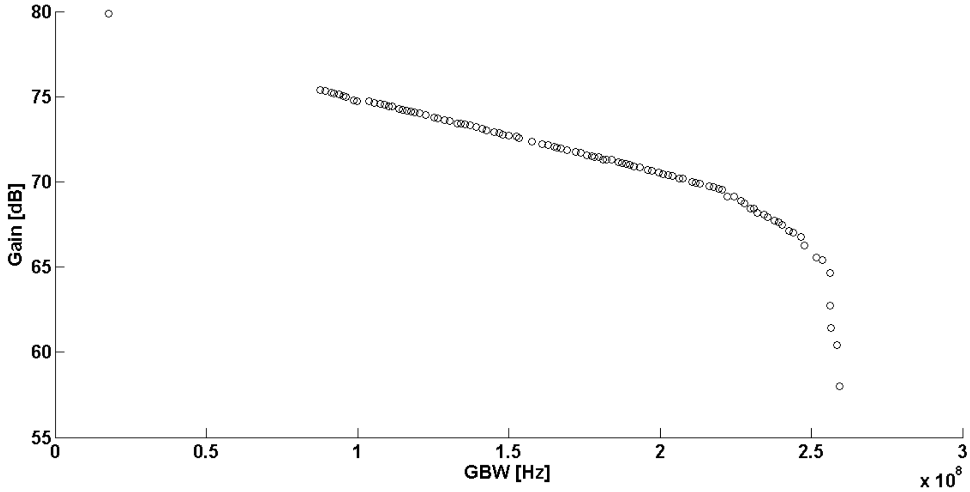

Figure 4.2. PoF of the Folded Cascode OTA where Gain and GBW have been maximized.

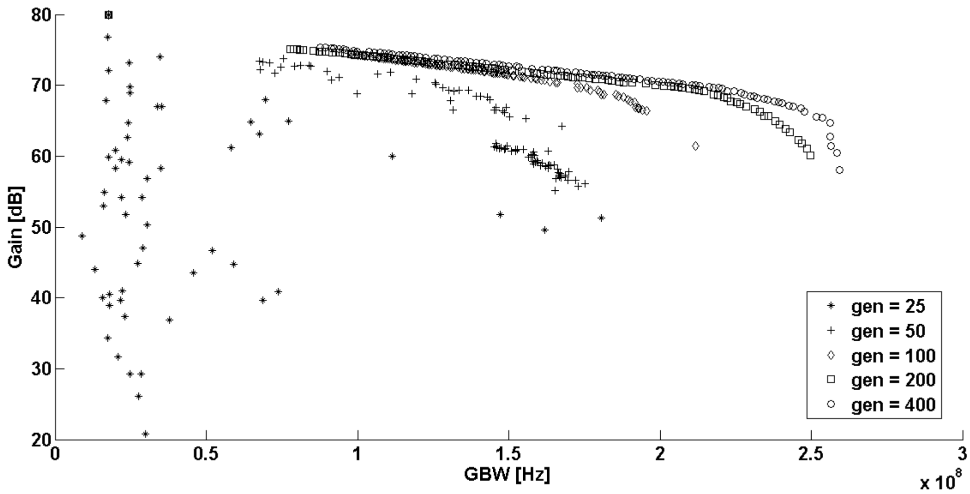

Figure 4.3. Evolution of the PoF of the Folded Cascode OTA.

4.1.2. Modeling of the Folded Cascode OTA for different power limitations

The goal of modeling the Folded Cascode OTA for different power limitations is to see the performance of the OTA for the maximum power consumption desired, without running any electrical simulations. The modeling technique makes use of the known performances under other power limitations where the circuit has been evaluated. The PoFs of the Folded Cascode OTA are obtained for $PowerLimit$ = 100μW, 200μW, 300μW, 400μW, and 500μW, as shown in Figure 4.4.

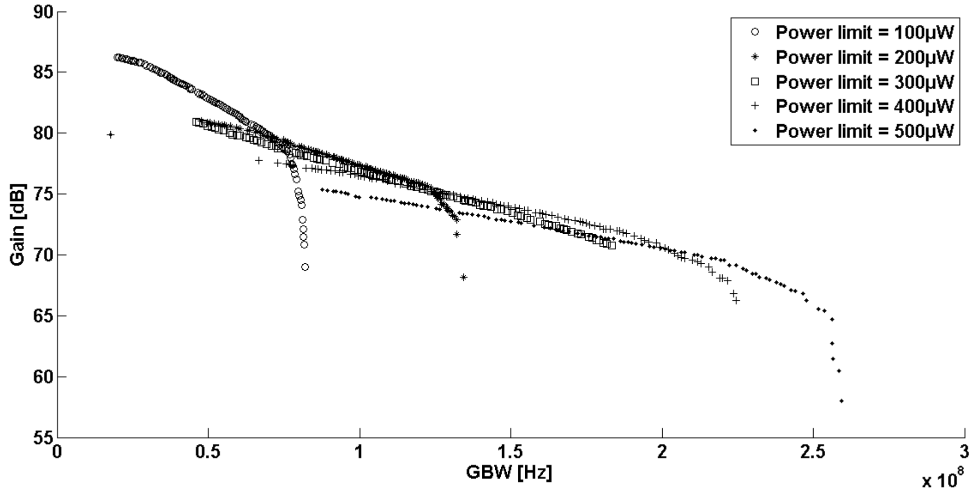

Figure 4.4. The PoFs obtained for different power constraints.

Table 4.2. Coefficients values of Equation 3.9 for each power limitation.

	p_1	p_2	p_3	p_4
100μW	-135.4	177.5	-84.47	98.04
200μW	-27.48	67.69	-60.65	98.12
300μW	-1.146	3.936	-11.34	85.45
400μW	-1.752	4.907	-8.456	81.68
500μW	-3.161	12.28	-18.52	83.9

Table 4.3. Coefficients values of Equation 3.10 for each p_i.

	a_1	a_2	a_3	a_4
p_1	6.731e+12	-7.812e+09	2.931e+06	-356.7
p_2	-3.302e+12	5.107e+09	-2.454e+06	376.2
p_3	-3.203e+12	2.067e+09	-8.233e+04	-95.21
p_4	1.562e+12	-1.312e+09	2.674e+05	83.21

Taking n=3, the coefficients of the 3^{rd} order polynomial in Equation 3.9 for each power limitation are found as shown in Table 4.2. Then, taking n=3, the coefficients of the 3^{rd} order polynomial in Equation 3.10 for each p_i are found as shown in Table 4.3. Finally, the p_i values in Equation 3.10 are substituted in Equation 3.9 so that $PoF(gain, GBW) = f(PowerLimit)$ is obtained.

The extracted PoF model is verified for power limitations of 350μW and 450μW. Figures 4.5 and 4.6 show the comparison of the extracted PoF models and the PoFs obtained by running the optimization algorithm, for these power limitation conditions.

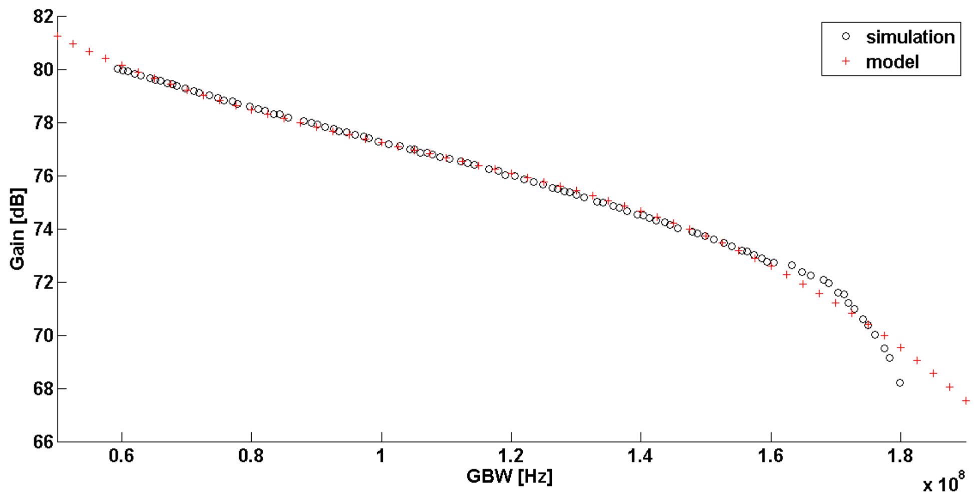

Figure 4.5. PoF for $PowerLimit$=350μW.

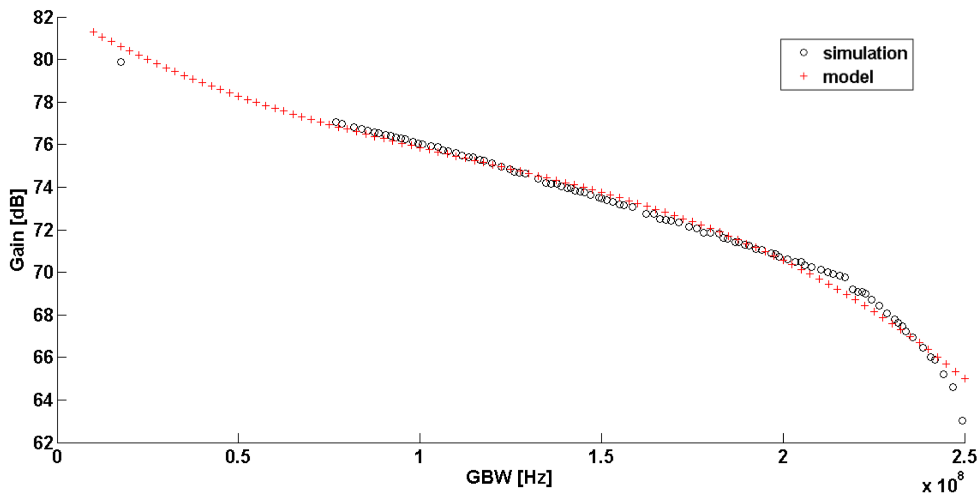

Figure 4.6. PoF for $PowerLimit$=450μW.

4.1.3. Modeling of the Folded Cascode OTA for different loading conditions

The performance of the Folded Cascode OTA changes depending on the loading conditions. However, it may be the case that the loading conditions are unknown by the time the OTA is being designed. To solve this problem, the Folded Cascode OTA is modeled for different loading conditions so that once the surrounding circuitry is known, the performance of the OTA in terms of the design objectives can be predicted, without running any electrical simulations. The modeling technique makes use of the known performances under other loading conditions where the circuit has been evaluated. The PoFs of the Folded Cascode OTA are obtained for C_{load} = 100fF, 200fF, 300fF, 400fF and 500fF, as shown in Figure 4.7.

