A New Approach for Combining Yield and Performance in Behavioural Models for Analogue Integrated Circuits

Sawal Ali, Reuben Wilcock, Peter Wilson, Andrew Brown
Electronics System Design Group, School of Electronics and Computer Science
University of Southampton, UK
{shma05r, rw3, prw, adb}@ecs.soton.ac.uk

Abstract

A new algorithm is presented that combines performance and variation objectives in a behavioural model for a given analogue circuit topology and process. The trade-offs between performance and yield are analysed using a combination of a multi-objective evolutionary algorithm and Monte Carlo simulation. The results indicate a significant improvement in overall simulation time and efficiency compared to conventional simulation based approaches, without a corresponding drop in accuracy. This approach is particularly useful in the hierarchical design of large and complex circuits where computational overheads are often prohibitive. The behavioural model has been developed in Verilog-A and tested extensively with practical designs using the Spectre™ simulator. A benchmark OTA circuit was used to demonstrate the proposed algorithm and the behaviour has been verified with transistor level simulations of this circuit and a higher level filter design. This has demonstrated that an accurate performance and yield prediction can be achieved using this model, in a fraction of the time of conventional simulation based methods.

1 Introduction

Advances in silicon technology over the last decade have led to increased integration of analogue and digital functional blocks onto the same chip. In such a mixed signal environment, the analogue circuits must use the same transistors as their digital neighbours. The increasing complexity and accuracy of device models has led to wide acceptance of simulation and optimisation based design techniques for the design of analogue blocks rather than hand calculations [1-4]. With reducing transistor sizes, the impact of process variations on analogue design has become very prominent and can lead to circuit performance and yield falling below specification. This issue has led to the consideration of yield in the design process, known as design for yield (DFY) [5]. The use of hierarchical design is commonplace in the IC design world and involves breaking down a large system into its constituent building blocks. A typical hierarchical design is shown in Figure 1. Not only does this approach simplify the design task but it also speeds up the design flow by encouraging design reuse. Behavioural and macro modelling is a useful technique that involves developing models from simulation that relate performance to circuit parameters. Although the initial time investment is high, subsequent design flows are significantly faster [6]. Recently, macromodelling has been used to predict the parametric yield and performance of a design [7].

In this paper, a novel approach is proposed that develops a combined performance and statistical variation behavioural model for analogue circuits. Multi-objective optimisation is used to capture optimal design points then a statistical variation analysis is performed using Monte Carlo simulation. A behavioural description is constructed to model the performance and variation of the circuit. The remainder of this paper is organized as follows: Section 2 provides necessary background; the proposed algorithm is detailed in section 3 and example results in sections 4 and 5. Concluding remarks are given in section 6.

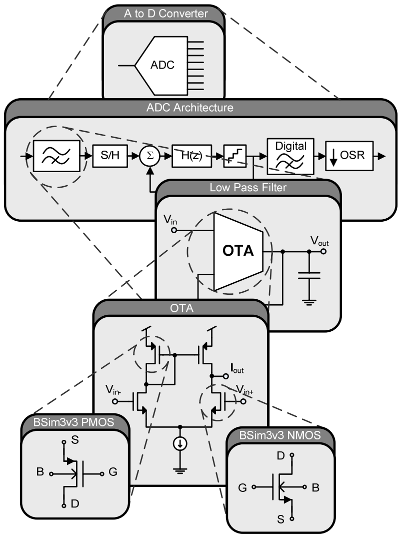


Figure 1. A typical system design hierarchy.

