# 6 Optimization of Analog Circuits and Systems – Applications

**Abstract.** In the previous chapters there was a description of the optimization methodology and the supporting tool that simplifies the design tasks of analog integrated circuits. The developed design optimization tool, GENOM, based on evolutionary computation techniques and incorporating heuristic knowledge on the automatic control mechanism was combined efficiently with a learning strategy based on SVM to improve the convergence speed of the optimization algorithm. This chapter demonstrates the capabilities and performances of the implemented design optimization methodology when applied to several analog synthesis experiments and provides some insight into factors that affect the synthesis process. Several state of the art circuit blocks will be introduced and optimized for performance and efficiency. Particularly, the performance and effectiveness of GENOM optimizer will be compared with one important reference tool.

## 6.1 Testing the Performance of Analog Circuits

Operational amplifiers (OpAmps) are the fundamental building blocks of many analog and mixed-signal systems. OpAmps arranged in structures of different levels of complexity are used to realize functions ranging from dc bias generation to high speed amplification or filtering. Table 6.1 presents the general characteristics [1] of some of the OpAmps that will be covered in this chapter.

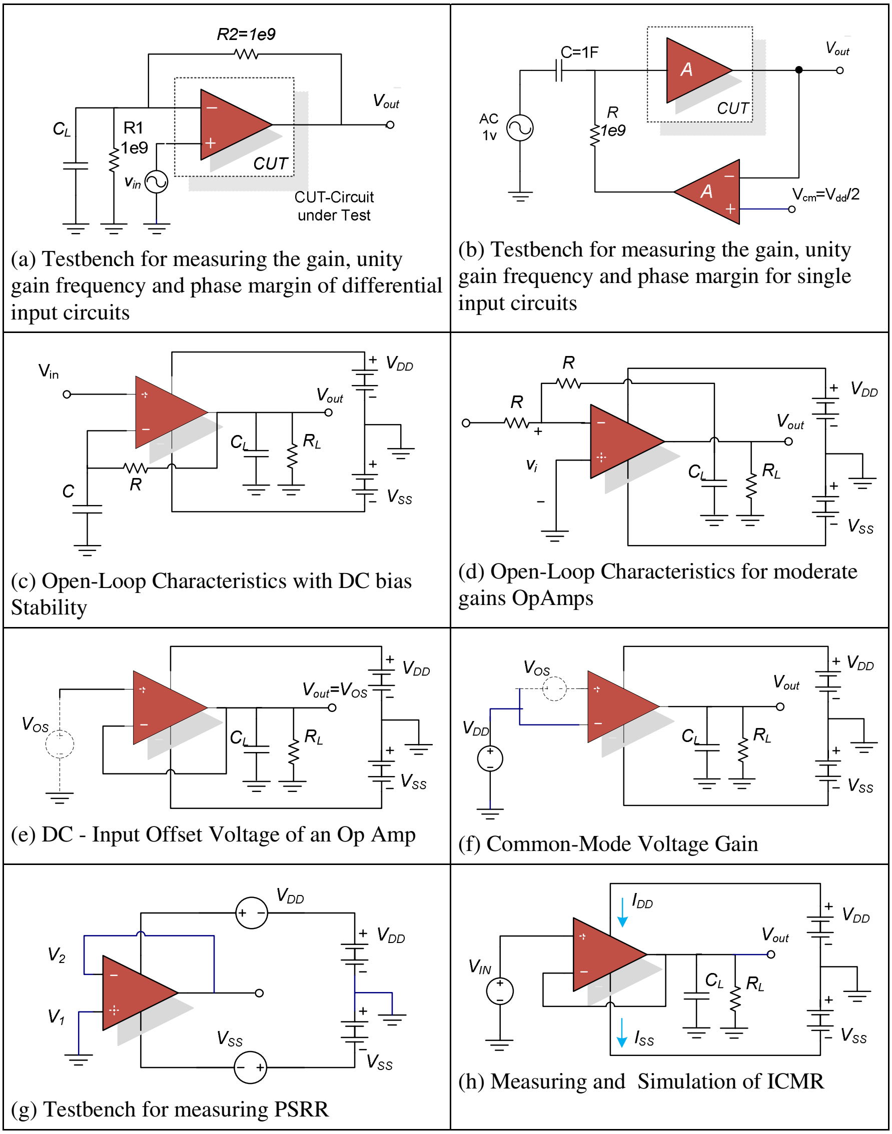
**Table 6.1** General comparison

| | **Gain** | **Output Swing** | **Power Dissipation** | **Speed** | **Noise** |
|---|---|---|---|---|---|
| **Two Stage** | Medium | Medium | Low | High | Low |
| **Folded Cascode** | Medium | Medium | Medium | High | Low |
| **Telescopic** | High | Low | Medium | High | Low |
| **Gain-Boosted** | High | Medium | High | Medium | Medium |

Simulation and testing of CMOS Opamps involve the measure of several performances parameters such as open-loop gain, open-loop frequency response (including phase margin), input-offset voltage, common-mode gain, common-mode rejection ratio (CMRR), power-supply rejection ratio (PSRR), output resistance, noise, output swing, power dissipation and transient response including slew rate.

M.F.M. Barros et al.: Analog Circuits and Systems Optimization, SCI 294, pp. 139–186.
springerlink.com

Special configurations and techniques are necessary to acquire these measurements. The testbench configurations supply the environment (stimulus, load, supplies, etc.) in which the circuit is to be tested. Fig. 6.1 presents the testbench configurations considered for the selected examples [2].



(a) Testbench for measuring the gain, unity gain frequency and phase margin of differential input circuits

(b) Testbench for measuring the gain, unity gain frequency and phase margin for single input circuits

(c) Open-Loop Characteristics with DC bias Stability

(d) Open-Loop Characteristics for moderate gains OpAmps

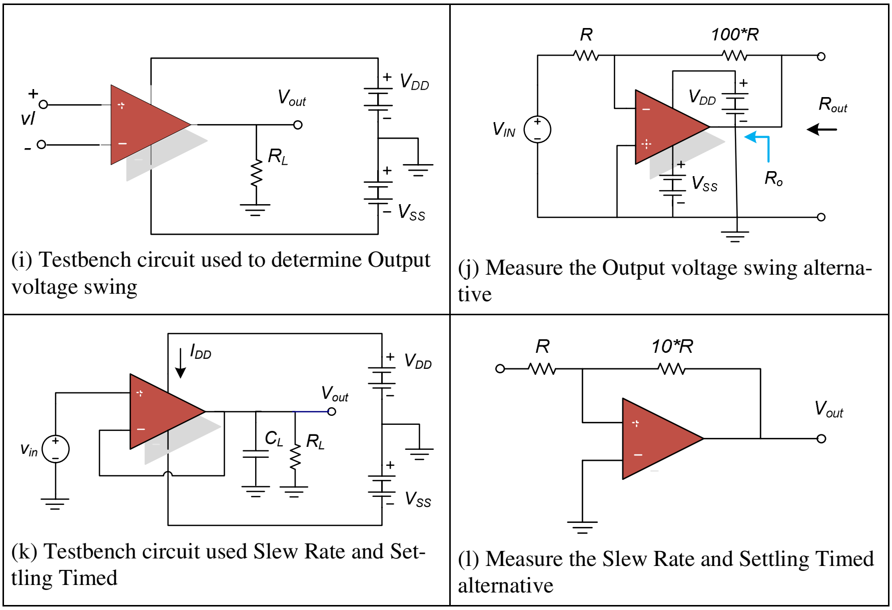
(e) DC - Input Offset Voltage of an Op Amp

(f) Common-Mode Voltage Gain

(g) Testbench for measuring PSRR

(h) Measuring and Simulation of ICMR

**Fig. 6.1** Testbenches to measure the performances values



(i) Testbench circuit used to determine Output voltage swing

(j) Measure the Output voltage swing alternative

(k) Testbench circuit used Slew Rate and Settling Timed

(l) Measure the Slew Rate and Settling Timed alternative

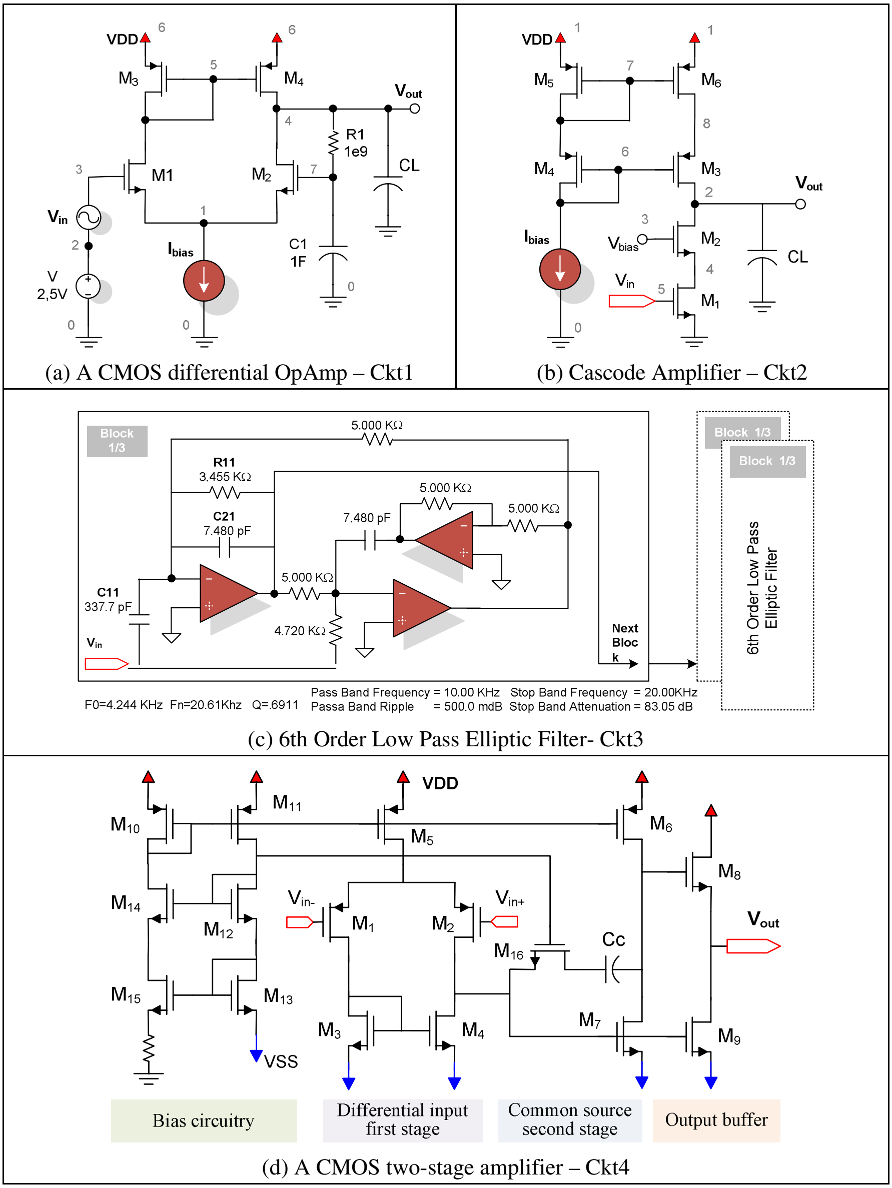
**Fig. 6.1** (*continued*)

## 6.2 Testing the GENOM – Selected Circuit Topologies

Since analog benchmark circuits are still unavailable for synthesis purposes, the first testing circuits were collected from the well-known class of CMOS operational amplifiers and also include a low pass elliptic filter listed in Table 6.2 ordered by circuit complexity. OpAmps and filters are fundamental building blocks often employed in analog circuit design applications. Each circuit includes appropriate testbenches to obtain the desired performances parameters measures. The testbench circuit configuration of Fig. 6.2 a), b) and d) were used in these experiments to determine the open loop gain, unity gain frequency, phase margin and power consumption for the single ended circuits. The filter specifications are different and will be defined later. All OpAmp circuits examples were designed using a 0.35-µm AMS (Austria Mikro Systems International AG) CMOS technology process with a supply voltage of 3.3V but the optimization process is fully independent from technology.

The design first step is to determine the design parameters, the functional constraints of the problem and the performance objectives for each topology. Table 6.2 describes the complexity level for each test circuit. In this study, the design parameters are composed by the lengths, widths and/or multiplicity of transistors and are constrained by the ranges in geometry defined in Table 6.3.

Once the parameters have been defined, the GA chromosome can be constructed representing an individual or a candidate solution. The optimization



(a) A CMOS differential OpAmp – Ckt1

(b) Cascode Amplifier – Ckt2

(c) 6th Order Low Pass Elliptic Filter- Ckt3

(d) A CMOS two-stage amplifier – Ckt4

**Fig. 6.2** The suite of circuit schematics used in tests

design parameters domain and the adopted technological grid define the complexity of the problem. A set of fundamental designer rules as well as the matching conditions for each design case is depicted in Table 6.4. This set of rules makes up the functional constraints of design optimization. All measures of performance and the conformance level for each designer rules ("*satisfiability*") are obtained by electrical simulation. In each optimization run, the GA generates and optimizes the design parameters according to the fitness function built based on the performance specifications defined in Table 6.5. The total fitness score for each individual was calculated using the fitness function presented in Sect. 3.3.1.

**Table 6.2** Class of circuits used in the tests

| **Ident.** | **Name** | **No.Devices** | **Opt.Var(a)** | **Constr.(b)** | **Refs** |
|---|---|---|---|---|---|
| **Ckt1** | Differential OpAmp | 4 | 4 | 12 | -- |
| **Ckt2** | Cascode Amplifier | 7 | 7 | 12 | [3] Exa. 3.11 |
| **Ckt3** | 6th Order Low Pass Elliptic Filter | 27 | 9 | -- | -- |
| **Ckt4** | Two-Stage OpAmp | 16 | 10 | 30 | [3] Exa. 5.2 |

(a) Number of optimization variables (b) Number of constraints

**Table 6.3** Design parameters range

| **Id** | **W's (µm)** (a) | **L's (µm)**(a) | **Ibias (µA)** | **Search Space** |
|---|---|---|---|---|
| **Ckt1** | [1, 400,1] | [0.35,10,0.1] | 200 fixed | 2.137e+10 |
| **Ckt2** | [1, 400,1] | [0.35,10,0.1] | [10,60,20] | 8.905e+14 |
| **Ckt4** | [1, 400,1] | [0.35,10,0.1] | none | 8.220e+18 |

(a) Note: all parameters ranges means [min, max, grid size] respectively

**Table 6.4** Matching and technology constraints details

| **Id** | **Techn. Constraints**<br>**Matching** | **$V_{GS}$ - $V_T$** (a)<br>**[Min - Max]** | **$V_{DS}$ - $V_{DSAT}$** (b)<br>**Min / Max** |
|---|---|---|---|
| **Ckt1** | M1=M2 | [50-200] mV | > 50 mV |
| | M3=M4 | [100-300] mV | > 50 mV |
| **Ckt2** | M1=M2; | > 50 mV | > 50 mV |
| | M3=M4=M5=M6 | [100-300] mV | > 50 mV |
| **Ckt4** | M1=M2; | > 50 mV | > 50 mV |
| | M3=M4=M7; | < 200 mV | > 50 mV |
| | M8=M9; | < 200 mV | > 50 mV |
| | M10=M11=M12 =M14 | < 200 mV | > 50 mV |

(a) Overdrive voltages (b) Drain-sources voltages margin

**Table 6.5** Specifications/requirements

| **Id** | **Gain** | **GBW** | **Phase** | **Power** |
|---|---|---|---|---|
| **Ckt1** | > 50 dB | >40 MHz | 60º<Ph<90º | Min (mW) |
| **Ckt2** | > 70 dB | >25 MHz | 60º<Ph<90º | Min (mW) |
| **Ckt4** | > 65 dB | >20 MHz | 60º<Ph<90º | Min (mW) |

The optimization algorithms were all initialized with the following default parameters listed in Table 6.6. In particular, the *GA-STD*, *GA-MOD* and *GA-SVM* will be used in the following experiments. GA-STD specifies the configuration of a standard GA, whereas, GA-MOD covers the new GENOM kernel, but, only the modified evolutionary module is considered, and finally, the proposed GA-SVM defines the hybrid method composed by the GA-MOD extended by the SVM learning method.