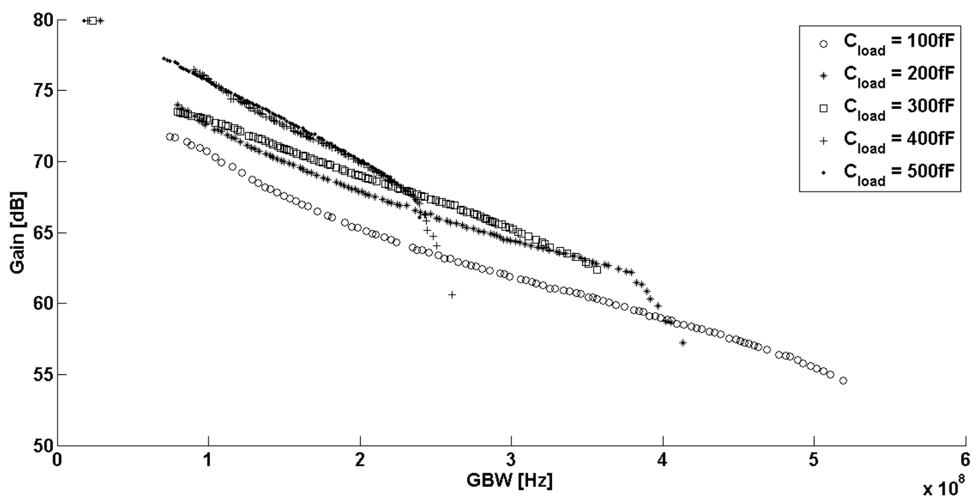

Figure 4.7. PoFs obtained for different loading conditions.

Table 4.4. Coefficients values of Equation 3.7 for each C_{load} value.

	p_1	p_2	p_3	p_4
100fF	-0.1923	2.0004	-9.7737	78.3849
200fF	-0.6194	4.6598	-14.6000	83.2967
300fF	-0.5657	3.6156	-10.9587	80.9385
400fF	-1.6350	6.1986	-12.5046	83.1950
500fF	-0.4577	1.3739	-6.5245	81.2435

Table 4.5. Coefficients values of Equation 3.8 for each p_i.

	a_1	a_2	a_3	a_4	a_5
p_1	2.072e+51	-2.34e+39	9.097e+26	-1.445e+14	7.293
p_2	-7.652e+51	8.874e+39	-3.597e+27	5.992e+14	-30.06
p_3	1.099e+52	-1.326e+40	5.634e+27	-9.75e+14	43.54
p_4	-8.628e+51	1.061e+40	-4.572e+27	8.075e+14	33.61

Taking n=3, the coefficients of the 3^{rd} order polynomial in Equation 3.7 for each C_{load} value are found as shown in Table 4.4. Then, taking n=4, the coefficients of the 4^{th} order polynomial in Equation 3.8 for each p_i are found as shown in Table 4.5. Finally, the p_i values in Equation 3.8 are substituted in Equation 3.7 so that $PoF(gain, GBW) = f(C_{load})$ is obtained.

Figure 4.8 shows the comparison of the extracted PoF model and the PoF obtained by running the optimization algorithm, for a load capacitance of 300fF.

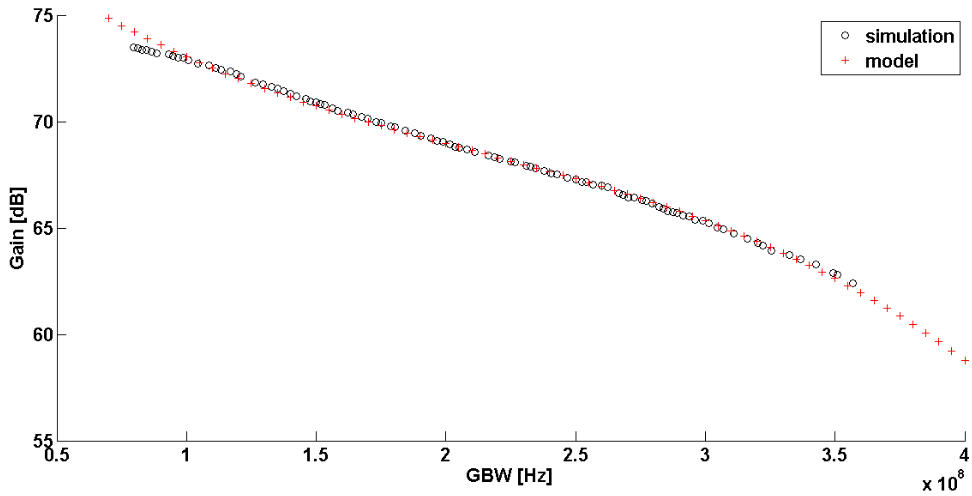

Figure 4.8. PoF for C_{load}=300fF.

4.1.4. Axes Variable Selection

In this study, the Folded Cascode OTA is optimized for the design objectives gain and GBW, and for the design constraints phase margin and power consumption. Therefore, the axes variables of the PoFs are gain and GBW. However, the user may want to compare the design solutions in terms of other design objectives such as area and power consumption. For this purpose, the user is allowed to see the area vs power plots which are obtained by the projection of area-power values of the individuals in a PoF that was optimized for the objectives gain and GBW. Figure 4.9 shows the area vs power curve that is extracted from the PoF in Figure 4.2. After eliminating the dominated solutions, the area vs power curve in Figure 4.10 is obtained.

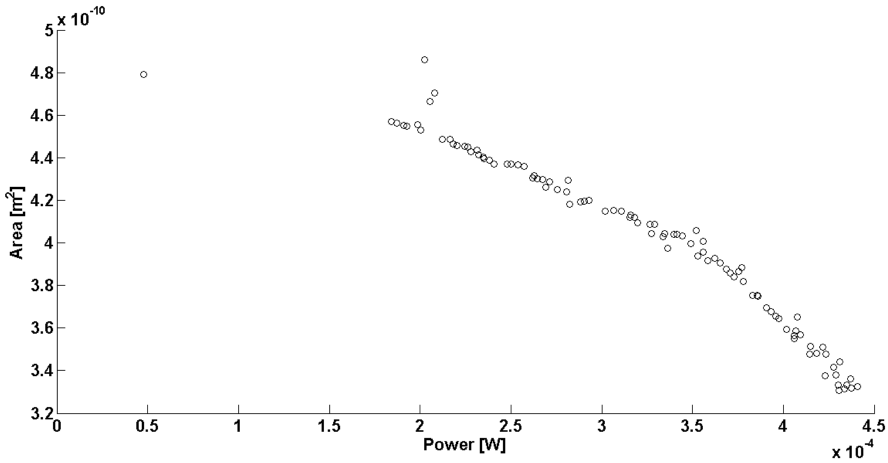

Figure 4.9. Area - Power front of the Folded Cascode OTA.

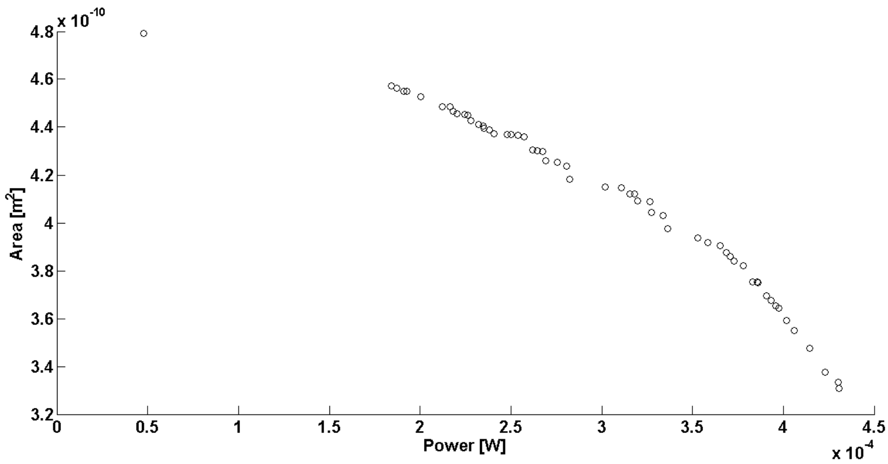

Figure 4.10. Area - Power front of the Folded Cascode OTA, after dominated solutions are eliminated.

4.2. Two-stage Miller OTA

In this section, the analysis of the Two-stage Miller OTA shown in Figure 4.11 is represented. The circuit consists of 8 transistors, a compensation capacitor and a compensation resistor. The W and L values of the mirror transistors (M1-M2 and M3-M4) are chosen the same. Therefore, the circuit has 15 design variables which are 6 × W, 6 × L, R_{com}, C_{com} and I_{bias}. The lower and upper limits of the design variables of the Two-stage Miller OTA are shown in Table 4.6.

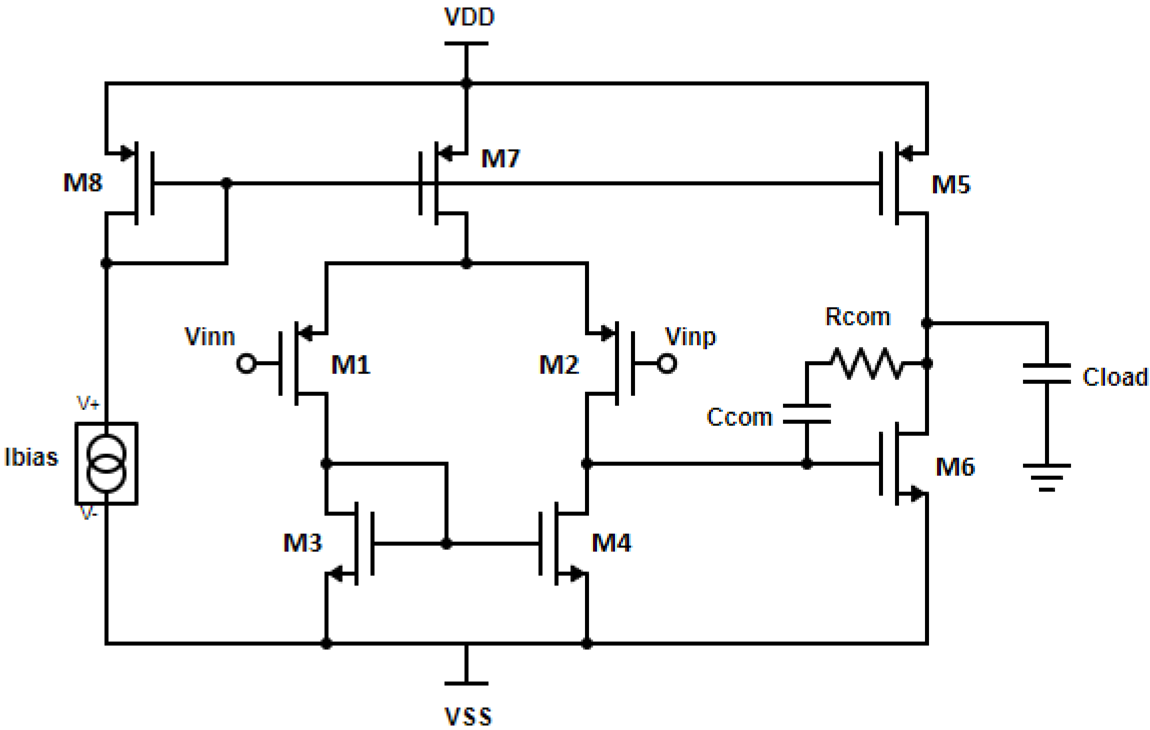

Figure 4.11. The Schematic of the Two-stage Miller OTA.