2 Background

2.1 Multi-Objective Optimisation

The optimisation formulation for more than one objective function is called multi-objective optimisation (MOO) which can be generally stated as:

$$\begin{gathered} \textit{Minimise/ Maximise } f_m(x),\ m = 1,2...M \\ \textit{Subject to } g_j(x) \geq 0,\ j = 1,2...J \end{gathered} \tag{1}$$

Where $f_m(x)$ is the set of M performance functions and $g_j(x)$ is the set of J constraints. The outcome from multi-objective optimisation is a set of optimal solutions [8]. MOO has an objective space with the number of dimensions equal to the number of objectives. Figure 2 shows the relationship between the parameter space and objective space where each point in the parameter space is a solution that corresponds to a point in the objective space. The black curve shown on the objective space is called the *Pareto front* and all solution points lying on this curve are called Pareto-optimal solutions. For example, point B shown in Fig. 2 is an example of a non-Pareto optimal point since a more optimal solution exists: point A. The method used in this work combines performance into a single objective using the following weighted summation, where W_m are the weightings for the performance functions:

$$\sum W_m f_m(x),\ m = 1,2...M \tag{2}$$

2.2 Table Model Functions

Behavioural models employing table model functions require the generation of sampled data points from circuit simulation. Interpolation and extrapolation techniques are then used to estimate a new value from the set of known values. Verilog-A supports three type of spline interpolation: linear, quadratic and cubic. The choice of interpolation is a trade off between accuracy and complexity. Cubic spline interpolation has been employed in this work to maximise accuracy. The third degree polynomial used to create the piece-wise interpolation curve is defined by equation (3), where a_i, b_i, c_i, and d_i are the coefficients for the polynomials.

$$S_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \tag{3}$$

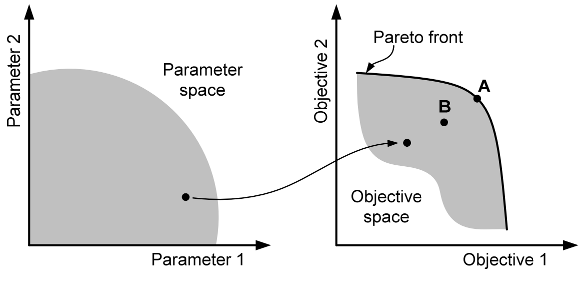


Figure 2. Parameter space and objective space.

3 Proposed Algorithm

The key steps in the proposed algorithm are shown in Figure 3. These steps are now discussed in more detail.

3.1 Netlist and Objective Function Generation

The starting point for the proposed algorithm is a circuit topology, process models and a set of performance functions. The first step involves generating a transistor level netlist for the chosen circuit topology. From this netlist a set of designable parameters are derived which will be used to change the circuit's performance. Examples of designable parameters include a transistor's length and width. Each parameter will have constraints imposed by the designer and once determined, these define the parameter space. The performance functions of the circuit are defined as the objective functions, for example open loop gain or phase margin. Testbench netlists are defined to simulate the performance for a certain set of parameters.

3.2 Muti-Objective Optimisation

In this stage the parameter space is explored and the design improved with respect to the objective functions. The optimisation implementation (MOO) is based on an evolutionary algorithm known as weight-based genetic algorithm (WBGA) [9]. WBGA is a powerful and efficient approach that uses a genetic algorithm (GA) to determine the objective function weighting. This is unlike classical weighted optimisations which often suffer difficulties in determination of the weight vector.

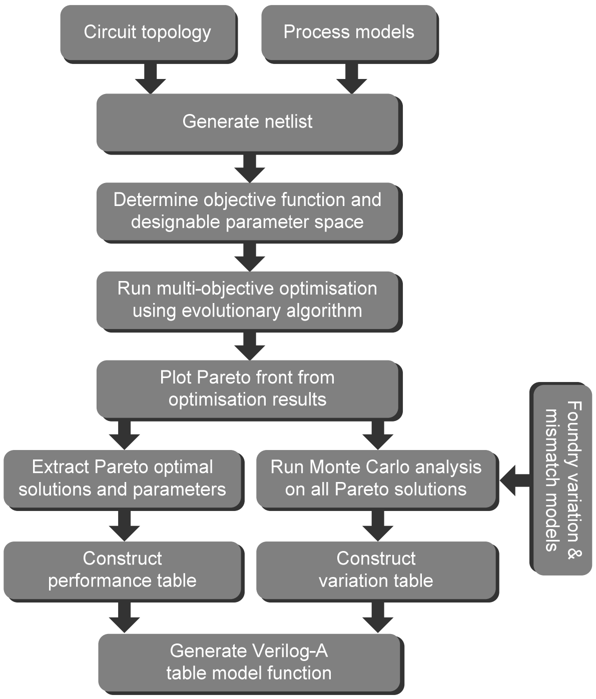
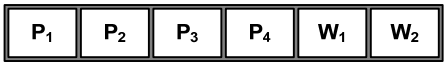