**Table 6.6** Optimization algorithm parameters

| **Algorithm Setup** | **GA-STD** | **GA-MOD** | **GA-SVM** |
|---|---|---|---|
| initPOP | 64 | 64 | 64 |
| popSize (μ) | initPOP/2 | initPOP/2 | iintPOP/2 |
| Elite members (λ) | initPOP /8 | initPOP /8 | initPOP /8 |
| Initial Sampling | Random | Latin Hyper Sampling | Grid |
| Selection | Random | Tournament w/ Feasibility | Tournament w/ Feasibility |
| Sort | Min. cost | Priority to Feasibility | Priority to Feasibility |
| Crossover | 1-Point Unif. | 2-Point Unif. | 2-Point Unif. |
| Mut.Rate | 5 % fixed | 5 % Dynamic | 5 % Dynamic |
| Kernel type | GA-STD | GA modified | GA+SVM |
| Training Set SVM | none | none | 2000 Unif Sampling points |
| Early Stop | yes | yes | yes |

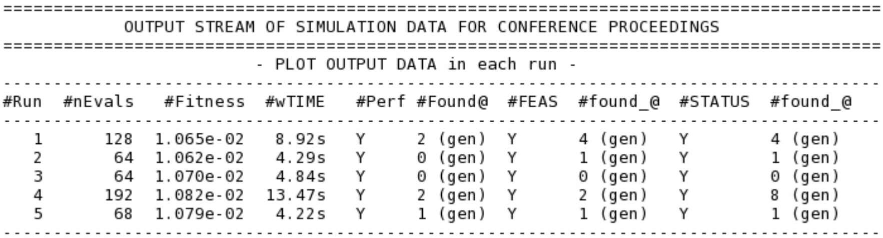
Some of the common parameters include the initial population size population size (μ=32), elite size (8), initial mutation rate (5%), a 2-pairs tournament-crossover probability in 50% of μ and a normal distribution method for generating the initial population. The stop criterion was here defined as a maximum number of generations or as soon as it reaches the first solution. For this particularly experiments, the SVM meta-parameters were found for the first time model generation and then fixed (regularization parameter C=4, variance σ=1/n). A cross validation method [4-6] for optimal parameter selection will execute automatically for each model update.

## 6.3 GENOM Convergence Tests

In this section a set of experiments that tests the convergence and performance of GENOM GA-MOD algorithm will be presented. In particular, a simple testbench OpAmp circuit from Fig. 6.2a) will be used in this study. This circuit has 4 independent variables and was synthesized within a 0.35μm, 3.3V technology. Each variable has a reasonable range and all were initialized by a random sampling methodology.

### *6.3.1 The Analog IC Design Approach*

The GENOM design methodology is ruled by two types of objectives: the goals and the constraints. All design goals and all design constraints must be satisfied in order to obtain a circuit, which fulfills the aims of the application. As soon as, a satisfactory solution is found, the optimizer continues his search for the improvement of each goal, while ensuring that the constraints are still satisfied. During the search, it can happen that a candidate solution may satisfy all performance constraints and goals but may not meet the functional constraints or vice-versa. The space of feasible solutions is given by the candidate solutions that belong simultaneously to the performance and feasibility regions. The computation effort spent to find the solution space will increase as more and more performance constraints, design trade-offs, or even process variation parameters are taken into account when designing robust design circuits. Fig. 6.3 and Table 6.7 show the algorithm performance result for the simple OpAmps for 5 runs executed on an AMD X64 2.8 GHz dual core machine and use HSPICE to simulate the circuit and extract performance parameters.



```
=============================================================================
          OUTPUT STREAM OF SIMULATION DATA FOR CONFERENCE PROCEEDINGS
=============================================================================
                       - PLOT OUTPUT DATA in each run -
-----------------------------------------------------------------------------
#Run  #nEvals   #Fitness  #wTIME   #Perf #Found@  #FEAS  #found_@  #STATUS  #found_@
-----------------------------------------------------------------------------
   1      128  1.065e-02   8.92s   Y      2 (gen)  Y      4 (gen)   Y        4 (gen)
   2       64  1.062e-02   4.29s   Y      0 (gen)  Y      1 (gen)   Y        1 (gen)
   3       64  1.070e-02   4.84s   Y      0 (gen)  Y      0 (gen)   Y        0 (gen)
   4      192  1.082e-02  13.47s   Y      2 (gen)  Y      2 (gen)   Y        8 (gen)
   5       68  1.079e-02   4.22s   Y      1 (gen)  Y      1 (gen)   Y        1 (gen)
-----------------------------------------------------------------------------
```

**Fig. 6.3** Print screen with statistical data from nominal optimization

Each line from Fig. 6.3 depicts the run number, *#Run*, the number of evaluations in each run, "*#nEvals*", the final fitness value, "*#Fitness*", simulation time, "*#wTIME*", then its followed by three binary values indicating whether a solution satisfies all performance constraints "*#PERF*", all feasibility (designer rules) constraints "*#FEAS*" or both, meaning that a solution was found *#STATUS*=Y at generation "*found_@*".

The "*Perf. Specs*" columns in Table 6.7 mean the fitness, time and evaluation number when the circuit meets all design specs of the problem. In the same way, the "*Specs&Rules*" column represents the same features when the circuit meets all design specs, as well as, and all functional constraints of the problem, considering the nominal optimization with typical working conditions. The "*Corners*" column also represents the same features, in case the circuit meets all design specs and all functional constraints in all corner points of the problem.

**Table 6.7** Overall performance measures

| | #Fitness | | | #TIME (s) | | | #nEVALs | | |
|---|---|---|---|---|---|---|---|---|---|
| RUNs | Perf. Specs | Specs & Rules | Corners | Perf. Specs | Specs & Rules | Corners | Perf. Specs | Specs & Rules | Corners |
| Run-1 | 1.065e-02 | 1.065e-02 | 1.059e-02 | 6.75 | 8.92 | 174.98 | 96 | 128 | 2578 |
| Run-2 | 1.062e-02 | 1.062e-02 | 1.062e-02 | 4.29 | 4.29 | 184.08 | <64 | <64 | 2720 |
| Run-3 | 1.07e-02 | 1.07e-02 | 1.061e-02 | 4.84 | 4.84 | 232.93 | <64 | <64 | 3440 |
| Run-4 | 1.073e-02 | 1.082e-02 | 1.059e-02 | 8.10 | 13.47 | 260.70 | 112 | 192 | 3888 |
| Run-5 | 1.079e-02 | 1.077e-02 | 1.057e-02 | 4.22 | 5.28 | 232.25 | <64 | 68 | 3424 |

The optimization process considering only typical conditions solved the problem quickly, and spent only a very few generations (from 0 to 8) as seen in Fig. 6.3 to achieve the performance specs satisfying all design constraints (rules). However, in corner optimization the number of generations increases for around 15-20 generations. Since each candidate solution for corner analysis requires 9 SPICE simulations (one simulation for each corner point), a minimum of 2578 and a maximum of 3888 HSPICE simulations were performed taking into account all runs.

### *6.3.2 Testing the Selection Approach*

Considering the search space subdivision in performance and feasibility spaces, this experiment tries to answer the question of which selection approach is more efficient to handle analog circuit candidates towards the optimum space. When two candidate solutions are compared, which one is more efficient, the one satisfying all performance specs less 50% of constraints or the one satisfying all design constraints less the 50% of specs? It will be seen in the following experiments the influence of the selection operator materialized in GENOM by the variation of the sort algorithm and the tournament selection scheme.

The following results, depicted in Table 6.8 and Table 6.9, present the effectiveness of the selection operator variants implemented in GENOM optimizer, using the same circuit of Fig. 6.2a) for the corner optimization case. In particular two variants will be tested. The first variant promotes the solutions close to the performance space, i.e., in the pathway to the solution space, and its first goal is to reach the performance space and then move towards the feasibility space (results in Table 6.8). A second variant uses the opposite strategy, the first approach is to reach the feasibility space and after that the performance space (Table 6.8). The performance of these two approaches will be compared with standard approach (Table 6.10).

**Table 6.8** Output results for each run - Priority to the performance space

| #Run | #nEvals | #Fitness | #wTIME | #PERF | #found_@ | #STATUS | #found_@ |
|---|---|---|---|---|---|---|---|
| 1 | 2578 | 1.060e-02 | 117.07s | Y | 1 (gen) | Y | 14 (gen) |
| 2 | 2720 | 1.061e-02 | 130.58s | Y | 1 (gen) | Y | 15 (gen) |
| 3 | 3440 | 1.061e-02 | 169.89s | Y | 3 (gen) | Y | 20 (gen) |
| 4 | 3888 | 1.060e-02 | 205.95s | Y | 3 (gen) | Y | 23 (gen) |
| 5 | 4144 | 1.060e-02 | 199.57s | Y | 4 (gen) | Y | 25 (gen) |

**Table 6.9** Output results for each run – Priority to the feasibility space

| #Run | #nEvals | #Fitness | #wTIME | #FEAS | #found_@ | #STATUS | #found_@ |
|---|---|---|---|---|---|---|---|
| 1 | 6472 | 1.060e-02 | 296.48s | Y | 1 (gen) | Y | 41 (gen) |
| 2 | 6314 | 1.059e-02 | 311.79s | Y | 1 (gen) | Y | 40 (gen) |
| 3 | 3024 | 1.060e-02 | 158.91s | Y | 1 (gen) | Y | 17 (gen) |
| 4 | 4464 | 1.060e-02 | 220.19s | Y | 1 (gen) | Y | 27 (gen) |
| 5 | 1432 | 1.060e-02 | 73.83s | Y | 1 (gen) | Y | 6 (gen) |

**Table 6.10** Output results for each run – Standard approach

| #Run | #nEvals | #Fitness | #wTIME | #FEAS | #found_@ | #STATUS | #found_@ |
|---|---|---|---|---|---|---|---|
| 1 | 3448 | 5.326e-02 | 190.46s | Y | 1 (gen) | Y | 20 (gen) |
| 2 | 6608 | 5.317e-02 | 318.82s | Y | 1 (gen) | Y | 42 (gen) |
| 3 | 2160 | 5.327e-02 | 125.24s | Y | 1 (gen) | Y | 11 (gen) |
| 4 | 2736 | 5.321e-02 | 121.55s | Y | 1 (gen) | Y | 15 (gen) |
| 5 | 2008 | 5.322e-02 | 89.36s | Y | 1 (gen) | Y | 10 (gen) |

In the standard approach, the best-ranked individual will always be the one with the lowest constraints and specs violation in each generation. From the analysis of these results it is verified that the standard ranking approach and the ranking strategy that gives priority to the solutions satisfying performances spaces produces the better results in terms of number of generations or computation time. In average, both strategies have similar performances (e.g., the average number of generations is 19.4 and 17.8 respectively), although the standard approach presents worse variances from run to run (13.1 against and 4.8 for the other strategy). For simple circuits like the one used in these experiments there is no apparent benefit in these two approaches.

However, for more complex circuits the great variance of standard approach will be amplified and will produce undesirable results, as shown in Table 6.11 and Table 6.12 for the fully differential OpAmp with 21 optimization variables and 43 constraints defined in Sect. 6.5.1.

**Table 6.11** Output results for each run – Priority to the performance space

| #Run | #nEvals | #Fitness | #wTIME | #FEAS | #found_@ | #STATUS | #found_@ |
|---|---|---|---|---|---|---|---|
| 1 | 288 | 7.542e-02 | 16.60s | Y | 7 (gen) | Y | 14 (gen) |
| 2 | 512 | 1.162e-01 | 26.10s | Y | 9 (gen) | Y | 28 (gen) |
| 3 | 1088 | 7.881e-02 | 54.80s | Y | 21 (gen) | Y | 64 (gen) |
| 4 | 608 | 3.428e-02 | 31.99s | Y | 7 (gen) | Y | 34 (gen) |
| 5 | 640 | 9.562e-02 | 33.91s | Y | 10 (gen) | Y | 36 (gen) |
| 6 | 1920 | 7.108e-02 | 93.19s | Y | 12 (gen) | Y | 116 (gen) |
| 7 | 640 | 9.099e-02 | 51.76s | Y | 18 (gen) | Y | 36 (gen) |
| 8 | 832 | 5.907e-02 | 64.20s | Y | 9 (gen) | Y | 48 (gen) |
| 9 | 1168 | 3.139e-02 | 65.11s | Y | 20 (gen) | Y | 69 (gen) |
| 10 | 832 | 1.230e-01 | 41.25s | Y | 7 (gen) | Y | 48 (gen) |

**Table 6.12** Output results for each run – Standard approach

| #Run | #nEvals | #Fitness | #wTIME | #FEAS | #found_@ | #STATUS | #found_@ |
|---|---|---|---|---|---|---|---|
| 1 | 368 | 6.846e-02 | 19.85s | Y | 11 (gen) | Y | 19 (gen) |
| 2 | 1328 | 3.895e-02 | 64.95s | Y | 10 (gen) | Y | 79 (gen) |
| 3 | 448 | 9.689e-02 | 23.30s | Y | 14 (gen) | Y | 24 (gen) |
| 4 | 1616 | 3.544e-02 | 78.14s | Y | 17 (gen) | Y | 97 (gen) |
| 5 | 384 | 1.141e-01 | 20.08s | Y | 14 (gen) | Y | 20 (gen) |
| 6 | 880 | 1.121e-01 | 44.30s | Y | 22 (gen) | Y | 51 (gen) |
| 7 | 2464 | 2.413e+00 | 120.51s | Y | 9 (gen) | N | >150 (gen) |
| 8 | 2464 | 9.955e-01 | 142.98s | Y | 7 (gen) | N | >150 (gen) |
| 9 | 528 | 5.027e-02 | 27.67s | Y | 18 (gen) | Y | 29 (gen) |
| 10 | 2464 | 1.443e+01 | 169.66s | Y | 14 (gen) | N | >150 (gen) |

In several runs, the standard ranking approach is not capable of finding a solution during the specified number of generations (150 in this case) for this nominal optimization problem. The ranking strategy with priority to performance space is able to find a solution in all cases (as noticed in Table 6.11) and, in general, it is more efficient to find a solution in each run.

## 6.4 Comparing GA-STD, GA-MOD and GA-SVM Performance

The objective of these experiments is to compare the performance of the proposed learning method GA-SVM against the earlier evolutionary approach GA-MOD, as well as, the standard GA-STD. The following case studies do not include the search space decomposition feature and the parallelism in the results analysis.

For all the following examples, the industry HSPICE simulator will be used as the evaluation engine, every time an electrical simulation is required. The testbench circuit configuration of Fig. 6.2 b), c) and d) were used in these experiments following the specifications, constraints and models already defined in Sect. 6.2.

In order to create an accurate SVM Feasibility model the optimization parameter space was uniformly sampled with 2000 points to produce the training set, 20% were used to balance the model class samples and 10% more to the validation data set. The class balance pre-processing module was achieved in two steps. First, by filtering those solutions that belong to regions of the design space that are far from fulfill the technological constraints (undersampling the majority class). Then build a two class feasibility model considering those samples which are close the feasibility region and the samples that really belong to the feasibility region. Next, use it to oversample the feasibility region (increasing the minority class) as well as its frontier as explained in Sect. 4.2.5. After that, a final accurate feasibility model is built to be use in the optimization process.