Table 4.6. Lower and Upper limits of the design variables of the Miller OTA.

	$L_{1-6}[\mu m]$	$W_{1-6}[\mu m]$	$I_{bias}[mA]$	$R_{com}[k\Omega]$	$C_{com}[pF]$
Lower limit	0.13	0.65	0.01	0.05	0.1
Upper Limit	1.30	97.50	1	5	1

4.2.1. Gain-GBW optimization of the Two-stage Miller OTA

The Two-stage Miller OTA is optimized for the design objectives gain and GBW. The design constraints are chosen as phase margin and power consumption such that:

$$\begin{aligned} &45^\circ < phase\ margin < 90^\circ \\ &power\ consumption < power\ limit = 500\mu W \end{aligned} \tag{4.2}$$

The population size and the maximum number of generations are set to 100 and 400, respectively. The load capacitance is taken as $C_{load} = 500$fF. The resulting PoF can be seen in Figure 4.12. Each point represents a sized circuit showing a best trade-off between the two design objectives. To illustrate the evolution of the PoF, Figure 4.13

shows the fronts after 25, 50, 100, 200, and 400 generations.

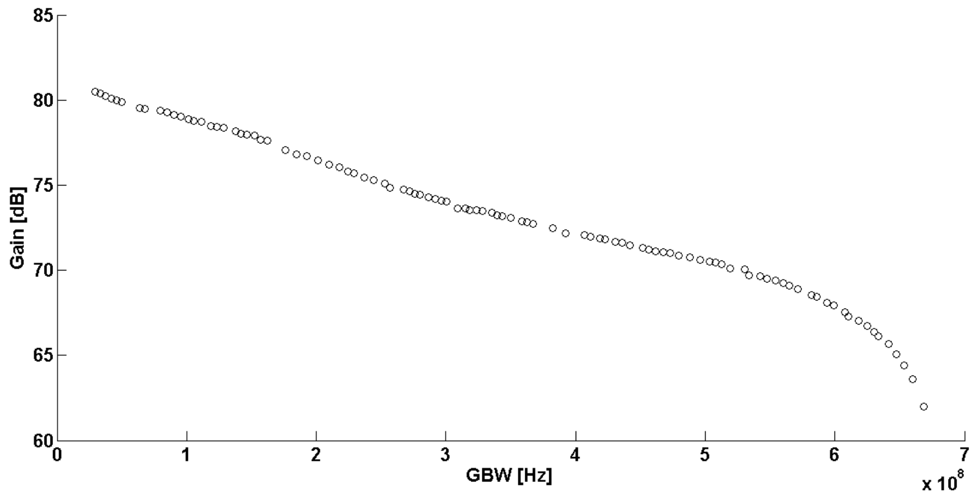

Figure 4.12. PoF of the Two-stage Miller OTA where Gain and GBW have been maximized.

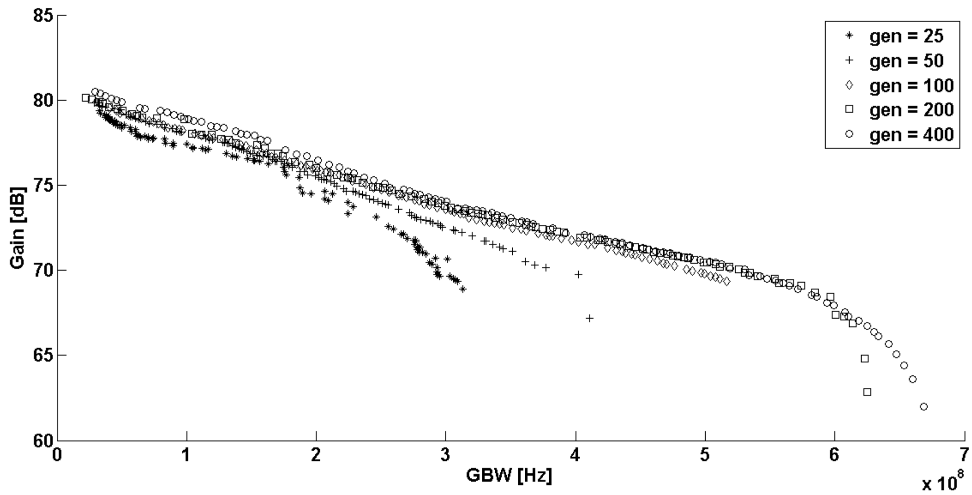

Figure 4.13. Evolution of the PoF of the Two-stage Miller OTA.

4.2.2. Modeling of the Two-stage Miller OTA for different power limitations

The goal of modeling the Two-stage Miller OTA for different power limitations is to see the performance of the OTA for the maximum power consumption desired, without running any electrical simulations. The modeling technique makes use of the known performances under other power limitations where the circuit has been evaluated. The PoFs of the Two-stage Miller OTA are obtained for $PowerLimit =$ 250μW, 350μW, 450μW, and 550μW, as shown in Figure 4.14.

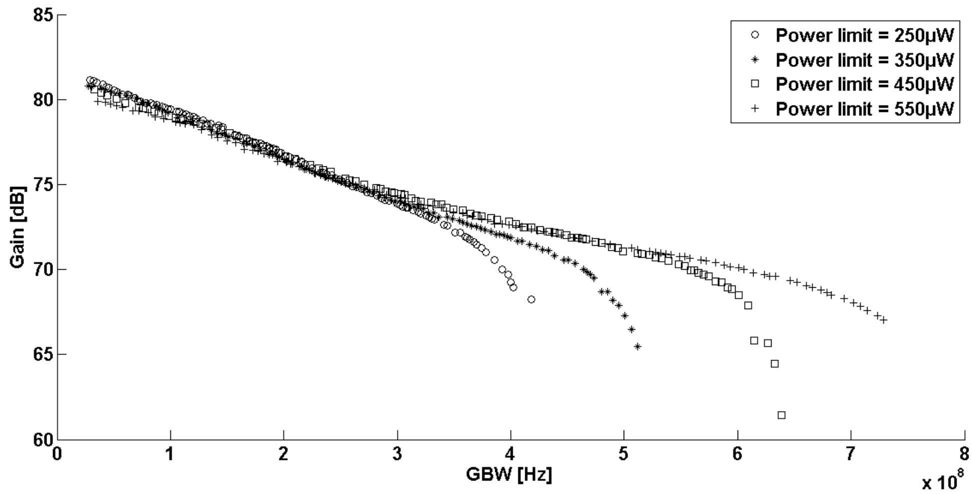

Figure 4.14. The Pareto-fronts obtained for different power constraints.

Table 4.7. Coefficients values of Equation 3.9 for each power limitation.

	p_1	p_2	p_3	p_4	p_5
250μW	-0.1232	0.9472	-2.5759	0.1654	81.0456
350μW	-0.1111	1.0666	-3.3181	1.1811	80.5196
450μW	-0.0763	0.9084	-3.4541	2.5141	79.4432
550μW	-0.0159	0.2120	-0.7923	-1.1809	80.4833

Table 4.8. Coefficients values of Equation 3.10 for each p_i.

	a_1	a_2	a_3	a_4
p_1	4.94e+08	6.151e+05	-382.5	-0.07373
p_2	-4.345e+10	3.174e+07	-6012	1.145
p_3	3.653e+11	-3.532e+08	1.05e+05	-12.45
p_4	-8.909e+11	9.513e+08	-3.179e+05	34.09
p_5	4.445e+11	-4.942e+08	1.701e+05	62.45

Taking n=4, the coefficients of the 4^{th} order polynomial in Equation 3.9 for each power limitation are found as shown in Table 4.7. Then, taking n=3, the coefficients of the 3^{rd} order polynomial in Equation 3.10 for each p_i are found as shown in Table 4.8. Finally, the p_i values in Equation 3.10 are substituted in Equation 3.9 so that $PoF(gain, GBW) = f(PowerLimit)$ is obtained.

The extracted PoF model is verified for power limitations of 300μW and 400μW. Figure 4.15 and 4.16 show the comparison of the extracted PoF models and the PoFs obtained by running the optimization algorithm.

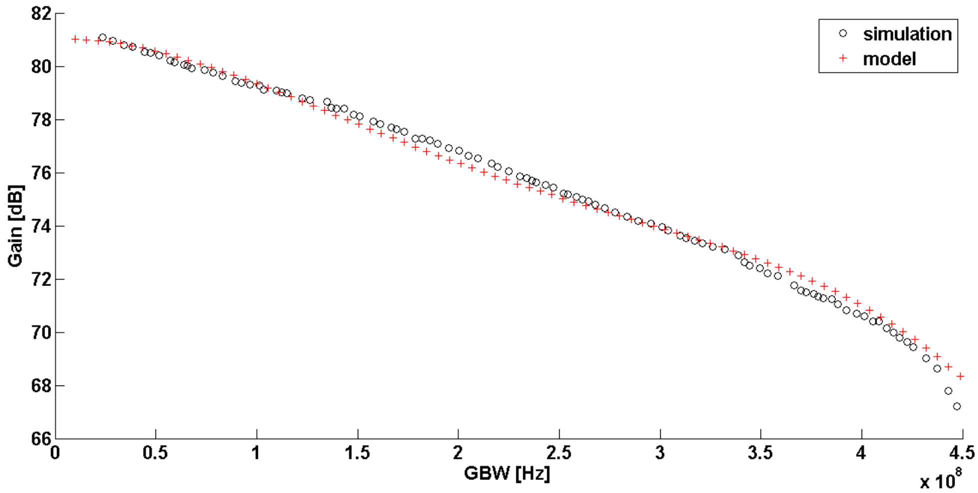

Figure 4.15. PoF for *PowerLimit*=300μW.

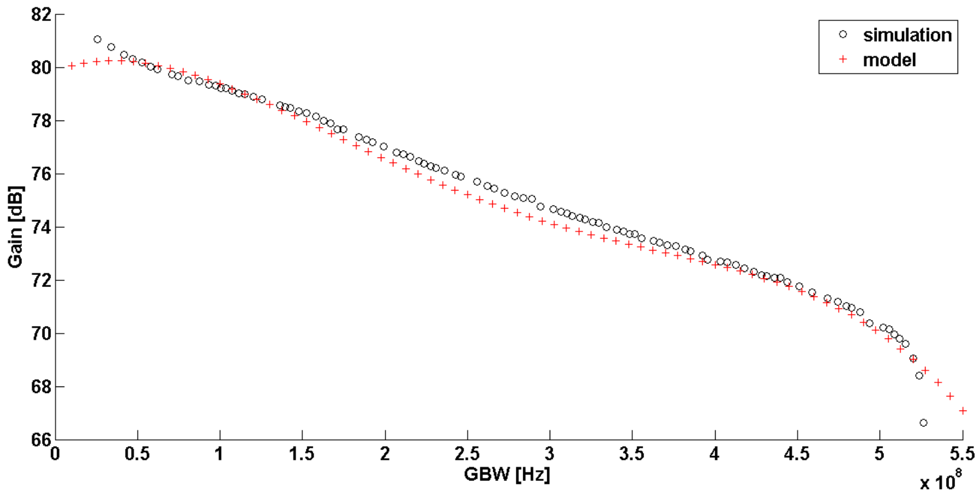

Figure 4.16. PoF for *PowerLimit*=400μW.

