Trade-Off Design of Analog Circuits using Goal Attainment and "Wave Front" Sequential Quadratic Programming

Daniel Mueller, Helmut Graeb, Ulf Schlichtmann
Institute for Electronic Design Automation, TU Muenchen

Abstract

One of the main tasks in analog design is the sizing of the circuit parameters, such as transistor lengths and widths, in order to obtain optimal circuit performances, such as high gain or low power consumption. In most cases one performance can only be optimized at cost of others, therefore a sizing must aim at an optimal trade-off between the important circuit performances.

In this paper we present a new deterministic method to calculate the complete range of performance trade-offs, the so-called Pareto-optimal front, of a given circuit topology. Known deterministic methods solve a set of constrained multi-objective optimization problems independently of each other. The presented method minimizes a set of Goal Attainment (GA) optimization problems simultaneously. In a parallel algorithm, the individual GA optimization processes compare and exchange their iterative solutions. This leads to a significant improvement in the efficiency and quality of analog trade-off design.

1. Introduction

Analog components play an important role for integrated circuits (ICs). Interfaces to the analog world and important functions like clock generation and power management include analog circuit blocks. Often they are realized together with digital components on a single system on a chip (SoC). In EDA Café Weekly of 21 March 2005 it is approximated that in 2006 75% of all ICs include analog components. The design of analog systems is usually done hierarchically through system partitioning into circuit blocks and top-down specification propagation [1,2,11].

Time-to-market requirements often allow only the generation of a single analog topology and sizing, which represents a single trade-off of a complete range of possible trade-offs of the circuit performances. However, knowledge of the complete range of performance capabilities is required for modern hierarchical analog design processes.

Various approaches have been presented for the systematic generation of the complete range of trade-offs of the main circuit performances, the so called Pareto-optimal front. Most current Pareto-optimal front generators are based on finding a set of Pareto-optimal performance vectors, which are evenly spread across the front. The complete Pareto-optimal front is approximated by interpolation of these performance vectors.

In [7,10,13,19] stochastic methods are applied to generate the Pareto-optimal front. These algorithms commonly require a large amount of simulations, which can be processed in parallel on multiple processors. The proposed genetic and evolutionary algorithms (GEAs) work with a complete set of parameter vectors, called population. By adding or changing members of the population, GEAs try to improve the complete range of possible performance trade-offs simultaneously. Mechanisms are required for these algorithms to assure that the final population produces solutions along the complete Pareto-optimal front.

In [4] the Normal–Boundary-Intersection (NBI) approach was introduced as a deterministic method to generate the Pareto-optimal front of an analog block. It formulates a set of optimization problems whose solutions spread almost evenly along the Pareto-optimal front. A Sequential Quadratic Programming (SQP) algorithm is used to minimize each of these optimization problems independently. SQP is one of the most efficient algorithms for constrained nonlinear optimization in terms of fast convergence.

In this paper a new method for the generation of the Pareto-optimal front is proposed. It minimizes a set of Goal Attainment (GA) optimization problems. The GA approach achieves the same almost even spread of the solutions across the front. Instead of solving each GA optimization problem independently with a SQP algorithm, all optimization problems are solved simultaneously with a new "Wave Front" SQP algorithm.

978-3-9810801-2-4/DATE07 © 2007 EDAA

The idea behind the Wave Front optimization is the following: All GA optimization problems aim at 'similar' goals. Each looks for improvement of the performances, aiming at different final trade-offs. Improvement in one optimization problem might improve the other optimization problems as well, especially in an early phase of the optimization, when the solutions are still far away from the front. Additionally SQP is a local optimization method, which converges into local minima. For the Wave Front optimization, global convergence is improved, because an optimization problem that terminates prematurely in a local minimum is still provided with parameter vectors from the other optimization problems. These could be nearer to the global optimum, helping the stuck optimization to jump out of the local minimum.

Instead of iterating independently, the simultaneous GA optimization problems share the generated parameter vectors in a set with each other. New parameter vectors are added to the set by running a single SQP step from the respective best parameter vector in the set for each GA optimization problem. This set of parameter vectors of Wave Front optimization corresponds to the population in GEAs.