### *6.4.1 GA-STD versus GA-SVM Performance – Filter Case Study*

The filter circuit shown in Fig. 6.2 c) was optimized according to the performance specifications of Table 6.13. The nine design parameters range and the achieved results concerning device sizes are presented in Table 6.14 using the HSPICE simulator as the evaluation engine.

**Table 6.13** Performance specifications/requirements

| SPECs | Initial | GA-STD | GA-SVM | Units |
|---|---|---|---|---|
| Maximum P-Band Ripple | < 1 | 9.13e-01 | 7.20e-01 | dB |
| Minimum P-Band Ripple | > -0.5 | -1.89e-01 | -3.93e-01 | dB |
| Stop Band Attenuation | < -82 | -8.25e+01 | -8.30e+01 | dB |

**Table 6.14** Design parameter specifications (GA-SVM)

| Optimization Parameters | Limits | Results |
|---|---|---|
| R11 (Ω) in block 1 | [1.0e+3, 5.0e+3] | 3.70e+03 |
| C11 (F) in block 1 | [250.0e-12, 400.0e-12] | 3.15e-10 |
| C21 (F) in block 1 | [1.0e-9, 10.0e-9] | 8.00e-09 |
| R12 (Ω) in block 2 | [7.0e+3, 15.0e+3] | 1.13e+04 |
| C12 (F) in block 2 | [250.0e-12, 400.0e-12] | 3.45e-10 |
| C22 (F) in block 2 | [1.0e-9, 5.0e-9] | 3.90e-09 |
| R13 (Ω) in block 3 | [30.0e+3, 40.0e+3] | 3.93e+04 |
| C13 (F) in block 3 | [50.0e-12, 100.0e-12] | 7.40e-11 |
| C23 (F) in block 3 | [1.0e-9, 10.0e-9] | 3.10e-09 |

**Table 6.15** Runtime info

| | GA-STD | GA-SVM |
|---|---|---|
| Optimization Variables | 9 | 9 |
| Number of evaluations to get first solution | 1670 | 1272 |
| Time elapse to get **1st** solution | 75s | 64s |

* In a dual processor core AMD at 2400 MHz running Linux OS.

The obtained performance specs obtained by the GA-STD and GA-SVM methods are included in Table 6.13. Finally, the overall computational times are presented in Table 6.15 and the first solution is the one which satisfies all the performance specs.

Both models GA-SVM and GA-STD obtain feasible solutions as outlined in Fig. 6.4, but with slight differences in time efficiency, about 15-20% of efficiency favorable to GA-SVM, as indicated in Table 6.15. With this optimization methodology the GA algorithm may lose some diversity, however the model will improve dynamically one step after the other, as it can be observed in Fig. 6.5, exploring very well, say aggressively, the performance space.

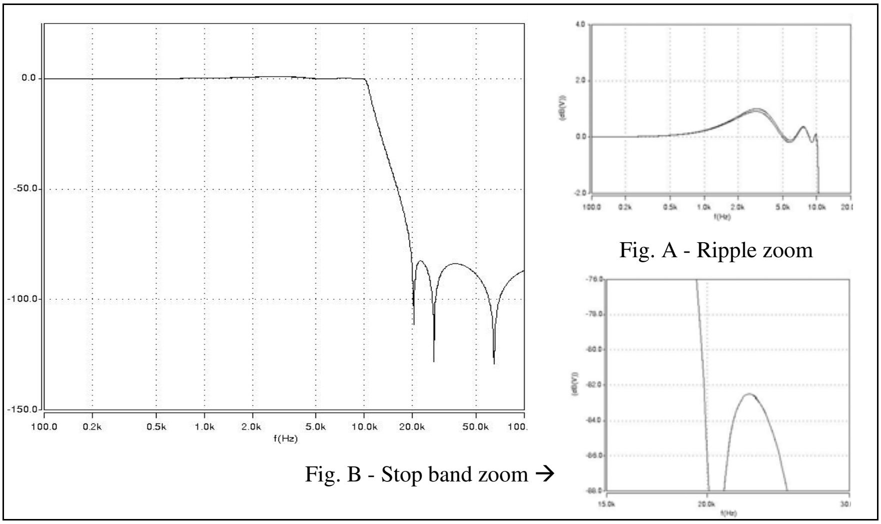


Fig. A - Ripple zoom

Fig. B - Stop band zoom →

**Fig. 6.4** Final Bode plot

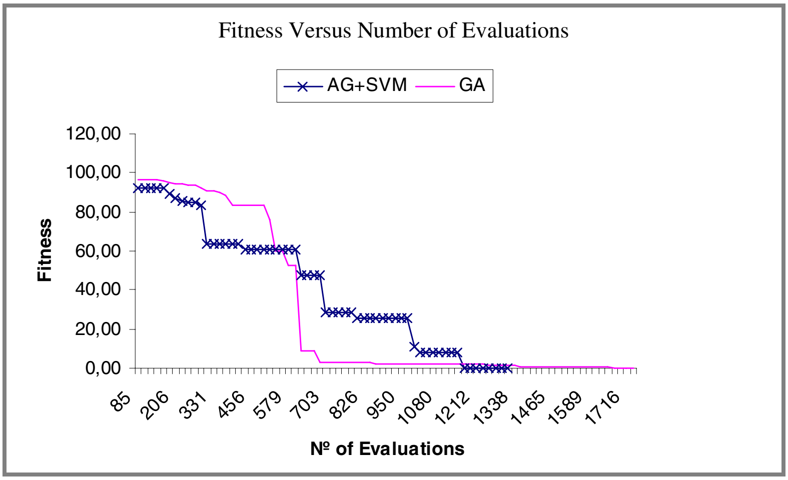


**Fig. 6.5** Performance: GA-STD versus GA-SVM kernel

### *6.4.2 Static GA-SVM Performance - OpAmp Case Study*

In this experiment the Ckt2 and Ckt4 OpAmp circuits shown in Fig. 6.2b) and d) were optimized according to the performance specifications of Table 6.5. All statistics measures presented in Table 6.16 and Table 6.17 are the mean and standard deviation obtained over 20 runs. "Cmean" and "Cstd" stand for the mean and the standard deviation of the cost function; "EVmean" and "EVstd" stand for the mean and the standard deviation of the number of evaluations necessary to get the first solution, and finally, the "Tmean" and "Tstd" represent the mean and the standard deviation of the time spent in the optimization process, not included the setup time to build the model in the case of the GA-SVM algorithm.

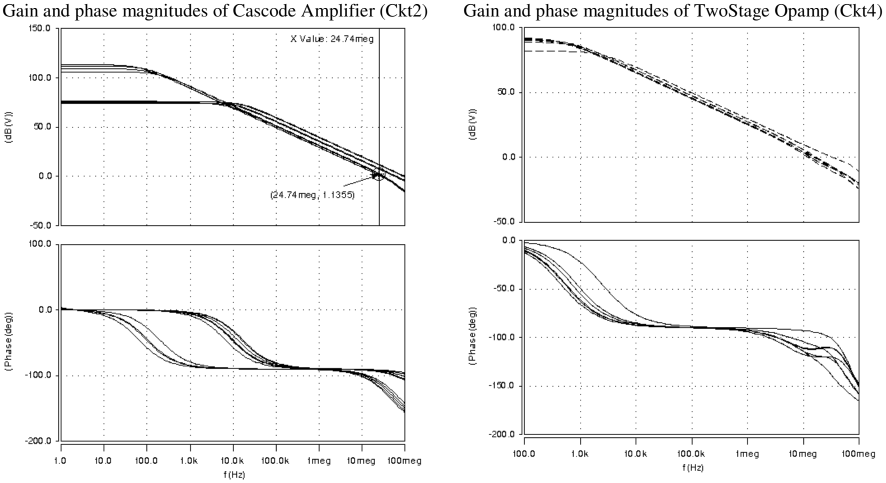
**Table 6.16** Comparison among different algorithms for Ckt2

| Cir-1 | GA-STD | GA-MOD | GA-SVM |
|---|---|---|---|
| Cmean | 9.090e-02 | 7.476e-02 | 7.181e-02 |
| Cstd | 2.128e-02 | 6.940e-03 | 9.646e-03 |
| EVmean | 1.888e+02 | 1.502e+02 | 7.285e+01 |
| EVstd | 8.490e+01 | 7.043e+01 | 2.377e+01 |
| Tmean | 2.026e+00 | 1.669e+00 | 7.275e-01 |
| Tstd | 1.109e+00 | 7.801e-01 | 3.246e-01 |

Fig. 6.6 and Fig. 6.7 show the electrical characteristics of the final population and some of the output reports from the optimization tool, respectively.

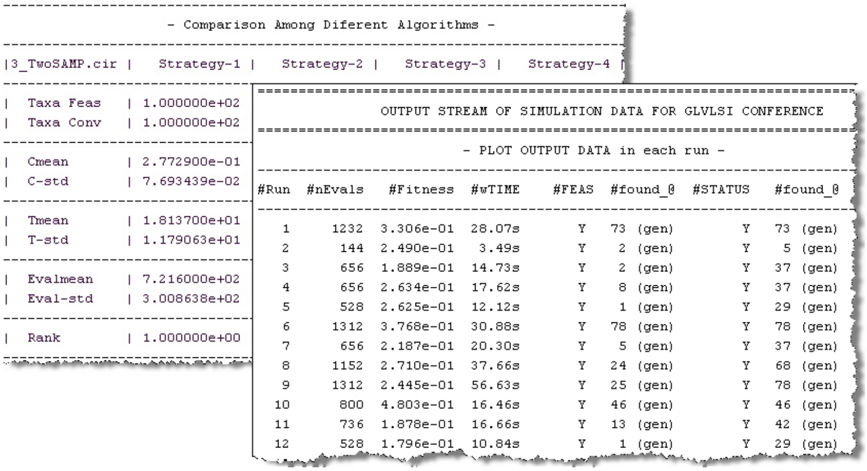
**Table 6.17** Comparison among different algorithms for Ckt4

| Cir-2 | GA-STD | GA-MOD | GA-SVM |
|---|---|---|---|
| Cmean | 2.772e-01 | 2.787e-01 | 2.376e-01 |
| Cstd | 7.693e-02 | 5.066e-02 | 5.034e-02 |
| EVmean | 7.216e+02 | 3.863e+02 | 4.196e+02 |
| EVstd | 3.008e+02 | 1.300e+02 | 1.325e+02 |
| Tmean | 1.813e+01 | 1.216e+01 | 1.029e+01 |
| Tstd | 1.179e+01 | 5.771e+00 | 4.161e+00 |


Gain and phase magnitudes of Cascode Amplifier (Ckt2)

Gain and phase magnitudes of TwoStage Opamp (Ckt4)

**Fig. 6.6** Electrical characteristics from final population


```
--------------------------------------------------------------------
                 - Comparison Among Diferent Algorithms -
--------------------------------------------------------------------
|3_TwoSAMP.cir |   Strategy-1 |   Strategy-2 |   Strategy-3 |   Strategy-4
--------------------------------
|  Taxa Feas    | 1.000000e+02
|  Taxa Conv    | 1.000000e+02
--------------------------------
|  Cmean        | 2.772900e-01
|  C-std        | 7.693439e-02
--------------------------------
|  Tmean        | 1.813700e+01
|  T-std        | 1.179063e+01
--------------------------------
|  Evalmean     | 7.216000e+02
|  Eval-std     | 3.008638e+02
--------------------------------
|  Rank         | 1.000000e+00
--------------------------------
```

```
==================================================================
          OUTPUT STREAM OF SIMULATION DATA FOR GLVLSI CONFERENCE
==================================================================
                     - PLOT OUTPUT DATA in each run -
------------------------------------------------------------------
#Run  #nEvals   #Fitness  #wTIME    #FEAS  #found_@   #STATUS   #found_@
------------------------------------------------------------------
   1     1232  3.306e-01  28.07s       Y   73  (gen)         Y   73  (gen)
   2      144  2.490e-01   3.49s       Y    2  (gen)         Y    5  (gen)
   3      656  1.889e-01  14.73s       Y    2  (gen)         Y   37  (gen)
   4      656  2.634e-01  17.62s       Y    8  (gen)         Y   37  (gen)
   5      528  2.625e-01  12.12s       Y    1  (gen)         Y   29  (gen)
   6     1312  3.768e-01  30.88s       Y   78  (gen)         Y   78  (gen)
   7      656  2.187e-01  20.30s       Y    5  (gen)         Y   37  (gen)
   8     1152  2.710e-01  37.66s       Y   24  (gen)         Y   68  (gen)
   9     1312  2.445e-01  56.63s       Y   25  (gen)         Y   78  (gen)
  10      800  4.803e-01  16.46s       Y   46  (gen)         Y   46  (gen)
  11      736  1.878e-01  16.66s       Y   13  (gen)         Y   42  (gen)
  12      528  1.796e-01  10.84s       Y    1  (gen)         Y   29  (gen)
```

**Fig. 6.7** Output reports from optimization tool (Ckt4)

#### 6.4.2.1 Evaluation Metric

The experiments were executed on AMD X64 2.8 GHz dual core machine and used HSPICE to simulate the circuit and extract performance parameters and the public domain LIBSVM tool [7] as the learning engine. Each algorithm was executed 20 times to acquire the mean and the standard deviation for the evaluation performance. The convergence behavior for the "Two-Stage" OpAmp experiment in one run is presented as an example in Fig. 6.8.

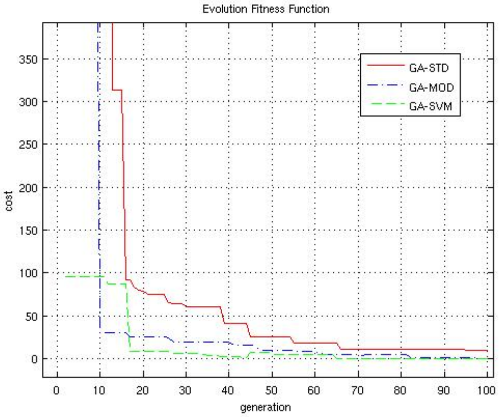


**Fig. 6.8** Evolution of the cost function

Analyzing this Fig. and the experimental data displayed in Table 6.16 and Table 6.17 and Fig. 6.9, it is noticeable the good accuracy and lower variance obtained by the GA-MOD and GA-SVM algorithms.

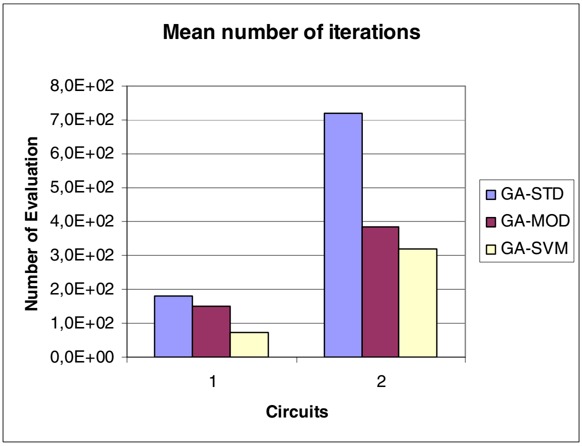


**Fig. 6.9** Comparative graph for the required number of evaluations

However, the GA-SVM achieved better results in all cases. Although the GA-STD uses electrical simulation too, the convergence is slower than the others. The algorithms using models are clearly more time efficient if it is not included the algorithm setup time to build the models. Among all the approaches under test, GA-SVM can achieve the lowest cost and the smaller amount of computation time followed by the GA-MOD.