4.2.3. Axes Variable Selection

In this study, the Two-stage Miller OTA is optimized for the design objectives gain and GBW. However, the user is also allowed to see the area vs power plots which are obtained by the projection of area-power values of the individuals in a PoF that was optimized for the objectives gain and GBW. Figure 4.17 shows the area vs power plot that is extracted from the PoF in Figure 4.12. It must be noted that the area of the compensation capacitor determines the area of the OTA, since it is much bigger than the area of the transistors. Therefore, it can be seen that the power consumption increases as the compensation capacitance increases. The area of the compensation capacitor is calculated according to the Equation 4.3.

$$A = \frac{C.d}{k.\epsilon_0} \tag{4.3}$$

where C is capacitance, d is separation, ϵ_0 is permittivity, and k is relative permittivity.

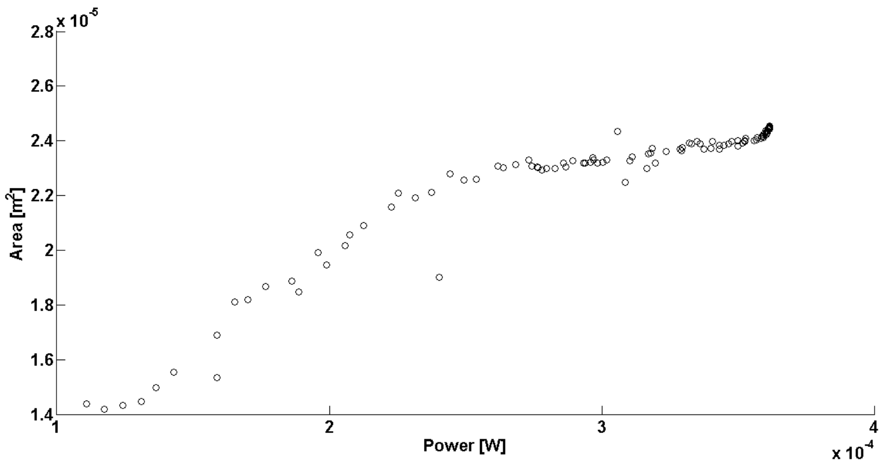

Figure 4.17. Area - Power front of the Two-stage Miller OTA.

4.3. OTA Topology Selection

In this section, an example use case of the topology selection is given. Figure 4.18 shows the PoFs of the Folded Cascode and two-stage Miller OTA for a maximum power consumption of 500μW, and for $C_{load} = 500fF$. It is seen that the two-stage Miller OTA can achieve higher gain-GBW values compared to the Folded Cascode OTA, for 500μW power consumption limitation. Figure 4.19 shows the PoFs of the Folded Cascode and the two-stage Miller OTA for a maximum power consumption of 100μW, and for $C_{load} = 500fF$. It is seen that for a lower power consumption limitation, the Folded Cascode OTA performs better than the two-stage Miller OTA. As in this example, the user can use the pre-optimized PoFs or the extracted PoF models to select the best OTA topology, once the design specifications are determined.

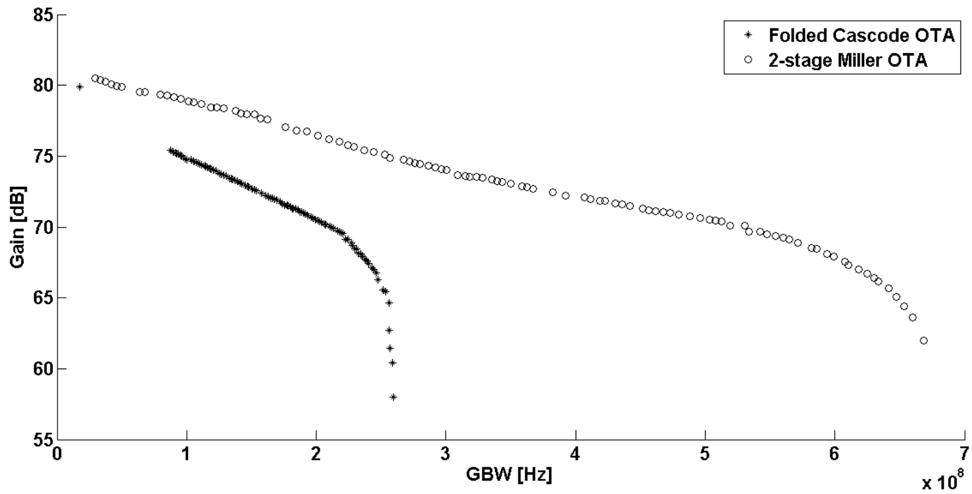

Figure 4.18. Comparison of the PoFs for power limit = 500μW.

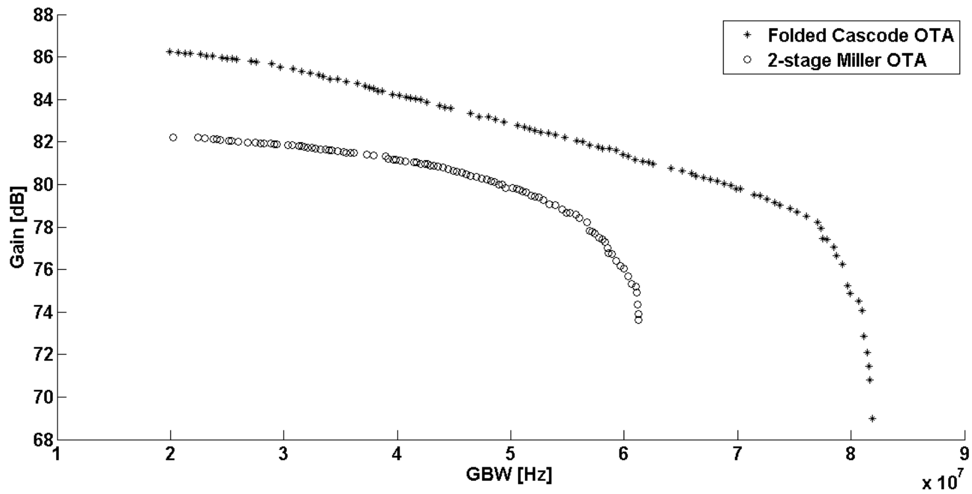

Figure 4.19. Comparison of the PoFs for power limit = 100μW.

4.4. Comparator

In this section, the optimization of the comparator circuit shown in Figure 4.20 is described. The circuit consists of 13 transistors. The W and L values of the differential pairs (M1-M2, M4-M5, M7-M8-M9-M10, and M11-M12) are chosen the same. Therefore, the circuit has 15 design variables which are 7 $\times$ W, 7 $\times$ L, and the bias voltage V_{bias}. The lower and upper limits of the design variables of the comparator are shown in Table 4.9.

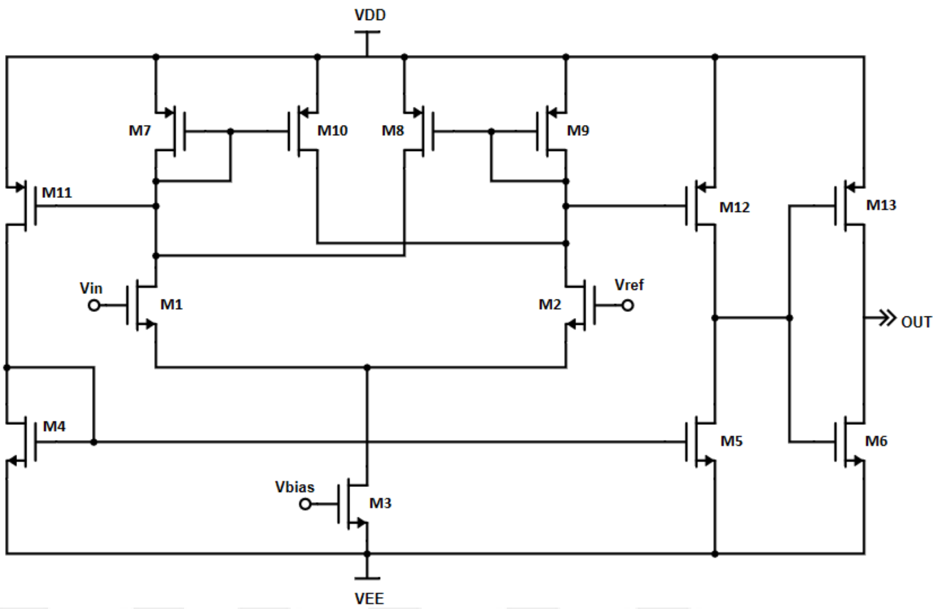

Figure 4.20. The Schematic of the Comparator Circuit.

Table 4.9. Lower and Upper limits of the design variables of the comparator.

	$L_{1-7}[\mu m]$	$W_{1-7}[\mu m]$	$V_{bias}[V]$
Lower limit	0.13	0.65	0.2
Upper Limit	1.30	97.50	0.6

4.4.1. Propagation delay (t_p) - Power optimization of the Comparator

The design objectives of the comparator are chosen as the propagation delay and power consumption. Propagation delay of a comparator is defined as the time difference between the moment the input signal crosses the reference voltage and the moment the output state changes (usually when the output signal crosses the mid-value between V_{OH} and V_{OL}, if nothing is specified). It is an important parameter for many applications because it limits the maximal input frequency that can be processed. The propagation delay time of the comparators varies as a function of the amplitude of the input signal. A larger input results in a smaller propagation delay time. However, there is an upper limit at which a further increase in the input voltage will no longer effect the delay, which is called slewing.

The following method is used to find the propagation delay of the comparator. First, the input offset voltage of the comparator is found by sweeping the DC level of V_{in} accross the reference voltage, $V_{ref} = 0.6$V, and recording the V_{in} voltage where the output state changes. Figure 4.21 shows the simulation result of a comparator, whose input offset voltage is found as 0.5998756V - 0.6V = -124.4μV. To eliminate the input offset voltage, the negative of the input offset voltage is applied at the negative input of the comparator (e.g. $V_{ref} = 0.6\text{V} + 124.4\mu\text{V} = 0.6001244\text{V}$). Then a pulse of amplitude 1mVpp is applied to V_{in}, and the time difference between the moment the input signal crosses 0.6V and the moment the output signal crosses 0.6V gives the propagation delay. Figure 4.22 shows the simulation result of the comparator, whose propagation delay is found as

$$t_p = \frac{t_{p_h} + t_{p_l}}{2} = \frac{1.409363 \times 10^{-8} + 1.197017 \times 10^{-8}}{2} = 1.303190 \times 10^{-8} \tag{4.4}$$

The generated PoF of the comparator is shown in Figure 4.23.