Circuit topology
Process models
Generate netlist
Determine objective function and designable parameter space
Run multi-objective optimisation using evolutionary algorithm
Plot Pareto front from optimisation results
Extract Pareto optimal solutions and parameters
Run Monte Carlo analysis on all Pareto solutions
Foundry variation & mismatch models
Construct performance table
Construct variation table
Generate Verilog-A table model function

Figure 3. Novel yield targetted algorithm.

The GA process involves generating a number of individuals (parameter sets), and optimising them over a number of generations, using selection techniques to identify the best solutions. The individuals are encapsulated in a set of parameters and weights defined as a GA string. Figure 4 shows an example of the GA string for 4 designable parameters and 2 objective function weightings.



P_1	P_2	P_3	P_4	W_1	W_2

Figure 4. Construction of an example GA string.

P_1 - P_4 and W_1, W_2 are the designable parameters and performance weights respectively, where the weights for the performance functions are normalised using (4).

$$w_i \leftarrow \frac{w_i}{\sum_{j=1}^{M} w_j} \tag{4}$$

During optimisation, populations of the GA string individuals are randomly generated. Throughout the evolutionary algorithm, the individuals will go through a process of crossover, mutation and selection from one generation to another [10]. The evolving designable parameter set replaces the existing designable parameters in the design netlist. This new netlist is then simulated and the performance for each of the objective functions is determined. The performance functions are multiplied by their respective weights given in the GA string and summed to determine a total (normalised) fitness score. This summation is shown in equation (5).

$$O_{(x^{(i)})} = \sum_{j=1}^{M} w_j^{x_w^{(i)}} \frac{f_j(x^{(i)}) - f_j^{\min}}{f_j^{\max} - f_j^{\min}} \tag{5}$$

Where $f_j(x^{(i)})$ is the objective function and w_j^x is its weight. This process will continue until the total number of generations is reached.

3.3 Performance Model from Pareto-front

In multi-objective optimisation with conflicting objectives, there cannot be a single optimum solution that optimises all the objectives. The previous optimisation step results in a number of optimal and non-optimal solutions. It is necessary at this point to determine the Pareto front which consists of the most optimal, non-dominated, solutions in the objective space. The two conditions below outline the procedure to establish these non-dominated solutions, thus giving the Pareto-front:

a) Any two solutions of the optimal set must be non dominated with respect to each other.

b) Any solution that does not belong to the optimal set is dominated by at least one member of optimal set

Having obtained the Pareto-front, the optimal performance functions and their designable parameters are stored in a data file which defines the optimal performance model of the design.

3.4 Variation Model from Monte Carlo Analysis

It is important to consider process variation as early as possible in the design flow. Such variations can cause a circuit's performance to vary from their nominal point, reducing the overall yield. This step in the proposed algorithm uses Monte Carlo (MC) analysis to model degradation of the performance function due to process variation. The MC analysis uses foundry variation models to simulate the effect of randomly selected parameter values on a circuit's performance [11]. During this step in the proposed algorithm, a MC analysis is run for each parameter solution set that lies on the Pareto-front. From this simulation, a set of performance variations is obtained.

3.5 Table Model Generation

The performance and variation data obtained from the previous stage are used to define the look-up table for a $table_model() function in Verilog-A. This function allows the module to approximate the behaviour of a system by interpolating between the performance and variations data points extracted from the MC analysis. At this stage, data files exist that describe the performance and variation functions for all the designable parameters. The syntax of the $table_model() function is shown below:

```
$table_model(f1,f2, "datafile.tbl", "control_string");
```

Where f1 & f2 are the performance functions, 'datafile.tbl' is the text file that contains the performance functions and design parameters and 'control_string' determines the interpolation and extrapolation method. In this algorithm, a cubic spline method is used for the interpolation. No extrapolation method is used, in order to avoid approximation of the data beyond the sampled data points. A $table_model() function is created for both the performance functions and the variation functions.