The setup time to build the model, 100 seconds approx. in each of the presented cases, can be problematic at first sight. The means and variances for the GA-SVM would be very different if they were included in statistics. However some points can be clarified in favor of this approach. First, the initial model is build only once and can be used many times to test different circuit's requirements since the parameters ranges don't change. Second, much of the time spent with model generation is due to the time spent in sampling and evaluation of the selected points for training and testing the model. The effective time to build the model is negligible when compared with circuit model sampling. Thus, the performance and constrained information resulting from each training set is stored, it is possible to build a model at any time, adapted for each circuit requirements and allowing posterior model upgrading and reusability. In conclusion a good compromise between accuracy and efficiency is given by the hybrid GA-SVM approach.

### 6.4.3 Testing the Dynamic GA-SVM Performance

The objective of these experiments is to study the impact of the proposed dynamic SVM model in the optimization process. Our purpose is to compare the performance of several dynamic learning strategies and compare the GA-SVM against the static GA-SVM defined in the previous section, as well as, GA-MOD. Specifically, four experiments defined in Table 6.18 will be performed considering, respectively, the SVM model built before the start of the evolutionary process – static model, SVM model built dynamically, i.e., during evolutionary process, and finally a combination of a static with dynamic SVM model – dynamic model, where the static model is here initialized with a subset of samples from the single static model. Table 6.19 gives the algorithm specifications details.

**Table 6.18** Experiments cases

| *Experiment* | *Model* | *SVM* | *Static Model* | *Dynamic Model* |
|---|---|---|---|---|
| Exp-1 | GA-MOD | No | No | No |
| Exp-2 | Static-SVM | Yes | Yes/3000(a) | No |
| Exp-3 | Dyn-SVM | Yes | No | Yes/100 |
| Exp-4 | S+D-SVM | Yes | Yes/1000 | Yes/100(b) |

(a) Number of uniform sampling points (b) Regeneration rate

**Table 6.19** Algorithm specifications under test

| *Algorithm* | *GA-MOD* | *Static-SVM* | *Dynamic-SVM* |
|---|---|---|---|
| Selection | 2-Tournament | 2-Tournament | 2-Tournament |
| Crossover | 2-Point Unif | 2-Point Unif | 2-Point Unif |
| Mutation Rate | 5% Dynamic | 5% Dynamic | 5% Dynamic |
| Kernel type | GA modified | SVM-RBF | SVM-RBF |
| Training Set | None | 3000 Unif Sampling points | None |

These experiments use exclusively the two-stage (Ckt4) ampop illustrated in Fig. 6.2, updated with appropriate test benches to allow the measures of the desired performances parameters. All experiments used the same computation resources, specifications and constraints as earlier and also used the same number of runs to extract the mean and the standard deviation for the evaluation performance. The convergence behavior for the two-stage OpAmp experiment in one run is presented as an example in Fig. 6.10.

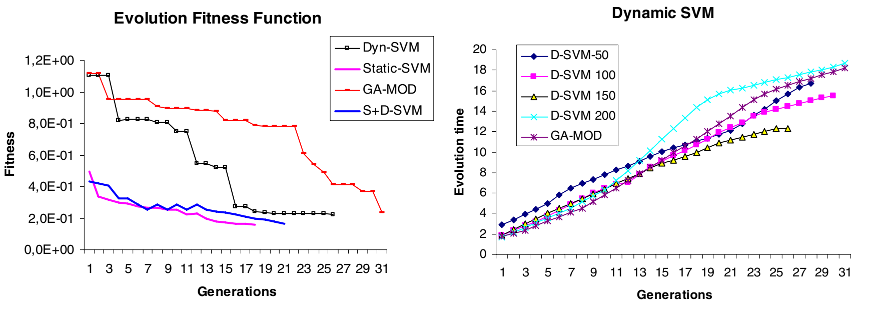
From the experimental data, displayed from Table 6.20 and Fig. 6.10, it is clear the good accuracy and time efficiency obtained with strategies embedded with SVM models built in offline mode. However, the overhead time to build the static model can be problematic for more complex circuits. Here, the static algorithm takes about 90 seconds approx. to evaluate 3000 uniform samples but in more complex circuits, this number rises considerably. The means and variances to setup the models using static modeling were not included in the final statistics given at Table 6.20.

**Table 6.20** Comparison among different algorithms

| *Algorit.* | *Cmean* | *Cstd* | *EVmean* | *EVstd* | *Tmean* | *Tstd* |
|---|---|---|---|---|---|---|
| Exp-1 | 2.55e-01 | 4.47e-02 | 3.95e+02 | 1.07e+02 | 1.14e+01 | 4.04e+00 |
| Exp-2 | 2.61e-01 | 4.93e-02 | 1.48e+02 | 1.08e+02 | 3.91e+00 | 2.28e+00 |
| Exp-3 | 2.19e-01 | 4.74e-02 | 2.74e+02 | 2.24e+02 | 6.93e+00 | 6.58e+00 |
| Exp-4 | 2.19e-01 | 5.29e-02 | 6.22e+02 | 1.73e+02 | 1.42e+01 | 4.31e+00 |

A different strategy has been taken towards a dynamic building model with data gathered during the early generations. Some configurations were tested as shown in the Fig. 6.10 (b).

This approach can be very sensitive to the value of the regeneration rate value. Using a lower value for the regeneration rate, e.g., 200, originates long processing times because it takes more training samples however a better accuracy model is obtained. A higher sampling rate at early generations causes better convergence but with a slightly increase in execution times. An automatic and dynamic control of the regeneration rate can be added using the information of the quality of SVM model. A good compromise between these two approaches is given by the test case joining the static and dynamic training model behavior (S+D-SVM).



(a) Cost function evolution

(b) Dynamic SVM model behavior with regeneration rate

**Fig. 6.10** Comparative performance analysis

### 6.4.4 *Final Comments*

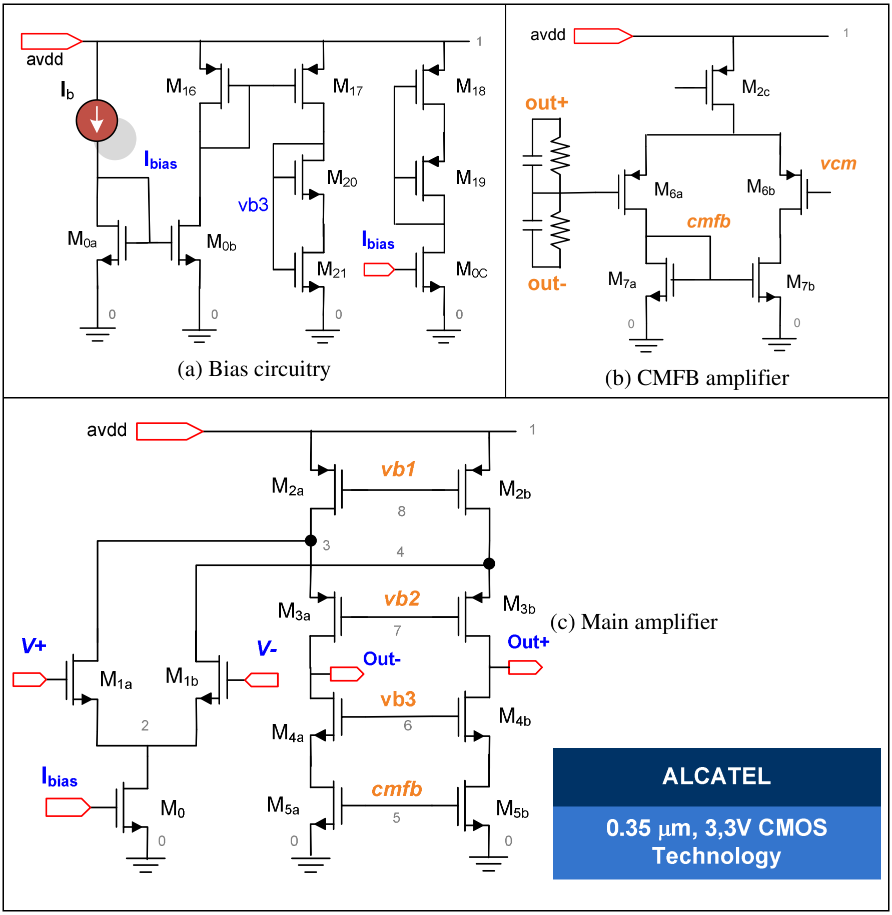
The proposed approach is a step forward when compared with the simple GA kernel, as it now incorporates performance modeling facilities, allowing an effective pruning of the candidate solutions before being submitted to the heavy time-consumption task of electrical evaluation. The achieved results show significant gains in efficiency and this approach also allows the reuse of the model generated during one optimization process in subsequent optimizations, which is again another significant advantage when compared with traditional approaches, especially in the areas of architecture exploration and synthesis of complex analog blocks.

## 6.5 General Purpose Circuits or High Performance Circuits Design

In this section, a case study for several high performance circuit designs will be presented passing by the following phases: full schematic, design specifications and constraints, variable ranges, optimization results such as variables size and achieved performance and time statistics. This set of circuits shows GENOM’s ability to design high-performance and novel circuit topologies. The design complexity decomposition was optionally not taken into account because the primarily objective is to test the algorithm not the design process.

### *6.5.1 Fully Differential OpAmp*

Fig. 6.11 illustrates the differential amplifier schematics considered to evaluate the performance of the presented optimization technique.



(a) Bias circuitry

(b) CMFB amplifier

(c) Main amplifier

**Fig. 6.11** Differential amplifier schematic

The topology, defining the connectivity of device-level components, consists of 25 transistors devices grouped in 3 main functional blocks: the main amplifier with differential input and output, the bias circuit and the common mode feedback circuitry. By looking at the circuit schematic, some groups of transistors like M3a and M3b, for instance, must be matched. Some dependent relations like, the multiplicity factor, m7, of transistors M7a and M7b is equal to m5/2 (this implies that m5 must be pair) must also be verified.

#### 6.5.1.1 Performance Specifications, Input Variables Ranges and Design Space Size

The main objective was to synthesize the presented differential amplifier using the ALCATEL 0.35µm, 3.3V CMOS technology according to the performance specifications, listed in Table 6.21, and always respecting the fundamental designer rules related to overdrive voltages and drain-sources voltages. The 7 performance constraints derived from Table 6.21 (excluding *CL*) and the 34 constraints derived from designer's rules depicted in Table 6.23, result in 41 optimizations constraints that must be satisfied by the optimization process. The 34 constraints are due to the 17 overdrive voltage and 17 drain source voltages considered on transistors *m0a*, *m0b*, *m16*, *m1a*, *m1b*, *m2a*, *m2b*, *m3a*, *m3b*, *m4a*, *m4b*, *m5a*, *m5b*, *m6a*, *m6b*, *m7a* and *m7b*.

A total of 21 independent variables (column "*Design Variables*" in Table 6.23) corresponding to widths, lengths and multiplicity factor of transistors represent the number of genes on each genetic algorithm chromosome. All the solutions were examined for each one of the 9 corner points resulting from the cross combination of process and operational variation listed on Table 6.22. For example, the combination (CSlow,-40º) means a circuit analysis at temperature -40º using NMOS and PMOS slow models. Then, it is followed by (CSlow,+50º) analysis, etc. Finally, Table 6.24 lists the main optimization parameters used on the genetic algorithm.

**Table 6.21** Performance parameter specifications

| | Specifications | Target | Units | Description |
|---|---|---|---|---|
| **Electrical** | GBW | > 100 | MHz | Unit-gain frequency |
| | Phase margin | > 60 | º | Phase margin |
| | DC gain | > 55 | dB | DC gain |
| | CMF GBW | > 50 | MHz | CMFB unit-gain frequency |
| | CMF Phase margin | > 60 | º | CMFB phase margin |
| **Environmental** | CL (fixed value) | 0.2 | pF | Capacitive Load |
| **Optimization** | Power Consumption | Minimum | mW | Objective |
| | Current Consumption | Minimum | µA | Objective |

**Table 6.22** Corners analysis data

| Conditions | Variation points | | |
|---|---|---|---|
| MOS worst case parameters | CSlow | CTyp | CFast |
| Temperature Range (º C) | -40º C | +50º C | +120ºC |

**Table 6.23** Matching and constraints details

| **Matching** | | | | **Constraints** | | |
|---|---|---|---|---|---|---|
| Dependent Variable | Design Variable | Range [Min;Max;Step] | Unit | $V_{GS}$ - $V_{T}$ (a) [Min - Max] | $V_{DS}$ - $V_{DSAT}$ (b) Min / Max | Unit |
| M0 (_w00, _l00, m02) | _w00 | [1; 20; 1] | µm | | | |
| M0a=M0b=M0c (_w02, _l00, 1) | _w01 | [1; 20; 1] | µm | [100 - 300] | >100 | mV |
| M1 (_w02, _l06, m06) | _w02 | [1; 20; 1] | µm | | | |
| M2a = M2b=M2c (_w02, _l02, m02) | _w04 | [1; 20; 1] | µm | [100 - 300] | >100 | mV |
| M20 (_w04, _l04, 1) | _w10 | [1; 20; 1] | µm | | | |
| M21 (_w11, _l11, 1) | _w11 | [1; 20; 1] | µm | | | |
| M3a = M3b (_w02, _l03, m03) | _l00 | [0.35; 10; 0.05] | µm | [100 - 300] | >100 | mV |
| M5a = M5b (_w04, _l05, m05) | _l01 | [0.35; 10; 0.05] | µm | [100 - 300] | >100 | mV |
| M6a = M6b (_w02, _l06, m06) | _l02 | [0.35; 10; 0.05] | µm | [50 - 300] | >100 | mV |
| | _l03 | [0.35; 10; 0.05] | µm | | | |
| **Bias Circuit** | _l04 | [0.35; 10; 0.05] | µm | | | |
| M16=M17 (_w02, _l02, 1) | _l05 | [0.35; 10; 0.05] | unit | [100 - 300] | >100 | mV |
| M18 (_w10, _l10, 1) | _l06 | [0.35; 10; 0.05] | µm | | | |
| M19 (_w02, _l03, 1) | _l10 | [0.35; 10; 0.05] | µm | | | |
| M4a=M4b (_w04, _l04, m04) | _l11 | [0.35; 10; 0.05] | µm | [100 - 300] | >100 | mV |
| M7a=M7b (_w04, _l05, m05/2) | _m01 | [1; 80;1] | unit | [100 - 300] | >100 | mV |
| M1a=M1b (_w01, _l01, m01) | _m02 | [1; 80;1] | unit | [50 - 300] | >100 | mV |
| | _m03 | [1; 80;1] | unit | | | |
| | _m04 | [1; 80;1] | unit | | | |
| | _m05 | [1; 80;1] | unit | | | |
| | _m06 | [1; 80;1] | unit | | | |

(a) Technology Constraints - overdrive voltages (b) Drain-sources voltages

**Table 6.24** Optimization algorithm parameters

| **Parameter** | **value** | **Parameter** | **value** | **Parameter** | **value** |
|---|---|---|---|---|---|
| **Kernel** | GA-MOD | **Selection** | Tournament | **Popsize** | 64 |
| **Strategy** | Corner Optimization | **Crossover** | Two point | **Init Pop** | 2*Popsize |
| **Sampling** | LHS | **Mutation** | Dynamic | **Generations** | 150 |
| **Sort** | Priority to performance fitness then perf. constraints. | **Adaptive** | No | **Stop condition** | End of generations |
| | | **Elite** | 25% of populat. | **Search** Space | 2,370e+37 |

### 6.5.1.2 Analysis

The attached test benches used for DC and AC simulations are illustrated in Fig. 6.12. The unusual values for the resistance (1T Ohm) and for the capacitance (1F) ensure the same voltage in DC Analysis for nodes *Vin-*, *Vin+*, *Voutp* and *Voutn*, it is also possible to analyze the amplifier open loop gain. A dependent source voltage is used to transform a differential output (*voutp*, *voutn*) into a single ended one (*voutd*).