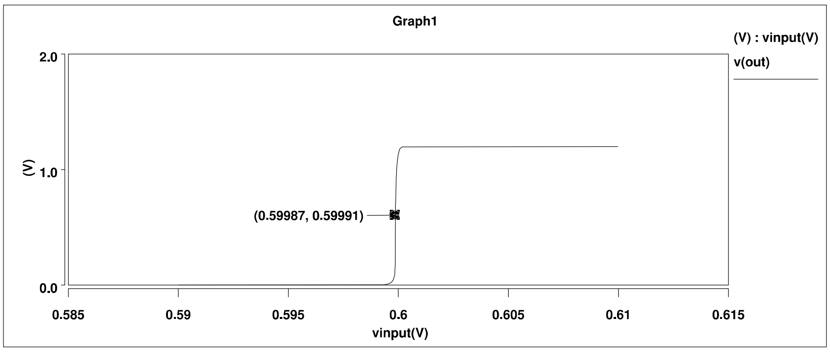

Figure 4.21. Input offset voltage simulation result of the comparator.

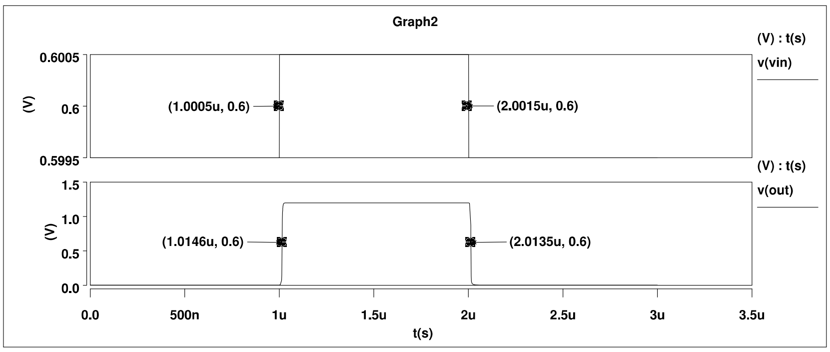

Figure 4.22. Propagation delay simulation result of the comparator.

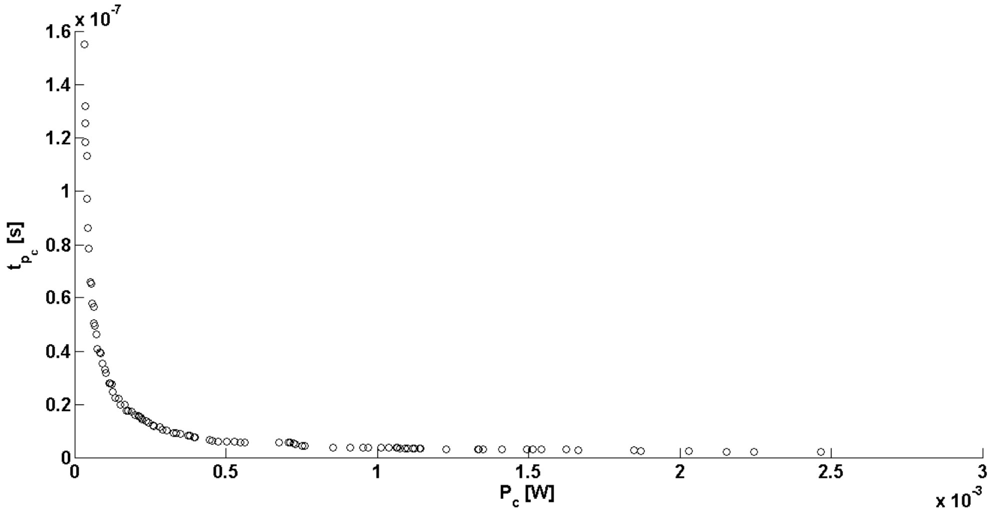

Figure 4.23. Propagation delay - power consumption front of the comparator.

4.5. Composition of the Pareto-optimal Fronts

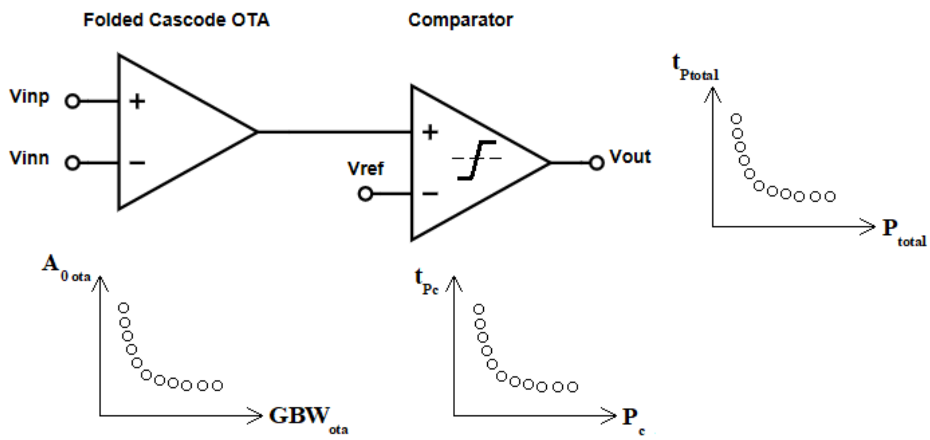

Figure 4.24. Composition of the PoFs.

In Section 4.1-4.4, generation of PoFs for the Folded Cascode OTA, Two-stage Miller OTA and the comparator has been described. As the next step, these PoFs are used to generate the PoFs of a higher-level block which is shown in Figure 4.24. The t_p - Power front of the comparator and the gain - GBW front of the Folded Cascode OTA are used to generate the t_p - Power front of the overall system, following the methodology described below. It is observed that the sensitivity of the comparators in Figure 4.23 is in the order of 0.1mV. Therefore, for input voltages below 0.1mV, the Folded Cascode OTA should be used to amplify the input voltage. However, for input voltages above 0.1mV, using the comparator without the pre-amplifier is more efficient;

because even if the propagation delay of the comparator decreases with larger input voltages, the overall propagation delay increases because of the high delay contribution of the amplifier.

The t_p - Power front of the OTA-comparator circuit is obtained for input voltages $V_{in} = 1\mu$V and $V_{in} = 0.1$mV. In Section 4.5.1, the OTA-comparator circuit is analyzed for $V_{in} = 1\mu$V, and in Section 4.5.2, the OTA-coparator circuit is analyzed for $V_{in} = 0.1$mV.

4.5.1. Analysis for $V_{in} = 1\mu$V

4.5.1.1. Generation of the new PoF using the interpolation method.

- The input capacitance of each comparator circuit in Figure 4.23 are calculated according to the Equation (4.5). The obtained C_{in} values constitute the load capacitance values of the Folded Cascode OTA circuits.

$$C_{in} = C_{ox}WL \tag{4.5}$$

  where $C_{ox} = 14.8 \times 10^{-3}\frac{F}{m^2}$ for 130nm NMOS technology.
- During the composition process, the PoF of the Folded Cascode OTA that has been obtained for the objectives gain-GBW, and a load capacitance C_{load}=500fF (shown in Figure 4.2) is used. Since the GBW of an OTA is dependent on its load capacitance, the new GBW values of each OTA are obtained for the load capacitance values calculated in Equation (4.5). This is done by simulating each OTA for three different load capacitance values (i.e. three different comparators) and then doing interpolation to obtain the GBW values of each OTA for all comparators. The $GBW - C_{load}$ plot of an OTA obtained by interpolating the three simulation results is shown in Figure 4.25. The interpolation is done by fitting a second order polynomial (i.e. $f(x) = p_1x^2 + p_2x + p_3$) to the three $GBW - C_{load}$ points that are obtained by simulation.

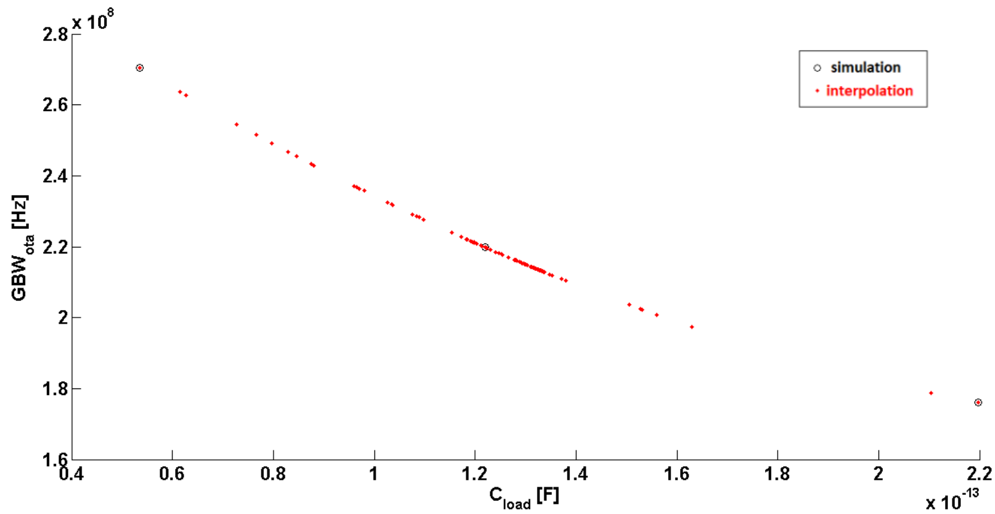

Figure 4.25. $GBW - C_{load}$ plot of an OTA. Interpolation technique is used to obtain the GBW of the OTA for all C_{load} values.

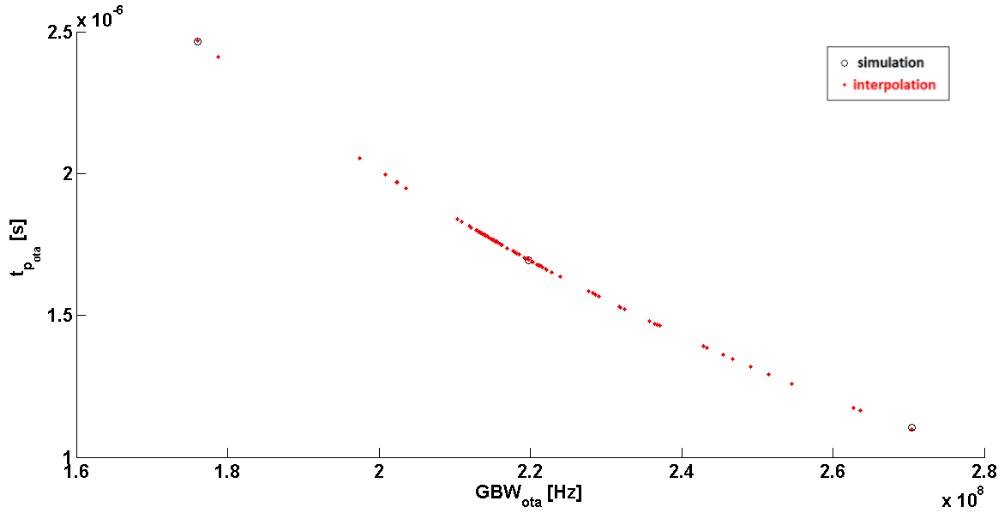

Figure 4.26. $t_{p_{OTA}}$ - GBW_{OTA} plot of an OTA. Interpolation technique is used to obtain the $t_{p_{OTA}}$ of the OTA for all GBW_{OTA} values.