4 Design Example: Symmetrical OTA

This section presents a complete design example using a symmetrical operational transconductance amplifier (OTA) as the target circuit. OTAs are fundamental building blocks, often employed in analogue circuit design applications such as filters. All the following simulations were performed using the industry standard Cadence Spectre™ simulator with foundry level BSim3v3 transistor models from a standard 0.35μm AMS process (C35B4).

4.1 OTA Design and Objective Functions

The initial chosen circuit topology is a symmetrical OTA shown in Figure 5 which is a common benchmark circuit. The first step was to determine the designable parameters for the topology. In this example the lengths and widths for M3 to M10 make up a total of 8 designable parameters (M1and M2 dimensions are fixed). The two performance functions for the OTA are the open-loop gain and phase margin which both have a weighting.

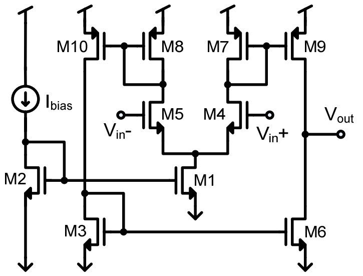


Figure 5. Symmetrical OTA topology.

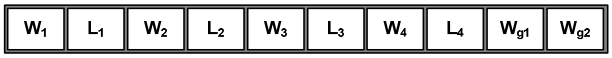
4.2 Multi-Objective Optimisation

The designable parameters, W_1-W_4 and L_1-L_4 are constrained within a reasonable range. Table 1 shows these ranges along with the two normalised performance function weights, W_{g1} and W_{g2}.

Design Parameter:	Range:
W_1 (M5,M4)	10um - 60um
L_1 (M5,M4)	0.35µm - 4µm
W_2 (M7,M9)	10um - 60um
L_2 (M7,M9)	0.35µm - 4µm
W_3 (M10,M8)	10um - 60um
L_3 (M10,M8)	0.35µm - 4µm
W_4 (M3,M6)	10um - 60um
L_4 (M3,M6)	0.35µm - 4µm
W_{g1} (Gain weight)	0 – 1 (normalised)
W_{g2} (Phase weight)	0 – 1 (normalised)

Table 1. Design parameters.

Once the parameters have been determined, a GA string can be constructed consisting of these and the performance weightings. The string is shown in Figure 6.



W_1	L_1	W_2	L_2	W_3	L_3	W_4	L_4	W_{g1}	W_{g2}

Figure 6. GA string for the design example.

The parameters are all normalised to keep them within the same range of [0~1]. The weighting vectors have already been normalised between [0~1] using equation (4). Each individual generated by the GA will consist of a set of designable parameters as defined by the GA String. The designable parameters are used for simulation and the weight vectors for the weight summation.

The same testbench netlist was used to determine both the open loop gain and phase margin for each individual. The total fitness score for each individual was calculated using the normalised weighted-summation formula explained in the previous section. A total of 100 generations each with a population size of 100 were used in this case, giving a total number of samples for the optimization of 10,000.

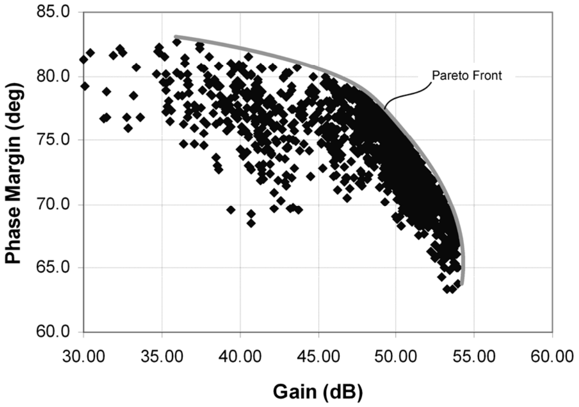


Figure 7. Gain and phase margin for individuals.