The simulation results for the main amplifier and *cmfb* circuit sizing achieved with the optimization module, and using the HSPICE simulator as the evaluation engine are presented in Table 6.25 and satisfy all the design requirements. The final transistor dimensions are displayed in Table 6.26. The proper biasing of all CMOS transistors are guaranteed once the final solution satisfies all the design specs and functional constraints for each of the corner points. The computational times were included, in Table 6.27, to illustrate the effectiveness of the proposed system.

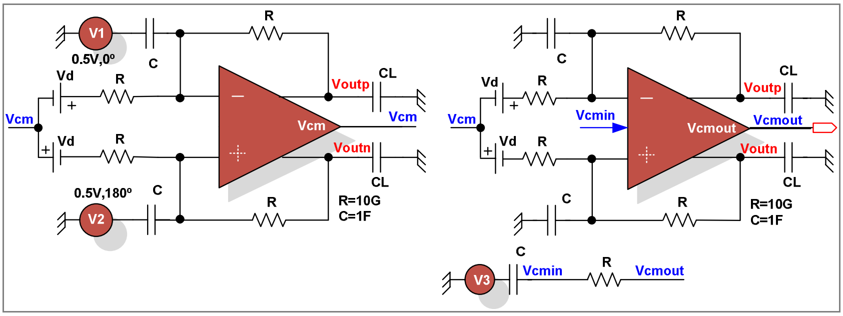


**Fig. 6.12** Testbench for (a) AC and (b) AC Common mode feedback specifications

**Table 6.25** Performance parameter specifications

| | Specifications | Target | Sizing Result | Units |
|---|---|---|---|---|
| **Electrical** | GBW | > 100 | 158.0 | MHz |
| | Phase margin | > 60 | 65.0 | º |
| | DC gain | > 55 | 66.6 | dB |
| | CMF GBW | > 50 | 64.1 | MHz |
| | CMF Phase margin | > 60 | 75.6 | º |
| **Optimization** | Power Consumption | Minimum | 4.2 | mW |
| | Current Consumption | Minimum | 1.2 | mA |

**Table 6.26** Final transistor dimensions

| **Main Amplifier** | **W/L (μm/μm)** | **Bias** | **W/L (μm/μm)** | **Common Mode** | **W/L (μm/μm)** |
|---|---|---|---|---|---|
| M0 | 54/0.40 | M0a, b, c | 1/0.40 | M6a, b | 170/0.95 |
| M1a, M1b | 41/0.95 | M16,17 | 10/0.40 | M7a, b | 45/1.75 |
| M2a, M2b | 90/0.40 | M18 | 3/0.95 | M2c | 90/0.40 |
| M3a, M3b | 170/0.90 | M19 | 10/0.90 | | |
| M4a, M4b | 441/7.80 | M20 | 9/7.80 | | |
| M5a, M5b | 54/0.95 | M21 | 6/4.40 | | |

**Table 6.27** Runtime info*

| **Design Problem** | | | |
|---|---|---|---|
| Opt. Variables / Constraints (Specs + Design Const.) | 21 | 7 + 34 = 41 | |
| | | | |
| **(1 -Step) Corners Optimization** | **Time** | **#Generation** | **#Evaluations** |
| Overall Optimization time | 17m05s | 150 | 2496 |
| First Feasible Solution | 10m56s | 79 | 1329 |
| Best feasible solution | 12m45s | 92 | 1616 |

* In a single processor Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz PC running Linux.

The next pages show all graphical and numerical results for the AC corner analysis. Fig. 6.13 shows all the gain magnitudes, it is interesting to observe the range of DC gain and GBW; all corner numerical results are reported in Table 6.28 while Table 6.29 shows minimum and maximum values.

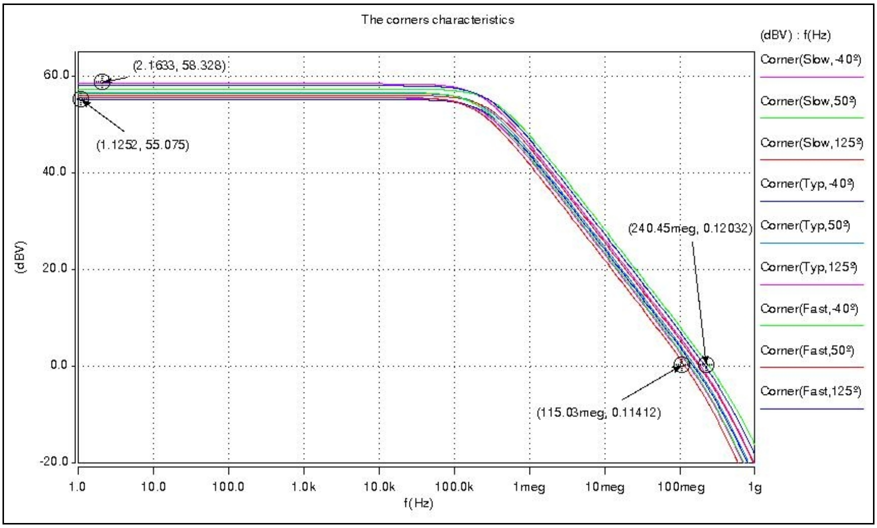


**Fig. 6.13** Gain magnitudes for corners analysis

Fig. 6.14 displays the output from corner simulation.

```
CORNER   MODEL  TEMP   WEIGHT  SATISFY   FITNESS(i)  VIOL(i)     SUM_FIT[10150]
[ 1 ]    CSLOW  -40º   1       7 / 34=41 6.779e-03   0.000e+00      6.779e-03
[ 2 ]    CSLOW  +50º   1       7 / 34=41 1.333e-02   0.000e+00      1.333e-02
[ 3 ]    CSLOW  +120º  1       7 / 34=41  1.980e-02  0.000e+00      1.980e-02
[ 4 ]    CTYP   -40º   1       7 / 34=41 3.438e-02   0.000e+00      3.438e-02
[ 5 ]    CTYP   +50º   1       7 / 34=41  4.830e-02  0.000e+00      4.830e-02
[ 6 ]    CTYP   +120º  1       7 / 34=41 6.199e-02   0.000e+00      6.199e-02
[ 7 ]    CFAST  -40º   1       7 / 34=41  7.012e-02  0.000e+00      7.012e-02
[ 8 ]    CFAST  +50º   1       7 / 34=41 7.774e-02   0.000e+00      7.774e-02
[ 9 ]    CFAST  +120º  1       7/ 34=41   8.518e-02  0.000e+00      8.518e-02
****** EUREKA ******
Byebye. AIDA - IC_DESIGN Terminate ...
Job done on a  Intel(R) Core(TM)2 Quad CPU Q6600  @ 2.40GHz
```

**Fig. 6.14** Output from simulation where all corners are satisfied

**Table 6.28** Results for corners analysis

| **Corner** | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| Process | | slow | | | typical | | | fast | | |
| Temperature | | -40° | 50° | 125° | -40° | 50° | 125° | -40° | 50° | 125° |
| **Specs** | **Values** | | | | | | | | | |
| DC Gain (dB) | > 55 | 58.3 | 56.5 | 55.5 | 57.9 | 56.3 | 55.4 | 57.2 | 55.9 | 55.3 |
| f (A=0dB) (MHz) | > 100 | 188 | 138 | 116 | 215 | 158 | 133 | 243 | 178 | 150 |
| Phase (A=0dB) (°) | >-120 | -117 | -116 | -115 | -116 | -115 | -115 | -114 | -114 | -114 |
| PM (grade) | > 60 | 63 | 64 | 65 | 64 | 65 | 65 | 66 | 66 | 66 |

**Table 6.29** Minimum and maximum values for AC corner analysis

| **Specs Range** | | | | |
|---|---|---|---|---|
| DC Gain (dB) | Min: | 55.3 dB | Max: | 58.3 dB |
| GBW (MHz) | Min: | 116 MHz | Max: | 243 MHz |
| PM (grade) | Min: | 63º | Max: | 66º |

Two critical corner points are pointed in Fig. 6.13. The corner in the bottom (magnitude 0) is achieved by Corner Slow, @125º and in the top by Corner Fast, @125º. To calculate the phase margin is not useful to plot all phases in the same graphic; Fig. 6.15 shows the gain magnitude and phase only for typical mean process and 50° conditions. The dot line depicts the gain magnitude and phase at the common mode output.

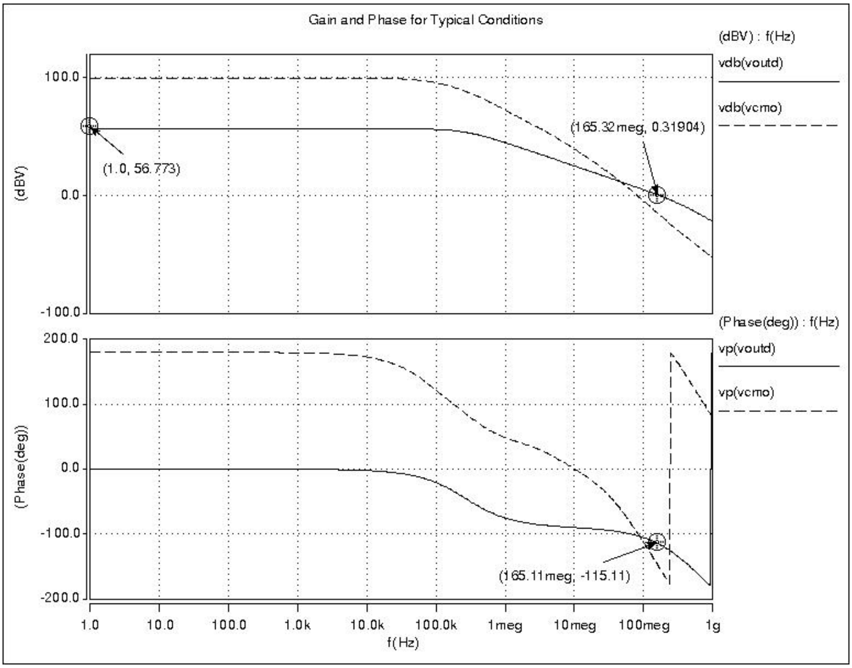


**Fig. 6.15** Gain magnitude and phase for typical conditions

### 6.5.1.3 Design Analysis

The GENOM optimization algorithm solves the circuit sizing problem with efficiency considering the type of optimization evolved in this experiment, the corner optimization and taking also in consideration the number of optimization variables and constraints. The first and the final solutions produced are presented in Table 6.30 and Table 6.31.

**Table 6.30** First feasible solution performance parameter specifications

| Specifications | Target | Sizing Result | Units |
|---|---|---|---|
| GBW (MHz) | > 100 MHz | 110.6 | MHz |
| Phase margin (deg) | > 60º | 74.0 | º |
| DC gain (dB) | > 55 dB | +61.1 | dB |
| CMF GBW (MHz) | > 50 MHz | 60.7 | MHz |
| CMF Phase margin (deg) | > 60º | +82.4 | º |
| Power Consumption (mW) | Minimum | 2.8 | mW |
| Current Consumption (μA) | Minimum | 8.7e-01 | mA |

Nº Eval = 1329 RealTime: 10min

**Table 6.31** Best solution performance parameter specifications

| Specifications | Target | Sizing Result | Units |
|---|---|---|---|
| GBW (MHz) | > 100 MHz | 189.38 | MHz |
| Phase margin (deg) | > 60º | 64.1 | º |
| DC gain (dB) | > 55 dB | 58.1 | dB |
| CMF GBW (MHz) | > 50 MHz | 75.6 | MHz |
| CMF Phase margin (deg) | > 60º | 84.8 | º |
| Power Consumption (mW) | Minimum | 2.1 | mW |
| Current Consumption (µA) | Minimum | +8.8e-01 | mA |

Nº Eval = 1626 RealTime: 12min

The power consumption is the power provided by the power supply (vdd) as defined in the HSPICE expression (6.1). The current consumption is defined by the expression (6.2), where avddpar is the supply voltage (3.3V). Both expressions are divided by two in order to reflect the differential status of this topology.

```
.MEASURE  AC 'power'  PARAM('-P(vdd)/2')
```
(6.1)

```
.MEASURE AC 'iavdd'  PARAM('-P(vdd)/avddpar/2')
```
(6.2)

## *6.5.2 A Common OTA Fully Differential Telescopic OpAmp*

### 6.5.2.1 Description

A common OTA (Operational Transconductance Amplifier) is the telescopic amplifier. The major drawback of this amplifier's topology is the reduced output swing when compared with other solutions, such as the folded cascade or two stage amplifiers, which becomes relevant in low voltage applications. On the other hand, its good speed performance associated with its low power consumption turns this topology into a competitive implementation. The schematic represented in Fig. 6.16 is an in-house fully differential version of this topology.

The topology consists in 24 transistors grouped in 2 main functional blocks: the main amplifier with differential input and the bias circuit. A quick inspection to circuit schematic highlights the potential matching of some groups of transistors like *M0* and *M19*, *M40* and *M43*, *M17* and *M18*, *M34* and *M36*. Some dependent relations like for instance, the multiplicity factor of transistors *M18*, *M17*, *M24* and *M5* and others listed in Table 6.33 should also be checked.

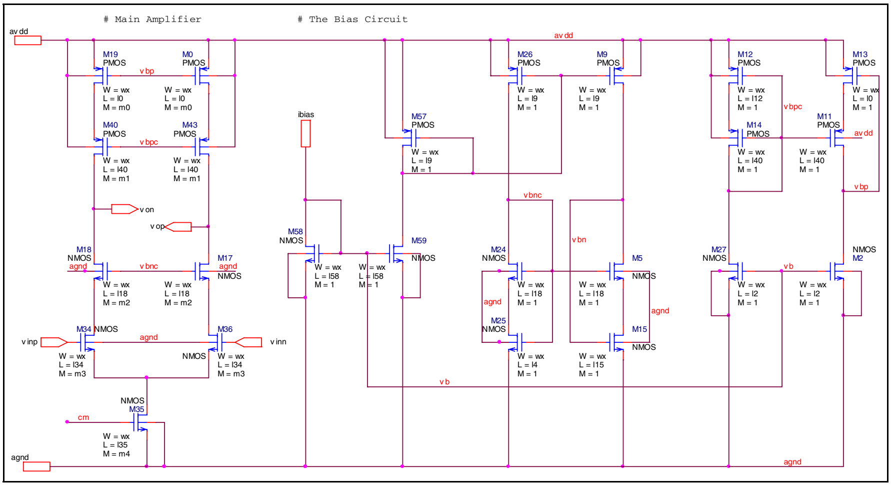


**Fig. 6.16** Telecopic OpAmp - Main amplifier and Bias circuitry

### 6.5.2.2 Problem Specifications and Design Configurations

The main objective was to synthesize the presented telescopic amplifier, using the *UMC 0.18μm* logic 1.8V Generic II process, according to the performance specifications listed in Table 6.32, and designed to follow the fundamental designer rules and optimization design constraints of Table 6.33. The total number of constraints, performance constraints and the constraints derived from designer's rules are composed by 23 optimizations constraints that must be satisfied for the optimization process described in Table 6.35. The specifications must be satisfied for the corners points of Table 6.36.