The Pareto-optimal front can be estimated in each iteration step by selecting the best vectors for each optimization problem of the set of accepted parameter vectors. The performance vectors build a wave front in the performance space, which is improved iteratively until it coincides with the Pareto-optimal front.

The algorithm requires the determination of the best parameter vector for each individual optimization problem in each iteration step. It will be shown that determining the best parameter vector from a set of possible candidates can be implemented easier for the GA approach through an equivalent Minmax formulation than for the NBI approach.

The presented approach shows advantages compared to an NBI approach with a standard SQP algorithm. First the GA approach is numerically easier to minimize. The simultaneous optimization of all optimization problems improves the convergence speed: Slowly improving optimization problems are supplied with better solutions from the other optimization problems. Global convergence is improved, as additional parameters are provided. Additionally the algorithm collects the new parameter vectors generated by the optimization problem. The performances are simulated in parallel for all new parameter vectors, reducing the computational time.

The rest of the paper is structured as follows: In Section 2 analog circuit sizing and Pareto optimization are reviewed. In Section 3 the NBI and GA approaches are discussed. In Section 4 the new wave front optimization algorithm is presented. In Section 5 we give experimental results, and Section 6 concludes.

2. Circuit Sizing and Pareto Optimization

2.1. Circuit Sizing and Constraints

By circuit sizing we refer to the process of determining the values of a set of design parameters **p**, such as transistor lengths and widths, such that the circuit performances **f**, such as gain or power consumption, are optimal [5,8,14]. To evaluate the performances **f** for a given set of parameters **p**, the circuit is simulated with an analog simulator like Spice.

Analog circuits usually include basic building blocks as current mirrors or differential pairs. Circuit sizing must fulfil technology-specific sizing constraints, so that the functionality of these blocks can be assured [9].

These sizing constraints can be geometrical, such as minimum transistor sizes, or electrical, such as keeping the transistor operating point in the saturation region.

The sizing constraints lead to a set of inequalities $\mathbf{c}(\mathbf{p})\geq\mathbf{0}$ which must be fulfilled, so that the circuit works in a technical meaningful operating region.

2.2. Multi-objective optimization

During circuit sizing a whole set of performances **f** must be minimized[1]. Therefore circuit sizing can be formulated as a multi-objective optimization problem:

$$\min_{\mathbf{p}} \mathbf{f}(\mathbf{p}) = \begin{bmatrix} f_1(\mathbf{p}) \\ \vdots \\ f_N(\mathbf{p}) \end{bmatrix} \quad \text{s.t.} \quad \mathbf{c}(\mathbf{p}) \geq \mathbf{0} \qquad (1)$$

In most cases a performance cannot be optimized independently. The solution of this problem is called the Pareto-optimal front. It consists of all performance vectors, whose characteristics are that one performance vector can only be optimized on cost of another.

Figure 1 shows the Pareto-optimal front for two performances f_1 and f_2. The front lies at the border of the feasible performance space, which includes all feasible performance vectors. The individual minima $\mathbf{f}^{*1}$ und $\mathbf{f}^{*2}$ form the borders of the Pareto-optimal front.

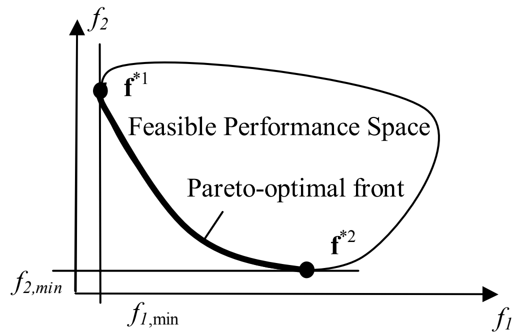

Figure 1: The Pareto-optimal front

[1] Maximization is included through max f = -min -f.