During the optimisation, the GA generates and optimises the designable parameters and weight vectors to achieve a higher fitness score, and hence optimises the performance functions. The result of the optimisation is a full set of designable parameters, weight vectors and performance functions.

4.3 Pareto-optimal front

To illustrate the results of the optimisation, Figure 7 shows a plot of open loop gain and phase margin for the 10,000 individuals in the example. The Pareto front can be clearly seen and contains 1022 optimum solutions (circuit candidates). These solutions define the performance model and this information is stored in a data file.

4.4 Monte Carlo analysis

Every optimal solution on the Pareto-front undergoes a Monte Carlo simulation using process variation and mismatch models. 200 samples were chosen for the MC simulation and from these the variation for each performance is calculated. This completes the variation model and this information is stored in a data file.

Design:	Gain (dB):	ΔGain (%):	PM (deg):	ΔPM (%):
21	49.78	0.52	76.3	1.50
22	49.90	0.52	76.1	1.51
24	49.98	0.51	76.0	1.51
25	50.17	0.51	75.8	1.52
26	50.35	0.50	75.5	1.56
27	50.45	0.49	75.3	1.57
34	51.06	0.44	74.1	1.69
35	51.14	0.51	74.0	1.71
37	51.24	0.42	73.8	1.69
38	51.62	0.42	73.2	1.68

Table 2. Performance and variation values.

At this point, a combined performance and variation model for the OTA is developed. Selection design points are shown in Table 2 which details the associated performance and variation values for each point. This table is defined as a look-up table for a $table_model() function with the resulting Verilog-A model given below:

```
analogue begin

  gain_delta = $table_model (gain, "gain_delta.tbl", "3E");
  pm_delta = $table_model (pm, "pm_delta.tbl", "3E");
  gain_prop = ((gain_delta/100)*gain)+gain;
  pm_prop = ((pm_delta/100)*pm)+pm;
  $display ("Propose Gain : %e", gain_prop);
  $display ("propose PM : %e", pm_prop);
  lp1 = $table_model (gain_prop,pm_prop,"lp1_data.tbl","3E,3E");
  lp2 = $table_model (gain_prop,pm_prop,"lp2_data.tbl","3E,3E");
  lp3 = $table_model (gain_prop,pm_prop,"lp3_data.tbl","3E,3E");
  lp4 = $table_model (gain_prop,pm_prop,"lp4_data.tbl","3E,3E");
  fptr=$fopen("params.dat");
  $fwrite(fptr, "\n Generated Design Parameters\n ");
  $fwrite(fptr, "%e %e %e %e", lp1,lp2,lp3,lp4);
  $fclose(fptr);
  $display ("params: = %e %e %e %e", lp1, lp2, lp3, lp4);
  gain_in_v = pow(10,gain_prop/20);
  V(out) <+ V(inp)*(-gain_in_v)-I(out)*ro;

end
```

From a given performance specification, the model will interpolate a new performance value that can produce the highest yield based on the performance variation. A new set of designable parameters is then interpolated from this new performance value. Table 3 shows an example where the required performance is a gain of greater than 50dB and a phase margin of greater than 74 degrees. The variation for the gain is obtained from the $table_model() function. In this case, the relevant lookup table points are those shown in Table 2 where it can be seen that the gain of 50dB is between design point 24 and 25. Interpolation is used to determine the variation for the gain between these points which is 0.51%. From this variation value it can be seen that the actual gain may vary from 49.75dB to 50.26dB. Therefore, in order to achieve maximum yield, the specified gain of the design must be at least 50.26dB. This will ensure that the required 50dB gain will be achieved within the process extremes. The value of 50.26dB therefore becomes the new targeted performance value and using this new value, the design parameters are interpolated from the performance table. The same strategy is applied for phase margin. Both of the new performance values for gain and phase margin will produce 100% yield.