**Table 6.32** Performance parameter specifications

| | Specifications | Target | Units | Description |
|---|---|---|---|---|
| **Electrical** | DC gain | > 75 | dB | DC gain |
| | GBW | > 100 | MHz | Unit-gain frequency |
| | Phase Margin | [ 60-90] | º | Phase margin |
| **Environmental** | Capacitive Load | 1.1 | pF | Capacitive Load |
| | Wi – fixed widths | 2 | μm | Fixed all widths |
| **Optimization** | Power Consumption | Minimum | mW | Objective |
| | Current Consumption | Minimum | μA | Objective |

**Table 6.33** Matching and constraints details

| **Matching** | | | | **Constraints** | | |
|---|---|---|---|---|---|---|
| Dependent Variable | Design Variable | Range [Min Max;Step] | Unit | $V_{GS} - V_{T}$ (a) [Min - Max] | $V_{DS} - V_{DSAT}$ (b) Min / Max | Unit |
| M19 – M0 | _m0 | [1; 100; 2] | unit | [100 - 200] | [50 - 150] | mV |
| M19 – M0 | _l0 | [0.18; 10; 0.05] | µm | | | |
| M40 – M43 | _m1 | [1; 100; 2] | unit | [100 - 200] | >50 | mV |
| M40 – M43 | _l40 | [0.18; 10; 0.05] | µm | | | |
| M18 – M17 | _m2 | [1; 100; 2] | unit | [100 - 200] | >50 | mV |
| M18 – M17 | _l18 | [0.18; 10; 0.05] | µm | | | |
| M34 – M36 | _m3 | [1; 100; 2] | unit | [50 - 200] | [50 - 150] | mV |
| M34 – M36 | _l34 | [0.18; 10; 0.05] | µm | | | |
| M35 | _m4 | [1; 100; 2] | unit | [100 - 200] | >50 | mV |
| M35 | _l35 | [0.18; 10; 0.05] | µm | | | |
| **Bias Circuit** | | | | | | |
| M24 – M5 | _l18 | [0.18; 10; 0.05] | µm | | | |
| M59 – M58 | _l58 | [0.18; 10; 0.05] | µm | | | |
| M9 – M57 – M26 | _l19 | [0.18; 10; 0.05] | µm | | | |
| M11 – M14 | _l40 | [0.18; 10; 0.05] | µm | | | |
| M13 | _l0 | [0.18; 10; 0.05] | µm | | | |
| M12 | _l12 | [0.18; 10; 0.05] | µm | | | |
| M27 – M2 | _l2 | [0.18; 10; 0.05] | µm | | | |
| M15 | _l15 | [0.18; 10; 0.05] | µm | | | |
| M25 | _l25 | [0.18; 10; 0.05] | µm | | | |

(a) Technology Constraints - overdrive voltages (b) Drain-sources voltages

Table 6.34 explains the rationale behind the achieved constraints values used in this experiment. In a fully differential amplifier, as the one shown in Fig. 6.16, the amplifier can be designed in two symmetrical parts. When one transistor changes value, its mirror also changes. This principle is used for the input differential pair, the cascode and load transistors. As for the overdrive voltage and margin, the constraints are as follows:

**Table 6.34** Matching and constraints details

| | **Overdrive voltage** | **Margin** |
|---|---|---|
| | Vgs-Vt = Vov | Vds-VDsat |
| differential pair | 50mV> Vov >200mV | > 50mV |
| current sources | 50mV>Vov>200mV | > 50mV |
| cascodes | 50mV>Vov>200mV | > 50mV |
| current sources with cascodes | 50mV>Vov>200mV | 50mV> Margin > 200mV |

**Table 6.35** Optimization algorithm parameters

| Parameter | Value | Parameter | value | Parameter | value |
|---|---|---|---|---|---|
| **Kernel** | GA-MOD | **Selection** | Tournament by "feasibility" | **Popsize** | 64 |
| **Strategy** | Typical + Corner Optimization | **Crossover** | Two point | **Init Pop** | 2*Popsize |
| **Sampling** | LHS | **Mutation** | Dynamic | **Generations** | 150 |
| **Sort** | Priority to performance fitness then performance constraints. | **Adaptive** | No | Stop | End of generations |
| | | **Elite** | 25% of population | **Search Space domain** | 2.344e+35 |

**Table 6.36** Corner analysis data

| Conditions | Variation points | | |
|---|---|---|---|
| MOS worst case parameters | SF-Slow | TT-Typ | FS-Fast |
| Temperature Range (° C) | -40° C | +50° C | +120ºC |

Where, SF, TT and FS means the Slow/Fast, Typical/Typical and Fast/Slow process, respectively. Instead of using the typical fast and slow device models sets, where all devices are supposed to be fast or slow, a mixture of slow nMOS devices and fast pMOS is here considered, for example purposes, namely the SF, TT and FS meaning the Slow/Fast, Typical/Typical and Fast/Slow process, respectively.

### 6.5.2.3 Analysis

The attached test bench circuit used for DC and AC simulations is illustrated in Fig. 6.17. A dependent source voltage is used to transform a differential output (*out1*, *out2*) into a single ended one (*outd*).

This experiment was executed on a single Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz dual core machine and use HSPICE to simulate the circuit and extract performance parameters. The simulation results of the main amplifier and bias circuit sizing are shown in Table 6.37. The final transistor dimensions for all the devices and biasing conditions resulting from the sizing process are displayed in Table 6.38.

Fig. 6.18 shows the gain magnitude and phase for typical process and 50°C conditions.

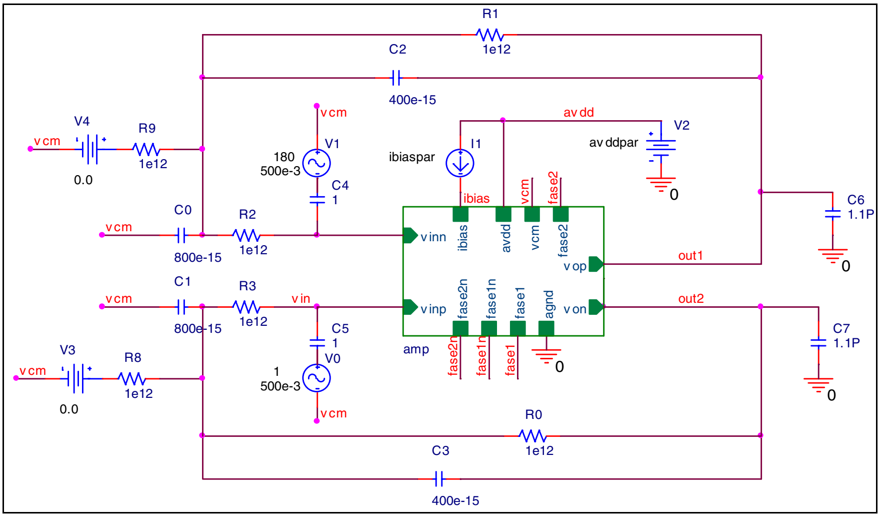


**Fig. 6.17** Telescopic OpAmp - Testbench for DC and AC specifications

**Table 6.37** Performance parameter specifications

| | Specifications | Target | Sizing Result | Units |
|---|---|---|---|---|
| **Electrical** | DC gain | > 75 | 77.6 | dB |
| | GBW | > 100 | 123.0 | MHz |
| | Phase margin | > 60 | 65.0 | º |
| **Optimization** | Power Consumption | Minimum | 5.6e-01 | mW |
| | Current Consumption | Minimum | 3.1e-01 | mA |

**Table 6.38** Final transistor dimensions

| **Main Amplifier** | W/L (μm/μm) | **Bias** | W/L (μm/μm) |
|---|---|---|---|
| M19 – M0 | 202 / 1.080e-06 | M24 – M5 | 2 / 1.33e-06 |
| M40 – M43 | 152 / 1.58e-06 | M59 – M58 | 2 / 1.38e-06 |
| M18 – M17 | 126 / 1.33e-06 | M9 – M57 – M26 | 2 / 4.03e-06 |
| M34 – M36 | 60 / 0.73e-06 | M11 – M14 | 2 / 1.58e-06 |
| M35 | 10 / 0.18e-06 | M13 | 2 / 1.08e-06 |
| | | M12 | 2 / 8.53e-06 |
| | | M27 – M2 | 2 / 8.63-06 |
| | | M15 | 2 / 1.73e-06 |
| | | M25 | 2 / 9.68e-06 |

Note: M0c belongs to Bias and have the same value that M0a and M0b.

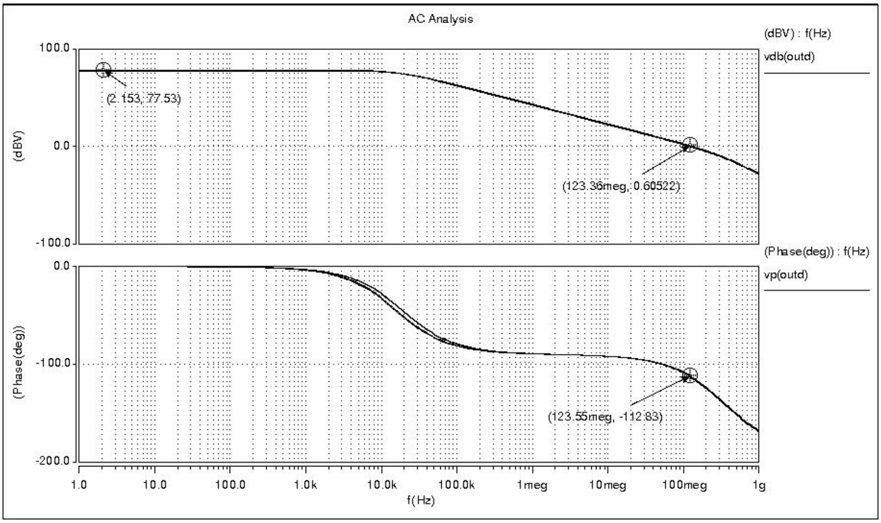


**Fig. 6.18** Gain magnitude and phase for typical conditions

As it can be noticed, this simulation design meets the required specs related to DC gain, gain bandwidth and phase margin satisfying all corners points as reported in Table 6.39, while Table 6.40 shows the minimum and maximum values. Obviously the amplifier was designed in order to obtain a worst case *DC gain* bigger than 75dB and a *GBW* bigger than 110MHz.

**Table 6.39** Numerical results for corner analysis

| **Corner** | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| **Process** | | Slow-Fast | | | Typical | | | Fast-Slow | | |
| **Temperature** | | -40° | 50° | 125° | -40° | 50° | 125° | -40° | 50° | 125° |
| **Specs** | **Values** | | | | | | | | | |
| DC Gain (dB) | > 55 | 78.3 | 76.5 | 75.5 | 78.9 | 77.6 | 76.4 | 77.2 | 75.9 | 75.1 |
| f (A=0dB) (MHz) | > 100 | 178 | 128 | 110 | 205 | 123 | 133 | 223 | 146 | 140 |
| Phase (A=0dB) (°) | >-120 | -117 | -116 | -115 | -116 | -115 | -115 | -114 | -114 | -114 |
| PM (grade) | > 60 | 63 | 64 | 65 | 64 | 65 | 65 | 66 | 66 | 66 |

**Table 6.40** Minimum and maximum values for AC corner analysis

| **Specs Range** | | | | |
|---|---|---|---|---|
| DC Gain (dB) | Min: | 75.1 dB | Max: | 78.9 dB |
| GBW (MHz) | Min: | 110 MHz | Max: | 223 MHz |
| PM (grade) | Min: | 63º | Max: | 66º |

#### 6.5.2.4 Design Analysis

Taking into consideration the type of optimization evolved in this experiment, the corner optimization of 16 optimization variables and 24 constraints, the optimization algorithm solves the problem with efficiency. In this experiment the two step evolutionary algorithm was used, which increases the computation efficiency as shown in Table 6.41, once the optimization algorithm achieves a promising solution using the typical optimization. After that, the optimization follows the corner analysis process. The switch between these two steps is when five solutions are found by the typical process, in such a way that the population is moderately populated with promising samples. This approach increases the computation efficiency once the same problem was not able to produce a feasible solution, within the same time constraint, when a single corner optimization was considered. The first feasible solution satisfying all corners was achieved in generation 102.

**Table 6.41** Runtime info

| **Design Problem** | | | |
|---|---|---|---|
| Opt. Variables / Constraints (Specs + Design Const.) | 16 | 5 + 18 = 24 | |
| **(1-Step) Typical Optimization** | **Time** | **#Generation** | **#Evaluations** |
| Overall Optimization time | 2m12s | 53 | 944 |
| First Feasible Solution | 1m00s | 22 | 427 |
| Best feasible solution | 1m48s | 37 | 661 |
| **(2 -Step) Corners Optimization** | **Time** | **#Generation** | **#Evaluations** |
| Overall Optimization time | 14m09s | 150 | 3184 |
| First Feasible Solution | 10m06s | 102 | 1664 |
| Best feasible solution | 12m07s | 120 | 1937 |

* In a single processor Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz PC running Linux.

The resolution for this problem was achieved using an iterative process very similar to the traditional analog design. In a first attempt to solve the problem, it was observed that one of the corners in particular was very difficult to satisfy. This corner was identified by the inspection of the run-time information returned from simulation and provided by the tool. This critical corner point (corner nº3) is pointed in Fig. 6.19. The simulation was interrupted and the static weight for that corner was changed as shown in Fig. 6.20, and the simulation was rerun again. Finally it was possible to obtain several solutions within the original time constraint.

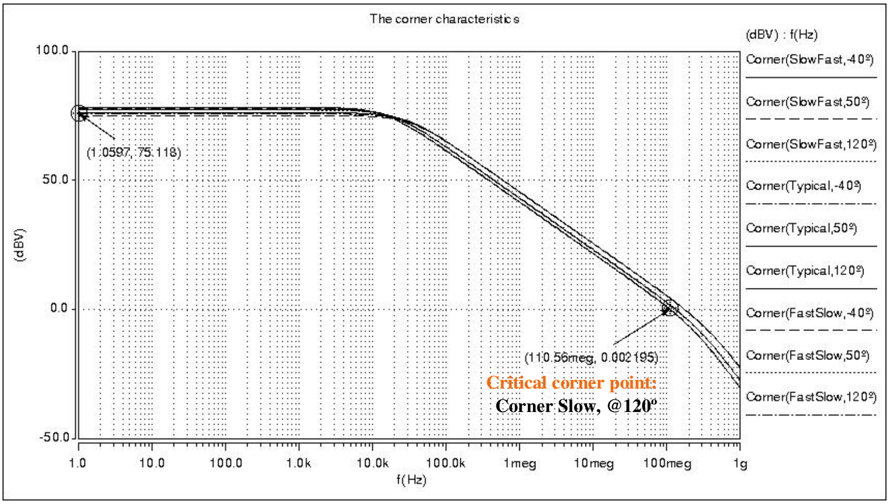


**Fig. 6.19** Gain magnitudes for corner analysis

```
CORNER MODEL    TEMP  WEIGHT  SATISFY   FITNESS(i)  VIOL(i)     SUM_FIT[23104]
[ 1 ]   SF-SLOW  -40º     1     5 / 18=23  1.196e-03  0.000e+00    1.196e-03
[ 2 ]   SF-SLOW  +50º     1     5 / 18=23  2.343e-03  0.000e+00    3.540e-03
[ 3 ]   SF-SLOW  +120º    2     5 / 18=23  3.592e-02  0.000e+00    3.946e-02
[ 4 ]   TT-TYP   -40º     1     5 / 18=23  5.782e-03  0.000e+00    4.524e-02
[ 5 ]   TT-TYP   +50º     1     5 / 18=23  3.923e-03  0.000e+00    4.916e-02
[ 6 ]   TT-TYP   +120º    1     5 / 18=23  2.040e-02  0.000e+00    6.957e-02
[ 7 ]   FS-FAST  -40º     1     5 / 18=23  2.224e-02  0.000e+00    1.418e-01
[ 8 ]   FS-FAST  +50º     1     5 / 18=23  4.035e-02  0.000e+00    1.821e-01
[ 9 ]   FS-FAST  +120º    1     5 / 18=23  5.146e-02  0.000e+00    2.336e-01
****** EUREKA ******
Byebye. AIDA - IC_DESIGN Terminate ...
Job done on a  Intel(R) Core(TM)2 Quad CPU Q6600  @ 2.40GHz
```

**Fig. 6.20** Output from simulation where all corners are satisfied

All parameters from column "*VIOL*" have null values indicating the constraints related to designer's rules (18) were totally satisfied. Additionally, the column "*SATISFY*" confirms that all constraints including the performances (5+18) were satisfied in all corner points. The column "*FITNESS*" represents the cost function values for each corner point. Generally, the sum of the fitness is not zero due to computation reasons. This amount is used to rank feasible solutions satisfying the main goals of the problem, i.e., minimization of power and current consumption.