3. Deterministic Approaches to Pareto Optimization

3.1. Normal Boundary Intersection (NBI)

In [4,20] the Normal–Boundary-Intersection (NBI) method was introduced for Pareto optimization. NBI is a two-step method. In a first step the individual minima (IM) $\mathbf{f}^{*i}$ of the performances f_i with $i=1 \ldots N$ are determined by minimizing the following objective function $o_{IM,i}$:

$$\min_{\mathbf{p}}[o_{IM,i} = f_i(\mathbf{p})] \text{ s.t. } \mathbf{c}(\mathbf{p}) \geq \mathbf{0} \tag{3}$$

The IM vectors are used as columns to build the matrix **F**:

$$\mathbf{F} = \left[\mathbf{f}^{*1} \quad \cdots \quad \mathbf{f}^{*N}\right] \tag{4}$$

The quasi-normal vector **n** can be calculated with:

$$\mathbf{n} = \mathbf{f}^{*1} + \ldots + \mathbf{f}^{*N} \tag{5}$$

Additionally a set of positive weight vectors $\mathbf{w}_s$ with $s=1 \ldots S$ is required. (S is equal to the number of Pareto points to be calculated between the IM). Then S optimization problems of the following form are solved:

$$\boxed{\min_{\mathbf{r}}[o_{NBI,s} = t] \text{ s.t. } \mathbf{F}\mathbf{w}_s + \mathbf{n}t = \mathbf{f}(\mathbf{p}) \wedge \mathbf{c}(\mathbf{p}) \geq \mathbf{0}} \tag{6}$$

An additional parameter t is introduced, so $\mathbf{r}=[\mathbf{p}^T \; t]^T$.

Figure 2 illustrates the approach for N=2 and S=3: The weight vectors are in this case: $\mathbf{w}_1=[0.25 \; 0.75]^T$, $\mathbf{w}_2=[0.5 \; 0.5]^T$ und $\mathbf{w}_3=[0.75 \; 0.25]^T$. The points $\mathbf{F}\mathbf{w}_s$ lie on the connecting line of the IM in the 2D case and on the convex hull of the individual minima (CHIM) in the general case. The additional constraints can be seen as search lines in the performance space, dependent on the point $\mathbf{F}\mathbf{w}_s$ and the search direction **n**, which is quasi-normal to the CHIM. The calculated trade-off points are located on the intersection of the search lines and the Pareto-optimal front and are evenly spread across the front.

3.2 Goal Attainment

The Goal Attainment (GA) approach exchanges the equality constraints of the NBI method with inequality constraints [14,15,16,20]:

$$\boxed{\min_{\mathbf{r}}[o_{GA,s} = t] \text{ s.t. } \mathbf{F}\mathbf{w}_s + \mathbf{n}t \geq \mathbf{f}(\mathbf{p}) \wedge \mathbf{c}(\mathbf{p}) \geq \mathbf{0}} \tag{7}$$

The inequality constraints demand that the performance vector $\mathbf{f}(\mathbf{p})$ must be smaller or equal in each element than the point on the search line. Figure 3 illustrates the GA approach. Highlighted in grey are the sectors for different t, for which the inequality constraints are fulfilled. As t is decreased the sector of feasible performance vectors is reduced until only the optimum is included, which is again located at the intersection of the search line and the Pareto-optimal front, as this point has the smallest associated t value. This allows a Pareto-front generation as shown in figure 2 with the same even spread as NBI.

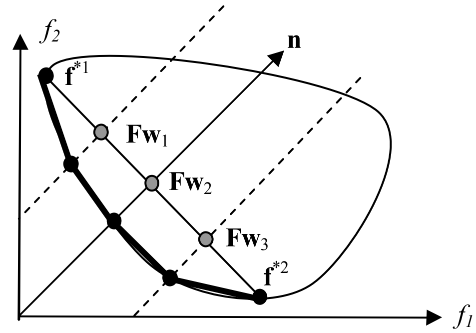

Figure 2: Pareto-optimal front obtained by NBI/GA

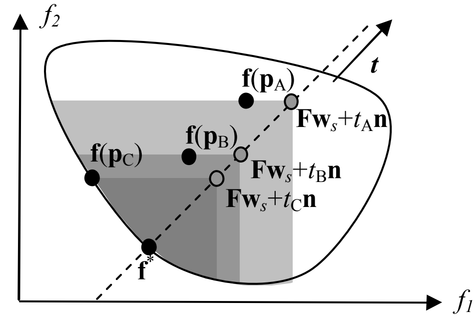

Figure 3: GA optimization

4. Wave Front Optimization

As was illustrated in Figure 2, each individual optimization problem (OP) to determine one Pareto point according to Eq. (6) or (7) is associated with a search line in the performance space. The OPs aim at different trade-offs, but improvement is achieved in a similar direction. A simultaneous optimization of all OPs is implemented, based on the idea that the OPs converge faster and more robust if supplied with the solutions of the other OPs. Slow converging OPs are provided with better solutions from the other OPs. Additionally, OPs trapped in individual minima are supplied with parameter vectors, which might aid them to overcome the individual minimum, improving global convergence.