Performance:	Required Performance:	Variation:	New Performance:
Gain	> 50dB	0.51%	50.26dB
Phase Margin	> 74 deg	1.71%	75.27 deg

Table 3. Interpolation example.

To verify the performance and yield from the behavioural model design, a comparison has been made with transistor level simulation using design parameters obtained from the $table_model(). This comparison is shown in Table 4. The percentage error in passband gain and phase margin was calculated between the OTA transistor simulation and interpolated values. Figure 8 shows the open loop gain for the Verilog-A model and transistor model. It can be seen from these comparisons that the Verilog-A function matches closely with the transistor level simulation.

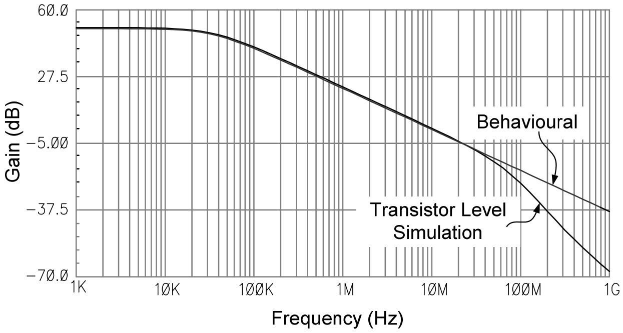


Figure 8. Open loop gain comparison.

Figure 8 shows a divergence in the comparison above 40MHz which is attributed to parasitic poles in the transistor circuit. Although these higher order effects are not modelled in this example, they could easily be incorporated if required. A Monte Carlo simulation using 500 samples was carried out and verified a yield of 100%.

Performance Functions	Transistor Model	Verilog-A Model	% error
Gain	50.73	50.26	0.93%
Phase Margin	76.06	75.27	1.03%

Table 4. Performance comparison.

Table 5 summarises the parameters associated with model development. A total of 10,000 simulations were run in the initial MOO step for the performance model and Monte Carlo analysis was performed on 1022 Pareto-optimal points for the variation model. The OTA design optimisation stage took 4 hours on a 1.2GHz Ultra Sparc 3, which compares well with a previously reported optimisation time of 7 hours for the same circuit [5].

Parameters:	Values:
No. Generations	100
Evaluation Samples	10,000
Pareto Points	1022
CPU Time (1.2GHz Sparc 3)	4 hours

Table 5. Design parameter summary.

5 Example Application

To demonstrate the advantages of the proposed approach the model developed for the OTA has been used to design a 2nd order low pass filter as shown in Figure 9.

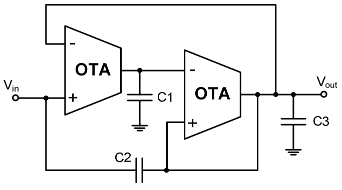


Figure 9. 2nd Order low-pass filter

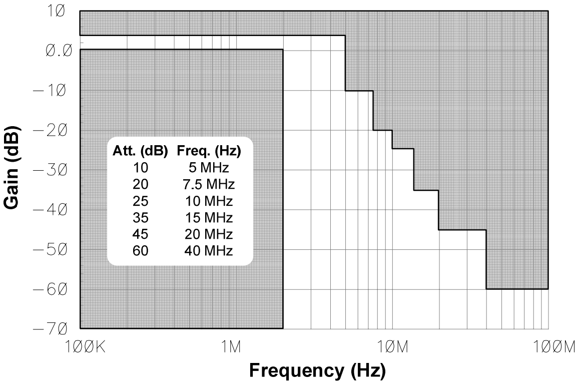


Figure 10. Filter specification

The filter was designed to typical anti-aliasing filter specification as shown in Figure 10. The specifications for the open loop gain and phase margin for the OTA are 50dB and 60 degrees respectively. The performance and variation model was used to select OTAs that met these specifications taking into account their variations. Simulation-based optimisation was performed on the filter design to find an optimum solution for capacitor values C1, C2 and C3 within the filter specifications. A total of 30 individuals and 40 generations were used for the MOO. To verify the predicted yield given by the proposed approach, a Monte Carlo analysis with 500 samples was run on the final design. This analysis confirmed a yield of 100%. Figure 11 shows the transistor level typical mean response of the filter which can be seen to meet the specifications.