## *6.5.3 Folded Cascode OpAmp with AB Output*

### 6.5.3.1 Description

Class AB amplifiers are typically used when there is a need to drive resistive or high capacitive loads. They provide a large output current during output voltage transients, while keeping a low current consumption when in quiet state. The architecture shown in Fig. 6.21 and Fig. 6.22 is a two stage topology, with the first stage being a typical folded cascade architecture, followed by a class AB output stage. Capacitor *C1* and resistor *R1* provide the necessary miller compensation with a pole zero solution to increase the phase margin.

Transistors *M6* together with transistors *M22* to *M27* provide the control of the class AB operation by controlling the maximum output current of *M7* and *M8*. This control is performed by keeping control of the *Vgs* voltage of *M6a* and *M8* so that *Vgs6a* + *Vgs8* = *Vgs24* + *Vgs27*. Therefore the maximum output current supplied by *M8* is controlled by the current in *M24* and *M27*. For positive currents the same principle is applied to *M6b*, *M7*, *M23* and *M25*.

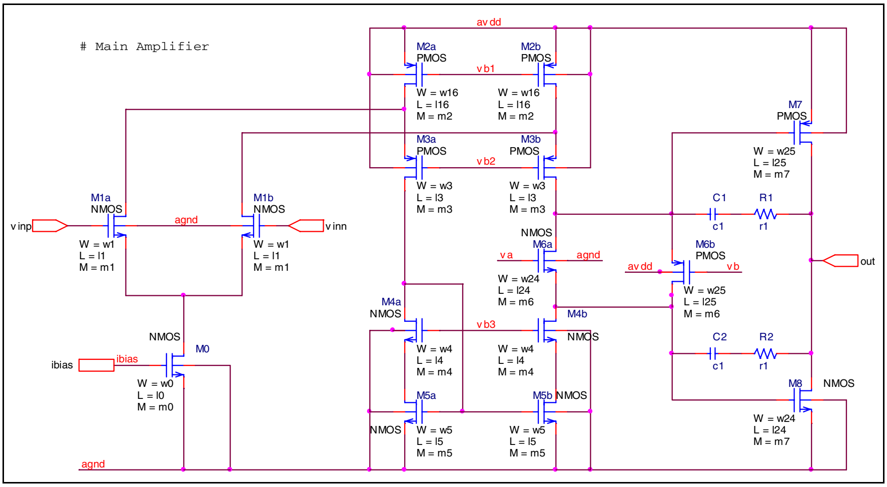


**Fig. 6.21** Main class AB Amplifier

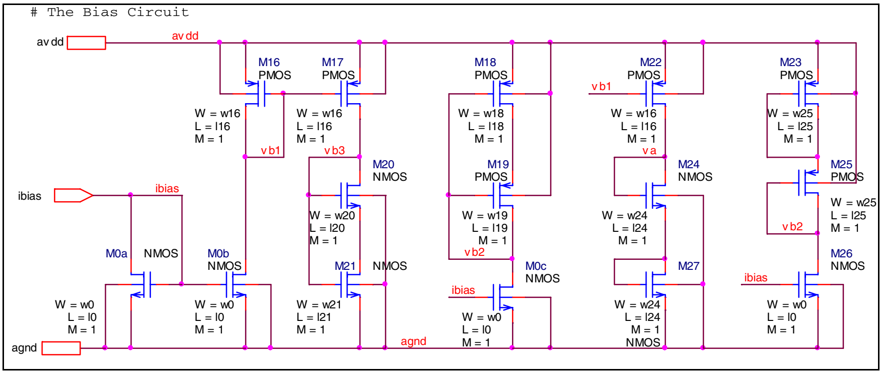


**Fig. 6.22** Bias circuit

### 6.5.3.2 Problem Specifications and Design Configurations

The main objective was to synthesize the presented folded cascode amplifier using the AMS (Austria Mikro Systeme Intl. AG) 0.35 µm, 3.3 V CMOS technology according to the performance specifications listed in Table 6.42, and designed to follow the fundamental designer rules and optimization design constraints of Table 6.43. The specifications must be satisfied for the corner points of Table 6.44. The total of constraints (performance constraints and the constraints derived from designer's rules) results in 33 optimizations constraints that must be satisfied in the optimization process described in Table 6.45.

**Table 6.42** Performance parameter specifications

| | **Specifications** | **Target** | **Units** | **Description** |
|---|---|---|---|---|
| **Electrical** | gain_dc | > 70 | dB | Unit-gain frequency |
| | gbw | > 75 | MHz | Phase margin |
| | phase | [ 60-90] | º | DC gain |
| **Environmental** | CL | 1 | pF | Capacitive Load |
| | Ibiaspar | 10 | µA | Ibias |
| | Wi fixed | 5 | µm | Fixed all widths |
| **Optimization** | Power Consumption | Minimum | mW | Objective |
| | Current Consumption | Minimum | µA | Objective |

**Table 6.43** Matching and constraints details

| **Matching** | | **Constraints** | | | **Design Variable** | | |
|---|---|---|---|---|---|---|---|
| Dependent Variable | Optimization Variable | $V_{GS}$ - $V_T$ [a] [Min - Max] | $V_{DS}$ - $V_{DSAT}$ [b] Min / Max | Unit | Name | Range [Min Max;Step] | Unit |
| M0 | (_wx, _l0, _m0) | [100 - 250] | >50 | mV | _l0 | [0.35; 10; 0.05] | µm |
| M1a = M1b | (_wx, _l1, _m1) | [50 - 250] | >50 | mV | _l1 | [1; 20; 1] | µm |
| M2a = M2b | (_wx, _l16, _m0) | [100 - 250] | [50, 250] | mV | _l3 | [0.35; 10; 0.05] | µm |
| M3a = M3b | (_wx, _l3, _m3) | [100 - 250] | >50 | mV | _l4 | [0.35; 10; 0.05] | µm |
| M4a = M4b | (_wx, _l4, _m4) | [100 - 250] | >50 | mV | _l5 | [1; 20; 1] | µm |
| M5a = M5b | (_wx, _l5, _m5) | [100 - 250] | [50, 250] | mV | _l16 | [0.35; 10; 0.05] | µm |
| M6a | (_wx, _l24, _m6) | [100 - 300] | >50 | mV | _l18 | [0.35; 10; 0.05] | µm |
| M6b | (_wx, _l25, _m6) | [100 - 300] | >50 | mV | _l21 | [0.35; 10; 0.05] | µm |
| M7 | (_wx, _l25, _m7) | [100 - 250] | >50 | mV | _l24 | [0.35; 10; 0.05] | µm |
| M8 | (_wx, _l24, _m7) | [100 - 250] | >50 | mV | _l25 | [0.35; 10; 0.05] | µm |
| R1=R2 | _r1 | | | | _m0 | [1; 100;1] | unit |
| C1=C2 | _c1 | | | | _m1 | [1; 100;1] | unit |
| **Bias Circuit:** | | | | | _m3 | [1; 100;1] | unit |
| Dep.Variable | Opt.Variable | Dep.Variable | Opt.Variable | | _m4 | [1; 100;1] | unit |
| M20 | (_wx, _l4, 1) | M16 =M17 | (_wx, _l16, 1) | | _m5 | [1; 100;1] | unit |
| M21 | (_wx, _l21, 1) | M18 | (_wx, _l18, 1) | | _m6 | [1; 100;1] | unit |
| M23 =M25 | (_wx, _l25, 1) | M19 | (_wx, _l3, 1) | | _m7 | [1; 100;1] | unit |
| M24 =M27 | (_wx, _l24, 1) | M0a=M0b== M0c | (_wx, _l0, 1) | | _r1 | [100;1000;50] | Ω |
| M26 | (_wx, _l0, 1) | | | | _c1 | [1;5;0.05] | tF |

(a) Technology Constraints - overdrive voltages (b) Drain-sources voltages

**Table 6.44** Corners analysis data

| **Conditions** | **Variation points** | | |
|---|---|---|---|
| MOS worst case parameters | Ws-Slow | Tm-Typ | Wp-Fast |
| Temperature Range (° C) | -40° C | +50° C | +120°C |

**Table 6.45** Optimization algorithm parameters

| **Parameter** | **value** | **Parameter** | **value** | **Parameter** | **value** |
|---|---|---|---|---|---|
| **Kernel** | GA-MOD | **Selection** | Tourn. by feas. | **Popsize** | 64 |
| **Strategy** | Corner Optim. | **Crossover** | Two point | **Init Pop** | 2*Popsize |
| **Sampling** | LHS | **Mutation** | Dynamic | **Generations** | 250 |
| **Sort** | Priority to performance fitness then performance constraints. | **Adaptive** | Yes | **Stop** | End of generations |
| | | **Elite** | 25% of pop. | **Search Space domain** | 6.417e+39 |

Note: Evaluation Engine by HSPICE simulator.

### 6.5.3.3 Design Analysis

The attached testbench circuit used for DC and AC simulations is illustrated in Fig. 6.23.

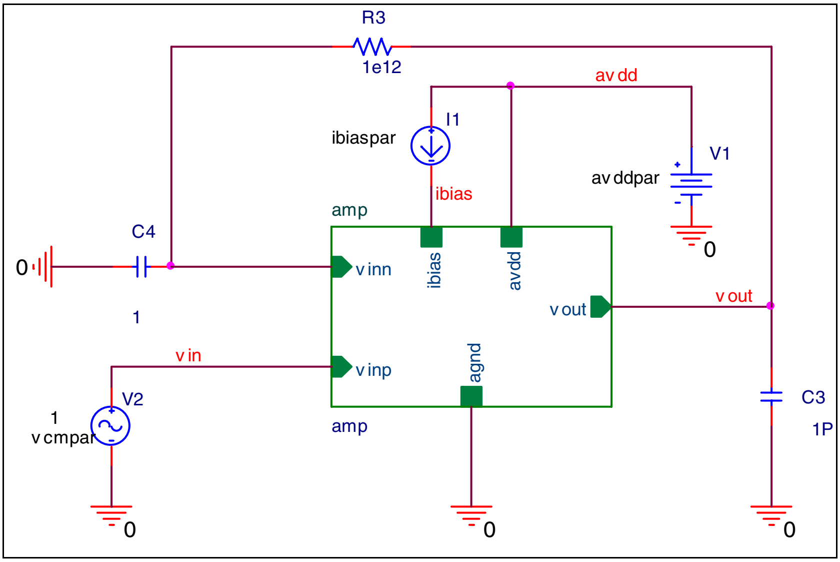


**Fig. 6.23** OpAmp testbench for DC and AC specifications

The simulation results of the main amplifier and bias circuit sizing are shown in Table 6.46. The final transistor dimensions are displayed in Table 6.47, while, Table 6.48 summarizes the runtime information for this one step corner optimization.

Fig. 6.24 gives an outline of the text simulation data produced by the optimization tool of one feasible solution.

All parameters from column “VIOL” have null values indicating the constraints related to designer’s rules (28) were totally satisfied. Additionally, the column “SATISFY” confirms that all constraints (5+28) were satisfied in all corner points. The column “FITNESS” represents the fitness values for each corner point. Generally, the sum of the fitness is not zero due to computation reasons. This amount is used to rank feasible solutions satisfying the goals of the problem.

**Table 6.46 - Performance parameter specifications**

| | Specifications | Target | Sizing Result | Units |
|---|---|---|---|---|
| **Electrical** | DC gain | > 70 | 94.7 | dB |
| | GBW | > 75 | 115.1 | MHz |
| | Phase margin | [ 60-90] | 69.0 | o |
| **Optimization** | Power Consumption | Minimum | 6.1 | mW |
| | Current Consumption | Minimum | 1.8 | mA |

**Table 6.47** Final transistor dimensions

| **Main Amplifier** | W/L (µm/µm) | **Bias** | W/L (µm/µm) |
|---|---|---|---|
| M0 | 280 / 1.05 | M20 | 5 / 1.60 |
| M1a = M1b | 95 / 0.45 | M21 | 5 / 3.55 |
| M2a = M2b | 280 / 0.50 | M23 =M25 | 5 / 0.45 |
| M3a = M3b | 165 / 0.45 | M24 =M27 | 5 / 1.15 |
| M4a = M4b | 220 / 1.60 | M26 | 5 / 1.05 |
| M5a = M5b | 85 / 0.75 | M16 =M17 | 5 / 0.50 |
| M6a | 55 / 1.15 | M18 | 5 / 3.05 |
| M6b | 55 / 0.45 | M19 | 5 / 0.45 |
| M7 | 430 / 0.45 | M0a=M0b=M0c | 5 / 1.05 |
| M8 | 430 / 1.15 | | |

**Table 6.48** Runtime info

| **Design Problem** | | |
|---|---|---|
| Opt. Variables / Constraints (Specs + Design Const.) | 19 | 5 + 28 = 33 |

| **(1 -Step) Corners Optimization** | **Time** | **#Generation** | **#Evaluations** |
|---|---|---|---|
| Overall Optimization time | 23m04s | 250 | 4061 |
| First Feasible Solution | 20m03s | 227 | 3702 |

* In a single processor Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz PC running Linux.

```
CORNER MODEL      TEMP   WEIGHT  SATISFY    FITNESS(i)  VIOL(i)     SUM_FIT[1130]
[ 1 ]  WS-SLOW   -40º    1       5 / 28=33  3.845e-03   0.000e+00   3.845e-03
[ 2 ]  WS-SLOW   +50º    1       5 / 28=33  6.113e-03   0.000e+00   6.113e-03
[ 3 ]  WS-SLOW   +120º   1       5 / 28=33  8.628e-03   0.000e+00   8.628e-03
[ 4 ]  Tm-TYP    -40º    1       5 / 28=33  1.570e-02   0.000e+00   1.570e-02
[ 5 ]  Tm-TYP    +50º    1       5 / 28=33  3.238e-02   0.000e+00   3.238e-02
[ 6 ]  Tm-TYP    +120º   1       5 / 28=33  4.600e-02   0.000e+00   4.600e-02
[ 7 ]  Wp-FAST   -40º    1       5 / 28=33  5.720e-02   0.000e+00   5.720e-02
[ 8 ]  Wp-FAST   +50º    1       5 / 28=33  7.653e-02   0.000e+00   7.653e-02
[ 9 ]  Wp-FAST   +120º   1       5 / 28=33  8.840e-02   0.000e+00   8.840e-02
****** EUREKA ******
Byebye. AIDA - IC_DESIGN Terminate ...
Job done on a  Intel(R) Core(TM)2 Quad CPU Q6600  @ 2.40GHz
```

**Fig. 6.24** Output from simulation where all corners are satisfied

Fig. 6.26 shows the gain magnitude and phase only for typical mean process and 50° C conditions. Fig. 6.26 shows the graphical results for the AC corner analysis. As it can be noticed, this simulation design meets the required specs related to DC gain, gain bandwidth and phase margin satisfying all corner points as reported in Table 6.49. Table 6.50 shows the maximum and minimum of the corner points.