- As the GBW of an OTA changes with the load capacitance, its propagation delay also changes. While simulating the OTAs for three different load capacitance values, the propagation delay of the OTAs are also simulated and then interpolation is done to obtain the $t_{p_{OTA}}$ - GBW_{OTA} plots of each OTA, using the GBW values found at the previous step. Figure 4.26 shows the $t_{p_{OTA}}$ - GBW_{OTA} plot obtained for the OTA in Figure 4.25. The interpolation is done by fitting a first order exponential curve (i.e. $f(x) = a.e^{bx}$) to the three $t_{p_{OTA}}$ - GBW_{OTA} points that are obtained by simulation. This interpolation technique is applied to all OTAs and the $t_{p_{OTA}}$ - GBW_{OTA} plot of all OTAs for all load capacitance values (i.e. for all comparators) is obtained as shown in Figure 4.27.

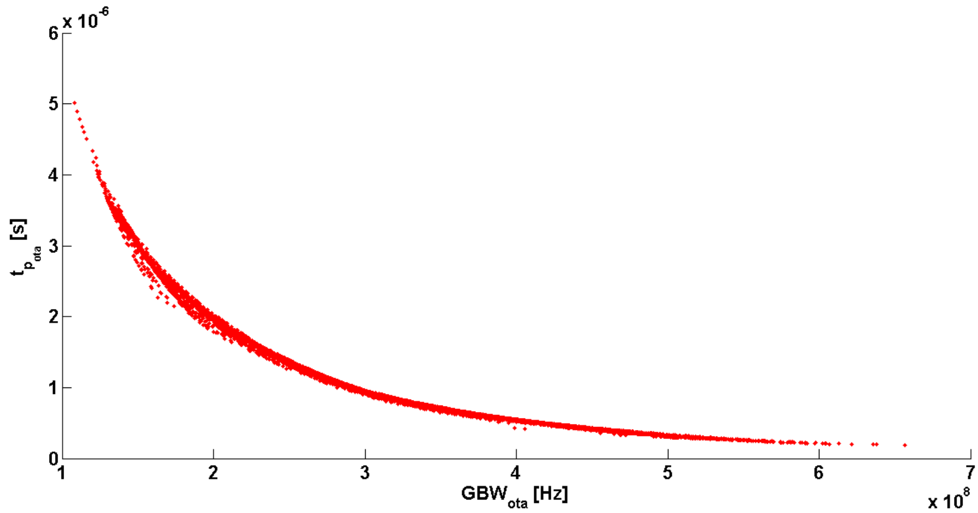

Figure 4.27. $t_{p_{OTA}}$ - GBW_{OTA} plot of all OTAs, obtained by interpolation.

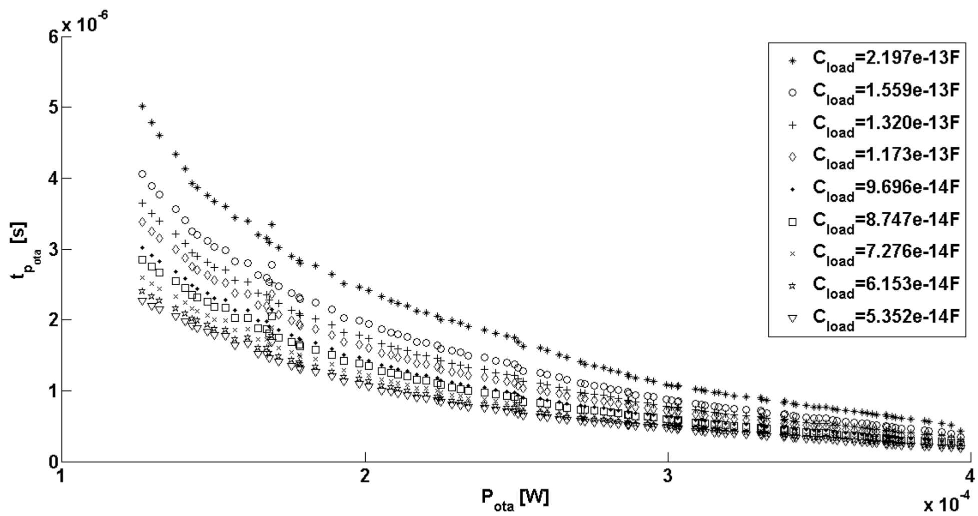

Figure 4.28. $t_{p_{OTA}}$ - P_{OTA} plot of all OTAs for different comparators (for different C_{load} values).

- Now that the propagation delay of all OTAs for all comparators are known, the $t_{p_{OTA}}$ - P_{OTA} plot of all OTAs for all load capacitance values (i.e. for all comparators) is plotted as shown in Figure 4.28. It can be seen that the OTAs with a smaller propagation delay have a larger GBW, and therefore have a higher power consumption.

- The comparator propagation delay is not only dependent on the comparator itself, but also the amplitude of the input voltage. Therefore, the propagation delay of a comparator changes depending on the OTA it is connected to. Using the interpolation method, the propagation delay of the comparators are found for the gains of each OTA, as shown in Figure 4.29. Interpolation is done by fitting a second order polynomial curve (i.e $f(x) = p_1x^2 + p_2x + p_3$) to the three t_{p_c} - $Gain_{OTA}$ points that are found by simulation. Then the t_{p_c} - P_c plots of each comparator are obtained for all OTAs, as shown in Figure 4.30. It must be noted that the input voltage of the OTA is taken as 1μV, which results in a comparator input voltage of $1\mu\text{V}\times Gain_{OTA}$.

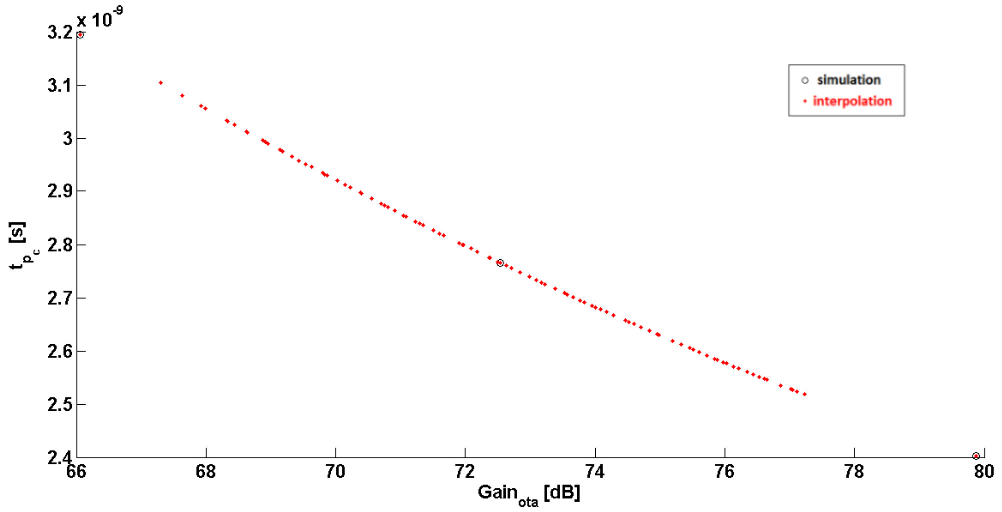

Figure 4.29. t_{p_c} - $Gain_{OTA}$ plot of a comparator. Interpolation technique is used to obtain the t_{p_c} value of the comparator for each OTA.

- Finally, the t_p - Power front of the overall system is obtained by summing the obtained t_p and power consumption values of the OTA and comparator, as shown in Equation (4.6). Figure 4.31 shows the PoF of the overall system, obtained after eliminating the dominated solutions in the $t_{P_{total}}$ - P_{total} plot.

$$\begin{aligned} t_{P_{total}} &= t_{p_c} + t_{P_{OTA}} \\ P_{total} &= P_c + P_{OTA} \end{aligned} \tag{4.6}$$

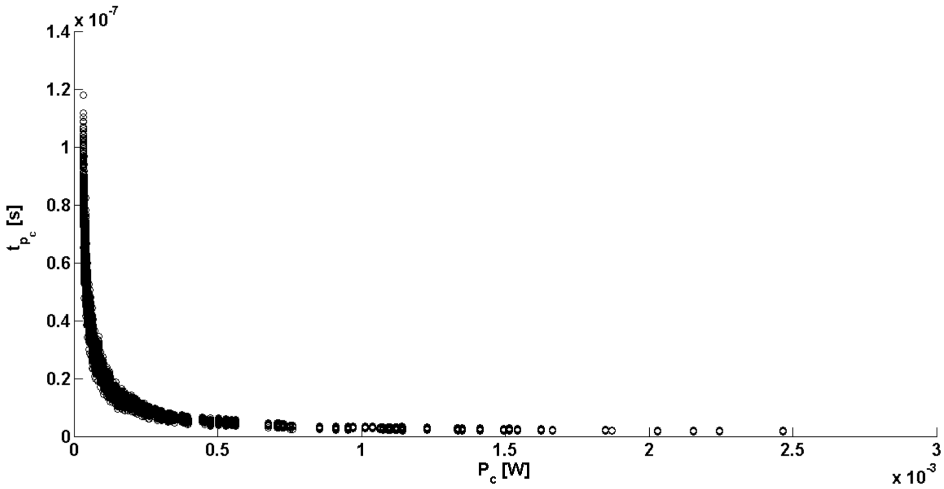

Figure 4.30. t_{p_c} - P_c plot of all comparators ($V_{in} = 1\mu V$).

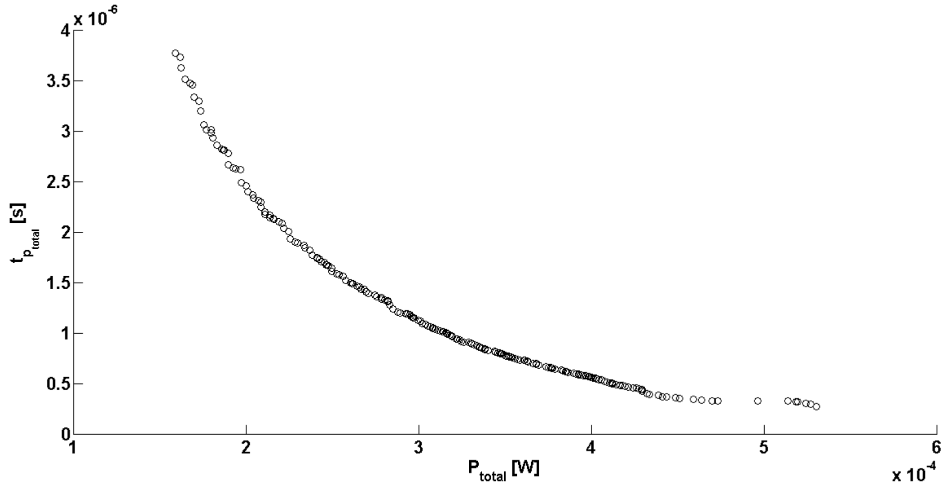

Figure 4.31. PoF of the overall system, obtained by interpolation ($V_{in} = 1\mu V$).