The more Pareto-optimal performance vectors should be generated on the front, the closer the search lines lie to each other, making the effectiveness of this method dependent on the desired number of Pareto-optimal performance vectors S.

The algorithm additionally features an acceptance function that avoids the simulation of the performances for parameter vectors with too many constraints violations. Simulations are processed in parallel to reduce the computational time.

4.1. Algorithm

We assume that the IM have been obtained by single objective optimization of the N optimization problems of (3) and that an initial sizing $\mathbf{p}_0$ is given. With the IM and the number S of trade-off points known, the matrix $\mathbf{F}$, the weight vectors $\mathbf{w}_s$ and the quasi-normal vector $\mathbf{n}$ are generated. With these vectors the optimization problems (OPs) are formulated. Should a better value for an IM be encountered during the Wave Front optimization, then the IMs and OPs are updated and a new optimization for this IM is started from the better solution.

The algorithm works iteratively and on all OPs simultaneously. The stopping criterion is met if all OPs are labelled inactive or the maximum number of iterations is reached. An iteration step can be divided into four sub-steps:

1st step: Finding best parameter vectors

In the first step the best parameter vector for each OP is determined. Candidates are all accepted parameter vectors, which have been encountered so far. The best parameter vector is compared to the one of the last iteration. If improvement was achieved during the last iteration step, the OP is kept active, otherwise it is labelled inactive. New parameter vectors are only produced for active OPs in the 2nd step. An OP can become active again, if another OP provides it with a better parameter vector, after it has gone inactive.

2nd step: Generating new parameter vectors

A line search SQP algorithm is used. A quadratic sub-problem is minimized in each iteration step for each OP. It provides a search direction from the current best parameter vector towards improvement. To provide a good convergence the algorithm uses two methods to improve feasibility in each step, if infeasible solutions are encountered: A tilting of the search direction, which makes the constraints stricter, so that the tilted search direction shows into the feasible space [18] and a second order correction step [17,18] for all encountered infeasible parameter vectors.

3rd step: Acceptance function

An acceptance function is used to evaluate all encountered parameter vectors. Only parameter vectors are accepted that have an equal or less number of violations of sizing constraints $\mathbf{c}(\mathbf{p})\geq\mathbf{0}$ compared to all parameter vectors found so far. As soon as a parameter vector is found that has no constraint violation only feasible parameter vectors will be further accepted. The acceptance function assures that the algorithm works also for infeasible start solutions, gradually reducing the number of violated constraints until feasibility is reached.

4th step: Performance simulation

The performances are simulated only for the accepted parameter vectors, reducing simulation cost. As in this step all parameter vectors produced by all the OPs have been collected, the simulation can be done in parallel, decreasing the computational time. The new parameter vectors are added to the set of candidates, from which the best parameter vector can be chosen in the 1st step.

4.2. Comparison of GA and NBI

The GA approach shows two advantages compared to NBI, which makes it the better choice for the optimization with the proposed wave front optimization algorithm:

1. It can easily determine a best parameter vector from a set of alternatives.
2. It is numerically easier to be minimized with SQP.

These two points will be discussed in the following:

1. The presented Wave Front algorithm looks for the best parameter vector out of the set of accepted parameter vectors. Determining the best start parameter vector for the NBI method is difficult because of the additional variable t. Suitable values for t are provided with the search direction of the quadratic sub-problem. But they are only valid for the OP the sub-problem was solved for. The presented algorithm additionally considers parameter vectors produced by other OPs. A wrong choice of t leads to the selection of a suboptimal parameter vector. This is illustrated in Figure 4. The performance vector $\mathbf{f}(\mathbf{p}_\mathrm{A})$ and $\mathbf{f}(\mathbf{p}_\mathrm{B})$ of two alternative vectors are given. By inspection it can be seen that $\mathbf{p}_\mathrm{B}$ is the better start parameter vector. A default initialization value t_{AB} would lead of course to identical objective function values. Looking at Figure 4, it becomes clear, that the distance from $\mathbf{p}_\mathrm{B}$ to the point on the line given by t_{AB} is larger than the distance from $\mathbf{p}_\mathrm{A}$. Therefore any penalty function would assign a smaller cost value for $\mathbf{p}_\mathrm{A}$, favoring the wrong vector to start from.