6 Conclusions

This paper has presented a new algorithm that combines performance and process variation objectives in a behavioural model for an analogue circuit topology. Multi-objective optimisation with genetic algorithm is used to explore tradeoffs between performance and yield, leading to a set of Pareto optimal solutions for the design. Monte Carlo variation analysis is performed on all the Pareto optimal solutions, and a table is constructed for both the performance and variation analysis. A behavioural model developed in Verilog-A is used together with this table to determine the parameters required to achieve the highest yield within a given specification. After the initial time investment to create the model and table there are significant improvements in overall simulation time and efficiency compared to conventional simulation based approaches. These benefits are enjoyed without a corresponding drop in accuracy. A benchmark OTA topology and standard filter design were used to demonstrate the proposed algorithm and the behaviour has been verified with transistor level simulations.

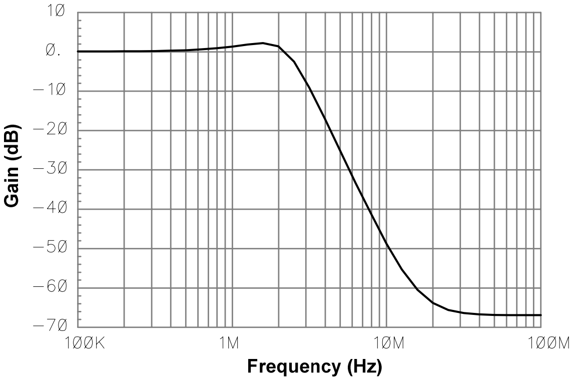


Figure 11. Filter response

References

[1] B.De Smedt and G.Gielen, “Watson: design space boundary exploration and model generation for analogue and rfic design,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transaction on*, vol.22, no.2, pp. 213-224, Feb. 2003.

[2] Stehr G., Graeb H., and Antreich K., “Performance trade-off analysis of analog circuits by normal-boundary intersection.,” *in Proc. Of Design Automation Conferenc 2003,* pp. 958-963.

[3] E.S. Ochotta, R.Rutenbar, and L.R. Carley, “Synthesis of high-performance analogue circuits in ASTRX/OBLX,” *IEEE Trans. Computer -Aided Design,* vol. 15, pp.273-294, Mar. 1996

[4] M. Krasnicki, R.Phelps, J.R.Hellums, M. McClung, R.A Rutenbar and L. Richard Carley, “ASF: a practical simulation-based methodology for the synthesis of custom analogue circuits,” in Proc. ICCAD 2001, pp 350-357.

[5] B. D. Smedt, G. Gielen, “HOLMES: Capturing the yield-optimized design space boundaries of analogue and RF Integrated Circuits.” In *Proc. Of the Design,Automation and Test* in Europe Conference and Exhibition, 2003.

[6] H. Liu, A. Singhee, R. Rutenbar and L. Carley, “Remembrance of circuits past: macromodeling by data mining in large analog design spaces,” in *Design Automation Conference, 2002. Proceedings. 39th,* 10-14 June 2002, pp. 437-442.

[7] Shao, J. and Harjani, R. “Macromodeling of analog circuits for hierarchical circuit design,” *In Proc. Of 1994 IEEE/ACM International Conference on Computer Aided Design,* pp. 656-663.

[8] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons Ltd, 2001.

[9] Hajela, P., Lee, E. and Lin, C.Y. , “Genetic Algorithms in structural topology optimization. *In Proc. Of the NATO Advanced Research Workshop on Topology Design of Structures* 1993, pp. 117-133.

[10] D.E. Goldberg: “Genetic Algorithms in search, optimization and machine learning,”Addison-Wesley (1989).

[11] R. Spence and R.S. Soin, “Tolerance Design of Electronic, Addison Wesley”, 1988.