4.5.1.2. Verification of the new PoF by simulation. In this section, the PoF of the OTA-comparator circuit that is obtained by using the interpolation method is compared with the PoF that is obtained by simulation.

- The $t_{p_{OTA}}$ - P_{OTA} plot that is obtained by interpolation method (shown in Figure 4.28) is re-generated by simulating all OTAs for all C_{in} values of the comparators. Figure 4.32 shows the $t_{p_{OTA}}$ - P_{OTA} plot that is obtained by simulation.
- The t_{p_c} - P_c plot that is obtained by interpolation method (shown in Figure 4.30) is re-generated by simulating all comparator circuits for all $Gain_{OTA}$ values of the OTAs. Figure 4.33 shows the t_{p_c} - P_c plot that is obtained by simulation.

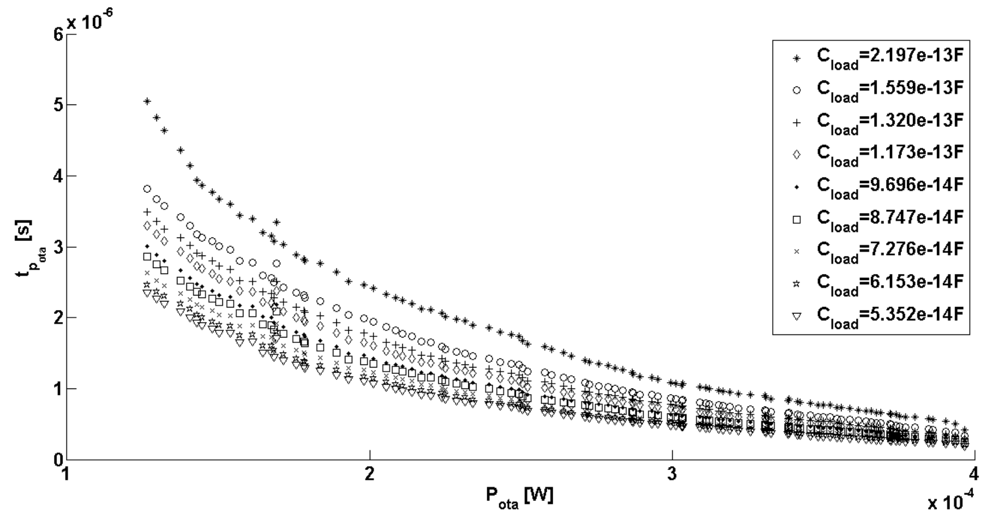

Figure 4.32. $t_{p_{OTA}}$ - P_{OTA} plot of all OTAs for different comparators (for different C_{load} values), obtained by simulation.

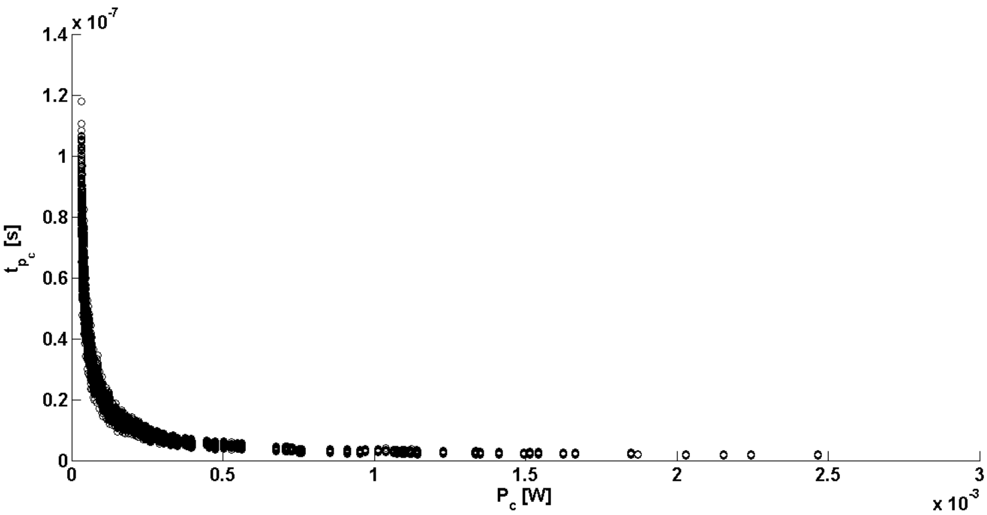

Figure 4.33. t_{p_c} - P_c plot of all comparators, obtained by simulation ($V_{in} = 1\mu V$).

- Finally, the t_p and power values of the OTAs and comparators that are found by simulation are summed up and the dominated individuals are eliminated to obtain the PoF of the overall system, as shown in Figure 4.34.
- Figure 4.35 shows the PoFs that are obtained by interpolation and simulation on one figure. It can be seen in Figure 4.36 that the maximum error introduced by the interpolation method is 4.5%. The error is defined as shown in Equation 4.7.

$$error[\%] = 100 \times \frac{simulation - interpolation}{simulation} \tag{4.7}$$

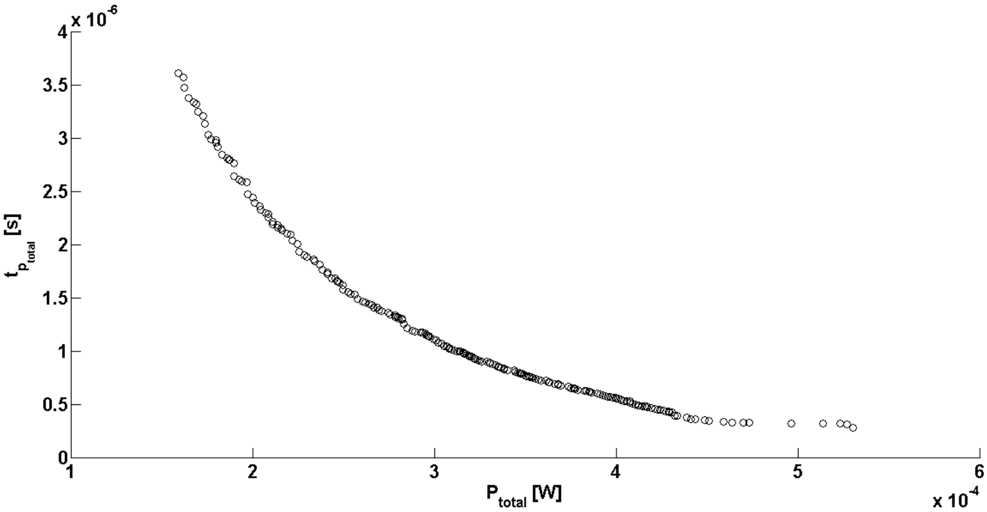

Figure 4.34. PoF of the overall system, obtained by simulation ($V_{in} = 1\mu V$).

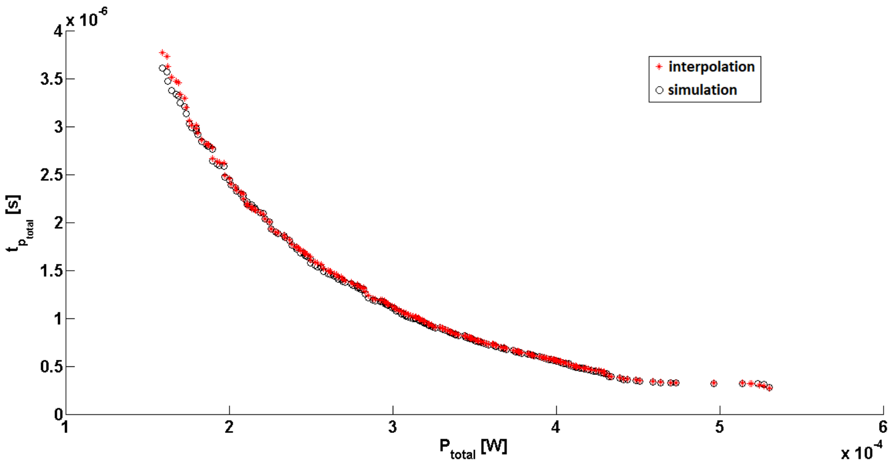

Figure 4.35. Comparison of the PoFs ($V_{in} = 1\mu V$).

4.5.2. Analysis for V_{in} = 0.1mV

In this section, the PoF of the OTA-comparator circuit is obtained for the minimum input voltage that the comparators are sensitive to, which is V_{in} = 0.1mV. The resulting PoF is compared with the PoF that is obtained by simulating the comparators in Figure 4.23 for V_{in} = 0.1mV. It is seen that using the comparator without the pre-amplifier results in a more efficient performance in terms of the design objectives; since the delay contribution of the OTA is much higher than the decrease in the delay of the comparator as a result of the amplified input voltage.

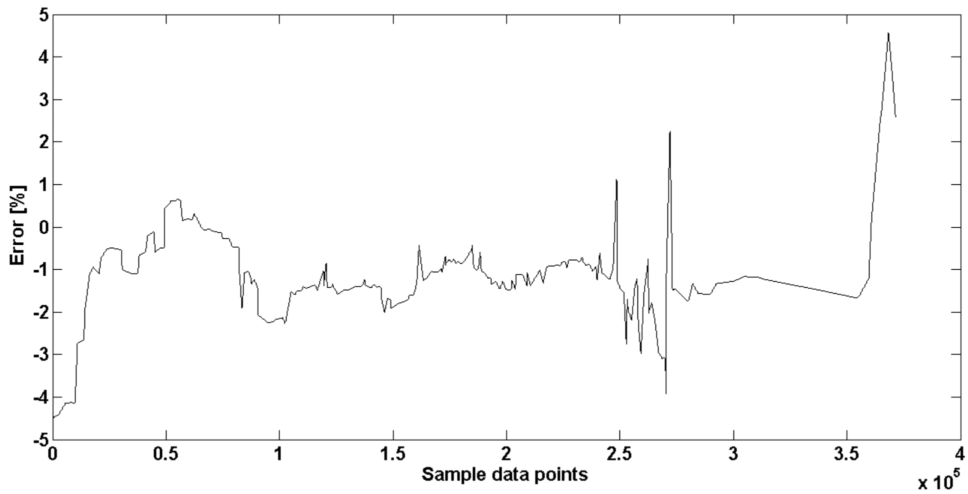

Figure 4.36. Error introduced by the interpolation method ($V_{in} = 1\mu V$).

4.5.2.1. PoF comparison of the comparator with and without using the OTA.

- Using the interpolation method described previously, the t_{p_c} - P_c plot is obtained for V_{in} = 0.1mV, as shown in Figure 4.37.