In order to make the right decision the optimal parameter t must be obtained for all start points by minimizing the used penalty function to find an optimal t. This can be done analytically, but difficulties like suitable choices of the penalty function and penalty factor arise.

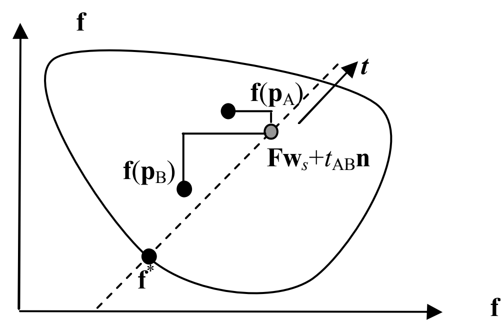

Figure 4: Start parameter vectors for NBI

The GA method offers a far easier way to find the best start parameter vector from a set of alternatives. There exists an equivalent Minmax formulation for every GA optimization problem. It minimizes the maximum of the competing objectives [14,15,16]:

$$\min_{\mathbf{p}}[o_{\max,s} = \max_i(\frac{1}{n_i}(f_i - [\mathbf{Fw_s}]_i))] \quad \text{s.t.} \quad \mathbf{c(p)} \geq \mathbf{0} \qquad (8)$$

The term $[\mathbf{Fw}_s]_i$ is the ith element of the vector $\mathbf{Fw}_s$. This objective function is non-smooth, therefore it is not suited for a gradient-based optimization algorithm like SQP. The equivalence of (7) and (8) will be illustrated with a simple example:

Example: Be N=2, $\mathbf{Fw}_s=[0\ \ 0]^T$, $\mathbf{n}=[1\ \ 1]^T$, and no constraints $\mathbf{c(p)}\geq\mathbf{0}$. The optimization problems (7) and (8) in this case are:

$$\min_{\mathbf{r}}[o_{\max} = \max(f_1, f_2)]; \qquad \min_{\mathbf{r}}[o_{GA} = t] \ \text{s.t.}\ t \geq f_1 \wedge t \geq f_2$$

For GA the objective function can be reduced by reducing t, which is a parameter of the optimization problem and therefore directly accessible. But t cannot be reduced arbitrarily as additional constraints have been added, which demand the parameter t to be greater than f_1 and f_2. For any f_1 and f_2, the value of t for which the objective function is minimized and feasibility is kept, is:

$$t = \begin{cases} f_1 & \text{if} \quad f_1 \geq f_2 \\ f_2 & \text{if} \quad f_1 < f_2 \end{cases} \Leftrightarrow t = \max(f_1, f_2)$$

In order to minimize t, the performances $f_1(\mathbf{p})$ and $f_2(\mathbf{p})$ have to be optimized by finding a parameter vector $\mathbf{p}$, such that t=max(f_1,f_2) becomes minimal. This is the optimization problem formulated in the Minmax approach, showing the equivalence of the two formulations.

The objective value of the Minmax objective function $o_{max}(\mathbf{f(p)})$ is directly based on the performance vector. It can be used to identify the best parameter vector.

2. In an SQP-based optimization, convergence problems may arise if the accepted step during the optimization is infeasible. The linearized feasible region of the quadratic sub-problem can be empty. In this case the constraints have to be relaxed [17], which can lead to poor search directions and a premature termination of the optimization run at a suboptimal solution.

Setting $t=o_{max,s}(\mathbf{f(p)})$ is equivalent to calculating a t, such that t is minimal and the additional constraints of the GA method are fulfilled. The algorithm automatically sets the variable t to o_{max} for all parameter vectors in the parameter set. Therefore infeasible solutions can only occur for GA, if a sizing constraints $\mathbf{c(p)}$ is violated.

The additional equality constraints of the NBI method can never be satisfied during a nonlinear optimization. Therefore the accepted steps remain infeasible during the complete optimization run, regardless if all sizing constraints $\mathbf{c(p)}\geq\mathbf{0}$ are fulfilled. As the equality constraints are also very stringent, empty feasible regions occur much more frequently for the NBI approach compared to the GA approach.

5. Experimental results

The method is applied on a five-stage ring voltage controlled oscillator (VCO). Figure 5 shows the schematic of the VCO. A 180nm process was used. Constraints were set on the output frequency range of the oscillator: The VCO must be able to generate a controlled frequency from 150 MHz up to 500 MHz.

Figure 6 shows the Pareto-optimal front of the VCO, in particular the trade-off between the supply current, the output jitter and the Gain of the VCO. To have a good estimation of the front 33 Pareto-optimal points were generated. The main Trade-Off, as can be seen in figure 6, is obviously a high jitter for high gain values. For values of VCO Gain above 2 GHz/V the jitter output is increasing far above values encountered for a VCO Gain below 2GHz/V. The generation of the front took about 8 hours on eight Pentium IVprocessor machines.

Additionally, the new method is compared with NBI and a standard SQP algorithm using a Miller operational amplifier. A 180 nm process was used. Figure 7 shows the results. As can be seen the standard SQP algorithm converges prematurely, so that important parts of the Pareto-optimal front are missing. The Wave Front algorithm produces parameter vectors in a more global region and converges to a solution that captures a much larger set of trade-offs. The optimization run with standard SQP required 60 min. The Wave Front optimization took 52 min. Both fronts were generated on seven Pentium IV machines. While producing Pareto fronts of much higher quality, the new Wave Front algorithm reduced the CPU time by 10%.

6. Conclusions

In this paper an advanced deterministic method for the calculation of the Pareto-optimal front of different performances of an analog circuit block has been presented. It is based on a Goal Attainment formulation of the optimization problem and a new Wave Front optimization algorithm. The optimization algorithm minimizes simultaneously individual optimization processes targeting discrete points of the Pareto front. Due to the equivalence of the Minmax Formulation and the Goal-Attainment formulation solutions can be exchanged easily between the different optimization processes.

The algorithm supports parallel simulation on multiple processors. It shows good global convergence due to the information sharing between the optimization processes. It shows a better convergence with less computational time compared to the state-of-the-art.