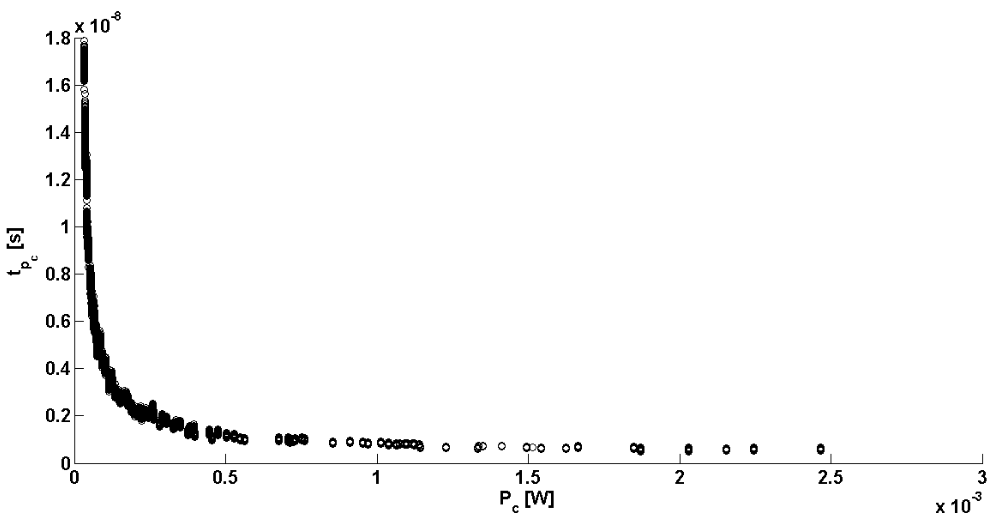

Figure 4.37. t_{p_c} - P_c plot of all comparators ($V_{in} = 0.1mV$).

- Then, the t_p - Power front of the overall system is obtained by summing the obtained t_p and power consumption values of the OTA and comparator, and eliminating the dominated solutions, as shown in Figure 4.38.
- Finally, the comparators in Figure 4.23 are simulated for V_{in} = 0.1mV and the PoF shown in Figure 4.39 is obtained. It can be seen that the PoF of the comparator

is much more optimal than the PoF of the OTA-comparator circuit.

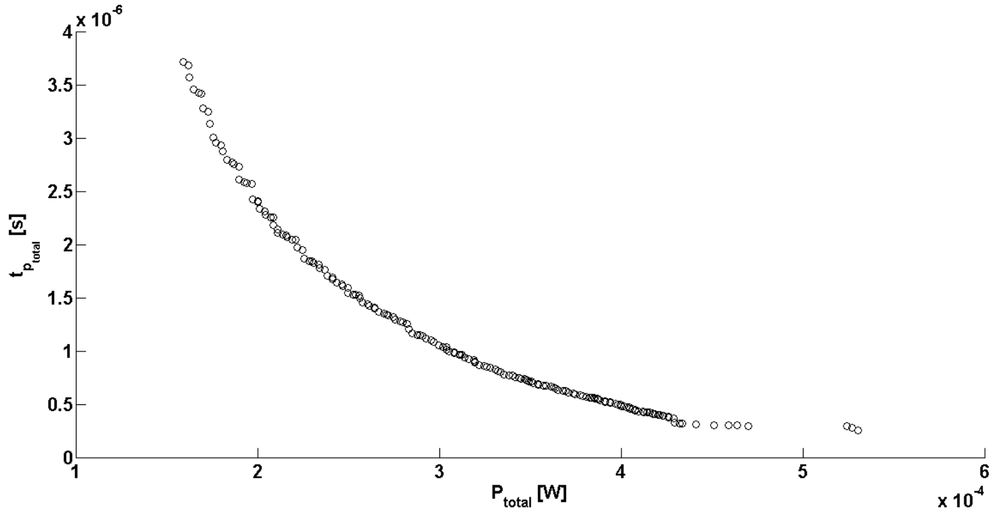

Figure 4.38. $t_{p_{total}}$ - P_{total} plot ($V_{in} = 0.1mV$).

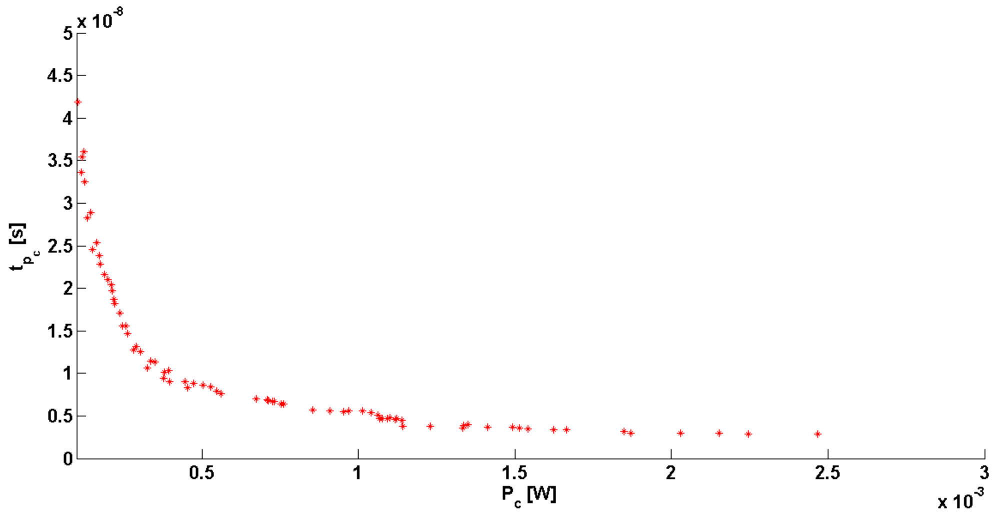

Figure 4.39. t_{p_c} - P_c plot of the comparators ($V_{in} = 0.1mV$).

5. CONCLUSION

Analog design optimization has been an active research area for many years. In the past couple of decades, most approaches to analog design automation have been built around optimization algorithms. Several multi-objective optimization algorithms have been reported and successfully applied to design automation problems. SPEA2 is one of the reported multi-objective optimization algorithms which has been proved to be successful in terms of converging near the ideal PoF and preserving the boundary solutions. In this thesis, a multi-objective optimization algorithm which adopts the SPEA2 approach is implemented and applied to three different circuit topologies, which are Folded Cascode OTA, Two-stage Miller OTA, and comparator. The OTAs are optimized for the design objectives gain and GBW, and the comparator is optimized for the design objectives propagation delay and power consumption. The PoF models of the OTAs are extracted for different loading and power limitation conditions. The models are verified by comparing the PoF models with the PoFs that are generated by running the optimization algorithm under the conditions the PoF models have been created. An analog library serving as analog IP is constructed, which includes the pre-optimized PoFs and the extracted PoF models for different loading and power limitation conditions. Finally, the PoFs that has been obtained for the Folded Cascode OTA and the comparator are composed to obtain the PoF of a higher level block, that is, the Folded Cascode OTA driving the comparator. The PoF of this higher level block is obtained by doing interpolation, and then it is verified by simulations. It is confirmed that the individuals on the resulting PoF make use of different OTAs and comparators.

A graphical user interface (GUI) is implemented, that allows running all the applications of the developed tool. The developed tool also allows the user to verify the Pareto-optimal points in the test benches, and provides visualization of the design variables, using the GUI.

It is a well-known fact that despite the availability of improved design automation tools, it is still hard to re-synthesis a circuit every time the design requirements change. Therefore, the developed tool aims to save the designers a lot of time.

REFERENCES

1. Islam, M. M., F. Al Mahmud, M. H. Maruf and M. A. Islam, “hebele Survey on Electronic Design Automation Tools And Software”, *SEU Journal of Science and Engineering*, Vol. 11, No. 1, 2017.

2. Rutenbar, R. A., “Analog design automation: Where are we? Where are we going?”, *Custom Integrated Circuits Conference, 1993., Proceedings of the IEEE 1993*, pp. 13–1, IEEE, 1993.

3. Saleh, R., S. Wilton, S. Mirabbasi, A. Hu, M. Greenstreet, G. Lemieux, P. P. Pande, C. Grecu and A. Ivanov, “System-on-chip: Reuse and integration”, *Proceedings of the IEEE*, Vol. 94, No. 6, pp. 1050–1069, 2006.

4. Jacome, M. F. and H. P. Peixoto, “A survey of digital design reuse”, *IEEE Design & Test of Computers*, Vol. 18, No. 3, pp. 98–107, 2001.

5. Brunoli, M., M. Hotta, F. James, R. Koch, R. McGuffin and A. J. Moore, “Analog intellectual property: now? Or never?”, *Proceedings of the 39th annual Design Automation Conference*, pp. 181–182, ACM, 2002.

6. Modi, S., S. Askari, S. Manohar, P. Balsara and M. Nourani, “Automated G m C filter design: A case study in accelerated reuse of analog circuit design”, *Circuits and Systems Workshop (DCAS), 2010 IEEE Dallas*, pp. 1–4, IEEE, 2010.

7. Coello, C. C., “Evolutionary multi-objective optimization: a historical view of the field”, *IEEE Computational Intelligence Magazine*, Vol. 1, No. 1, p. 28–36, 2006.

8. Edgeworth, F. Y., *Mathematical physics: an essay on the application of mathematics to the moral sciences*, London School of Economics and Political Science, 1932.

9. Pareto, V., *Cours deconomie politique professe a lUniversite de Lausanne*, Vol. 2, Rouge, 1896.

10. Goldberg, D. E., *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley, 2012.

11. Zitzler, E., K. Deb and L. Thiele, *Comparison of multiobjective evolutionary algorithms: empirical results*, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology Zurich (ETH), 1999.

12. Deb, K., *Multi-objective optimization using evolutionary algorithms*, John Wiley & Sons, 2008.

13. Deb, K., A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, p. 182–197, 2002.

14. Srinivas, N. and K. Deb, "Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms", *Evolutionary Computation*, Vol. 2, No. 3, p. 221–248, 1994.

15. Knowles, J. D. and D. W. Corne, "Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy", *Evolutionary Computation*, Vol. 8, No. 2, p. 149–172, 2000.

16. Konak, A., D. W. Coit and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial", *Reliability Engineering & System Safety*, Vol. 91, No. 9, p. 992–1007, 2006.

17. Silverman, B. W., *Density estimation for statistics and data analysis*, Chapman & Hall/CRC, 1998.

18. Kim, M., T. Hiroyasu, M. Miki and S. Watanabe, "SPEA2+: Improving the Perfor-

mance of the Strength Pareto Evolutionary Algorithm 2", *Lecture Notes in Computer Science Parallel Problem Solving from Nature - PPSN VIII*, p. 742–751, 2004.

19. Sonmez, O. S. and G. Dundar, "Simulation-based analog and RF circuit synthesis using a modified evolutionary strategies algorithm", *Integration, the VLSI Journal*, Vol. 44, No. 2, p. 144–154, 2011.