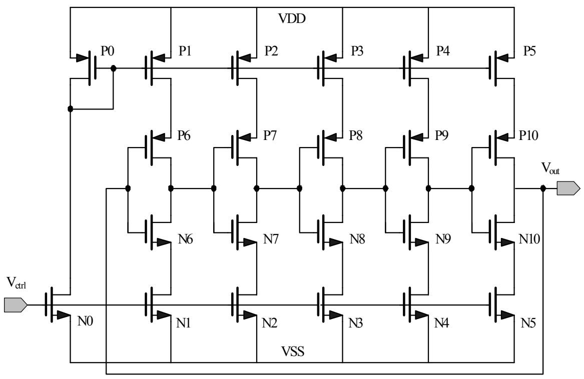

Figure 5: Schematic of a five-stage ring VCO

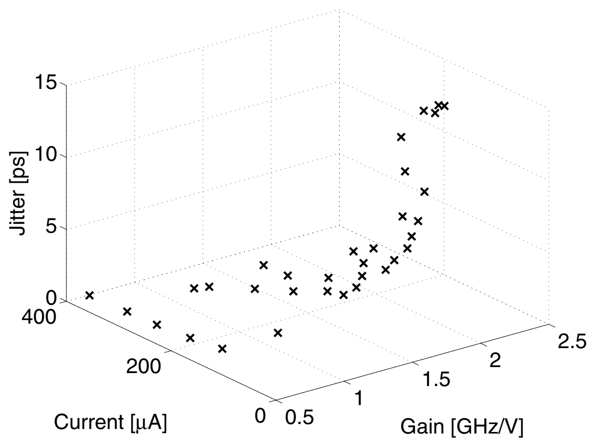

Figure 6: Pareto-optimal front of the Ring-VCO

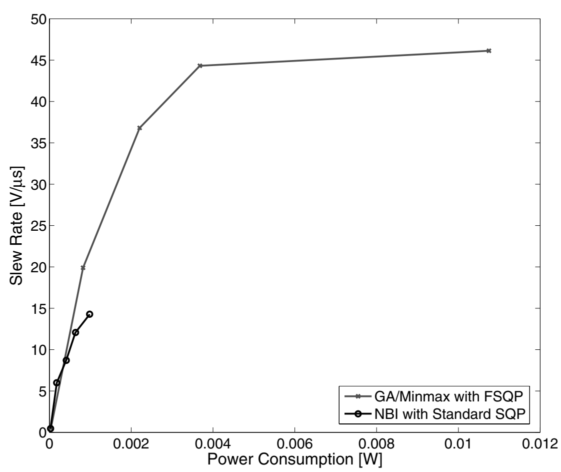

Figure 7: Pareto-optimal front of a Miller OpAmp

7. References

[1] S.K. Tiwary, S.Velu, R.A. Rutenbar und T. Mukherjee: Pareto-Optimal Modeling for Efficient PLL Optimization. NSTI Nanotech 2004, Boston

[2] J. Zou, D. Mueller, H. Graeb und U. Schlichtmann. A CPPLL Hierarchical Optimization Methodology Considering Jitter, Power and Locking Time. ACM/IEEE DAC 2006

[3] F. De Bernardinis, M. I. Jordan, und A. Sangiovanni-Vincentelli. Support vector machines for analog circuit performance representation. ACM/IEEE DAC 2003.

[4] G. Stehr, H. Graeb und K. Antreich. Performance Trade-off Analysis of Analog Circuits By Normal-Boundary Intersection. ACM/IEEE DAC 2003.

[5] M. Hershenson, S. Boyd, T. Lee. Optimal design of a CMOS Op-Amp via geometric programming. IEEE Trans. CAD 2001.

[6] R. Harjani und J. Shao. Feasibility and performance region modeling of analog and digital circuits. Analog Integrated Circuits and Signal Processing 1996.

[7] B. D. Smedt und G. G. E. Gielen. WATSON: Design space boundary exploration and model generation for analog and RF IC design. IEEE Trans. CAD 2003.

[8] R. Phelps, M. Krasnicki, R. Rutenbar, L.R. Carley, J. Hellums. Anaconda: Simulation-based synthesis of analog circuits via stochastic pattern search. IEEE Trans. CAD 2000.

[9] H. Graeb, S. Zizala, J. Eckmueller, und K. Antreich. The sizing rules method for analog integrated circuit design. IEEE/ACM ICCAD 2001.

[10] A Somani, P P Chakrabarti und A Patra. Mixing global and local competition in genetic optimization-based design space exploration of analog circuits, DATE 2005.

[11] T Eeckelaert, T McConaghy und G Gielen. Efficient multiobjective synthesis of analog circuits using hierarchical Pareto-optimal Hypersurfaces, DATE 2005.

[12] J. Dawson, S. Boyd, M. del Mar Hershenson, T. Lee. Optimal Allocation of Local Feedback in Multistage Amplifiers via Geometric Programming. IEEE Trans. CAS-I 2001.

[13] M. Chu, D. J. Allsot. Ellitist Nondominated Sorting Genetic Algorithm Based RF IC Optimizer. IEEE Trans. CAS 2005

[14] M. Lightner, S. Director. Multiple Criterion Optimization for the Design of Analog Circuits. IEEE Trans. CAS 1981

[15] Ji Guan, G Lin. On min-norm & min-max methods of multi-objective optimization. Mathematical Programming, 2003

[16] R.T. Marler und J.S. Arora. Survey of multi-objective optimization methods for engineering. Structural and Multidisciplinary Optimization 2004

[17] Nocedal, Wright. Numerical Optimization. Springer 1999

[18] C. Lawrence, A. Tits: A computationally efficient feasible sequential quadratic programming algorithm, Siam Journal on optimization, Vol. 11, No. 4 2000

[19] S. Tiwary, P. Tiwary, R. Rutenbar: Generation of Yield-Aware Pareto Surfaces for Hierarchical Circuit Design Space Exploration, ACM/IEEE DAC 2006

[20] I. Das and J. Dennis: Normal Boundary Intersection: A new Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. SIAM Journal on Optimization, 8(3): August1998