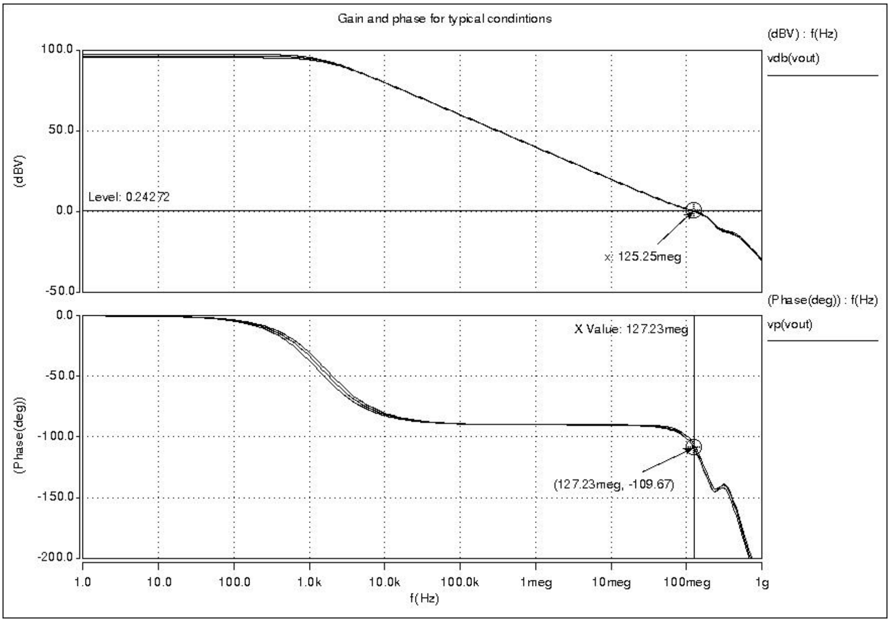


**Fig. 6.25** Gain magnitude and phase for typical conditions

**Table 6.49** Numerical results for corners analysis

| **Corner** | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Process | | Slow | | | Typical | | | Fast | | |
| Temperature | | -40° | 50° | 125° | -40° | 50° | 125° | -40° | 50° | 125° |
| **Specs** | **Values** | | | | | | | | | |
| DC Gain (dB) | > 70 | 102.1 | 98.9 | 97.6 | 96 | 94.6 | 92.7 | 89.1 | 87.3 | 84.4 |
| f (A=0dB) (MHz) | > 75 | 131 | 101.7 | 100.5 | 173.4 | 115.1 | 104.5 | 246 | 158 | 109 |
| Phase (A=0dB) (°) | ------ | -115 | -118 | -119 | -112 | -111 | -112 | -110 | -102 | -101 |
| PM (grade) | [60-90] | 64 | 61 | 60 | 67 | 68 | 67 | 69 | 77 | 76 |

**Table 6.50** Minimum and maximum values for AC corners analysis

| Specs Range | | | | |
|---|---|---|---|---|
| DC Gain (dB) | Min: | 84,4 dB | Max: | 102.1 dB |
| GBW (MHz) | Min: | 100.5 MHz | Max: | 246 MHz |
| PM (grade) | Min: | 60º | Max: | 77º |

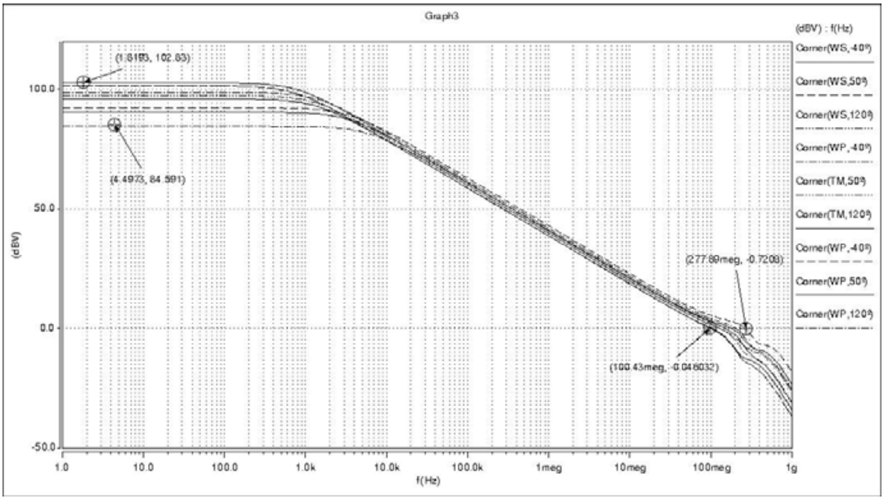


**Fig. 6.26** Gain magnitudes for corners analysis

## 6.6 Comparison with Other Tools/Approaches

The lack of a known open reference tool for IC design automation makes it difficult to the evaluation task of comparing objectively different implementations, although, the analog design automation community is developing efforts to circumvent this situation. Comparing the performance and effectiveness of the final GENOM optimizer with published reference tools is not always possible because the information contained in most of the publications omit some detail of the implementation, maybe imposed by logistics limitations or by author intentionality focusing only the most important piece of interest. Some common ignored items are related with incomplete definition of testbench circuitry, range of optimization variables, used device models and insufficient output data exposed. An exception is made for the first benchmark circuit presented above that was gently provided by Prof. Francisco Fernandez, IMSE-CNM-CSIC/University of Seville which allows the comparison between GENOM and one important reference tool for analog design, the FRIDGE optimizer [8].

### 6.6.1 FRIDGE Benchmark Circuit Tests

The benchmark circuit of reference is a novel single ended folded cascode OpAmp tested with FRIDGE synthesis tool [8], whose results are used to compare the performance and effectiveness of the final GENOM optimizer. This benchmark circuit includes all items necessary to the implementation and test, including the original netlist, testbenchs, device models, performance measures, constraints, range of variables and performance results obtained by the FRIDGE optimization tool. With this data, GENOM is able to test exactly in the same conditions as the FRIDGE tool. The schematic of the circuit is shown in Fig. 6.27 and testbench defined in Fig. 6.28.

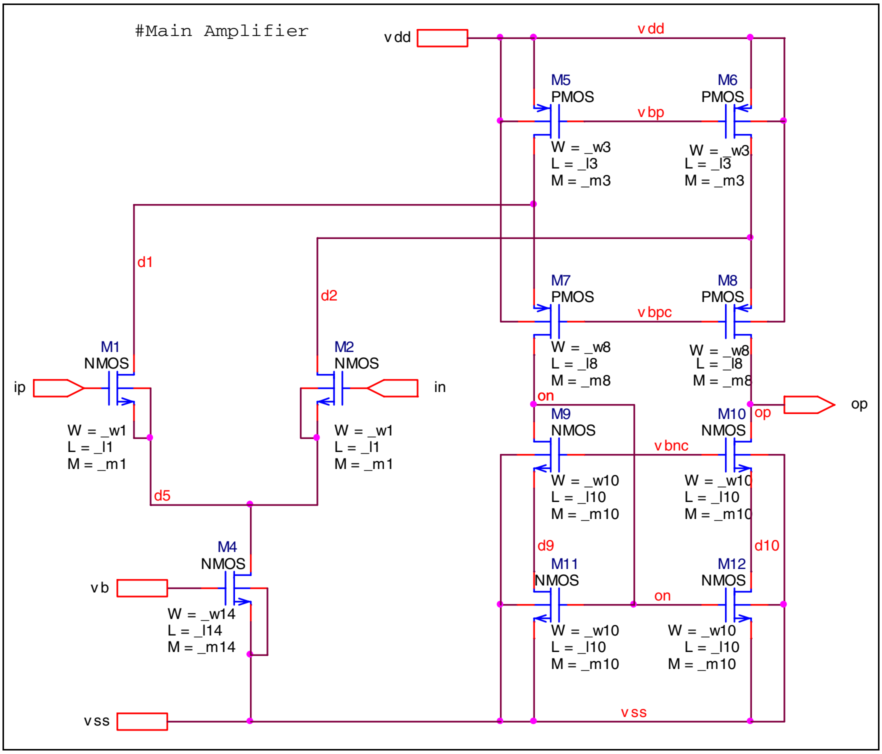


**Fig. 6.27** Main Amplifier

### 6.6.2 Optimization Test with FRIDGE Ampop

Following the original FRIDGE approach, this experiment does not optimize the bias circuit, only the main circuit. The experiments were synthesized with the UMC 0.18um Regular Vt 1.8V Mixed Mode process Spice Model and were

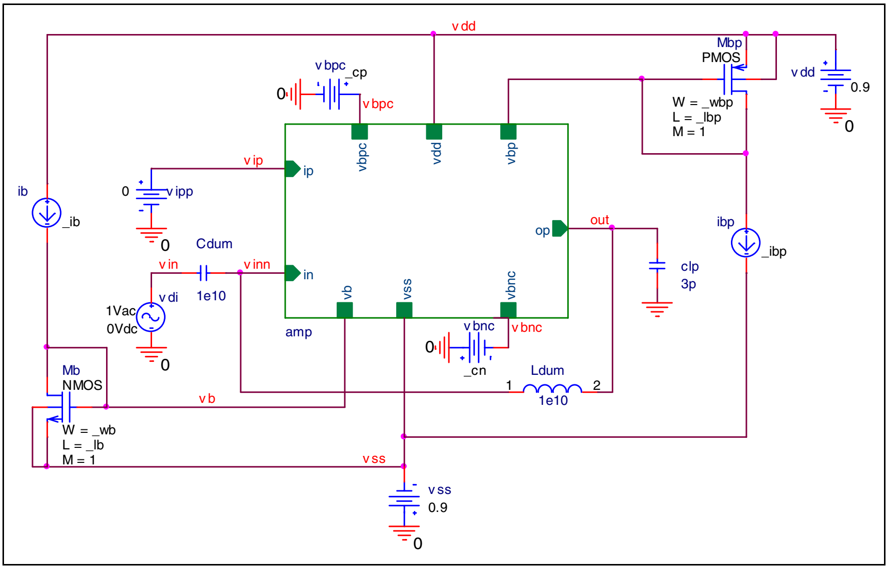


**Fig. 6.28** OpAmp testbench for DC and AC specifications

executed on an AMD X64 2.8 GHz dual core machine and use HSPICE [9] to simulate the circuit and extract performance parameters. The performance constraints and the constraints derived from designer's rules result in 20 optimizations constraints that must be satisfied by the optimization process described in Table 6.51. The design performances and final results achieved with both tools are depicted in Table 6.52. Optimization process uses 15 independent variables whose ranges and respective final transistor dimensions are given in Table 6.53.

**Table 6.51** Optimization algorithm parameters

| **Parameter** | **value** | **Parameter** | **value** |
|---|---|---|---|
| Kernel | GA-MOD | Crossover | Two point |
| Strategy | Typical + Corner Optimization | Mutation | Dynamic |
| Sampling | LHS | Adaptive | No |
| Sort method | Priority to constraints then performance fitness | Elite | 25% of population |
| Selection | Tournament by ”feasibility” | Generations | 150 |
| Popsize | 32 | Search Space | 4.716883e+53 |

**Table 6.52** Design performance and final results

| **Target** | **FRIDGE** | **GENOM** | **GENOM Test specification** |
|---|---|---|---|
| gbw > 1.20e+07 | 1.603e+07 | **1.535e+07** | (gbw > 1.2e+07) |
| gain > 7.00e+01 | 7.000e+01 | **7.061e+01** | + (gain > 70.0) |
| pm > 5.50e+01 | 8.064e+01 | **7.960e+01** | + (verify_bound(pm,55,90)) |
| sr > 1.00e+07 | 1.533e+07 | **1.536e+07** | + (sr > 1.0e+7) |
| dm2 > 1.20e+00 | 9.785e+00 | **9.245e+00** | + (check_bound(dm2, 1.2,1000)) |
| dm4 > 1.20e+00 | 5.200e+00 | **1.568e+00** | + (check_bound(dm4, 1.2,1000)) |
| dm5 > 1.20e+00 | 2.214e+00 | **1.836e+00** | + (check_bound(dm5, 1.2,1000)) |
| dm7 > 1.20e+00 | 1.055e+01 | **8.171e+00** | + (check_bound(dm7, 1.2,1000)) |
| dm9 > 1.20e+00 | 3.055e+00 | **2.807e+00** | + (check_bound(dm9, 1.2,1000)) |
| dm11 > 1.20e+00 | 1.9594+00 | **1.653e+00** | + (check_bound(dm11,1.2,1000)) |
| onm2 > 1.00e-01 | 1.004e-01 | **1.098e-01** | + (check_bound(onm2, 0.100,1000)) |
| onm4 > 3.00e-02 | 3.023e-02 | **3.240e-01** | + (check_bound(onm4, 0.030,1000)) |
| onm5 > 3.00e-02 | 5.662e-02 | **9.866e-02** | + (check_bound(onm5, 0.030,1000)) |
| onm7 > 3.00e-02 | 4.255e-02 | **8.761e-02** | + (check_bound(onm7, 0.030,1000)) |
| onm9 > 3.00e-02 | 4.919e-02 | **3.802e-02** | + (check_bound(onm9, 0.030,1000)) |
| onm11 > 3.00e-02 | 1.782e-01 | **2.451e-01** | + (check_bound(onm11,0.030,1000)) |
| osp > 5.00e-01 | 6.253e-01 | **5.660e-01** | + (check_bound(osp, 0.5, 1000)) |
| osn < -5.00e-01 | -5.022e-01 | **-5.057e-01** | + (check_bound(osn,1000, -0.5)) |
| **Area** (min) | 2.371e+01 | **1.6873e+01** | + (min( area, 0, 30)) |
| **Power** (min) | 2.333e-04 | **2.446e-04** | + (min( rmspow, 0, 0.001)) |
| Cost value | -0.292589 | **8.0704e-02** | --- |
| Iter 1st/ (last) solution | ---- / 2497 | **1110/ (2464)** | --- |
| Time (s) 1st/(last) sol. | n.a. | **25.08/(53.68)** | --- |

The main performance spec *gbw* stands for gainbandwidth, *gain* means the dc gain, *pm* is the phase margin, *sr* is the slew rate and the optimization goal is to minimize both the area (*Area*) and power dissipation (*power*). Both, the optimization goals and constraints used in the experiments were defined by the original benchmark circuit. The electrical constraints, as defined by the original benchmark circuit, are illustrated in HPSICE style in expression (6.3):

```
.meas ac dm2 param ='abs(lx3(x1.m1)/lv10(x1.m1))'
.meas ac onm2 param ='vgs(x1.m1) - vth(x1.m1)'
.meas ac osp param ='($cp + abs(vth(x1.m8)))'
.meas ac osn param ='($cn - abs(vth(x1.m10)))'
```

( 6.3)

**Table 6.53** Ranges and Final Transistor Dimensions

| **Optimization Var.** | **FRIDGE** | **GENOM** |
|---|---|---|
| $cn = [-0.4,0]; | $cn = -8.755479e-02 | _cn = -4.490000e-02 |
| $cp = [0.0,0.4]; | $cp = 6.247103e-02 | _cp = 1.000000e-03 |
| $l1 = [0.18u,5u]; | $l1 = 1.560000e-06 | _l1 = 1.380000e-06 |
| $l4 = [0.18u,5u]; | $l4 = 4.700000e-07 | _l4 = 1.940000e-06 |
| $l5 = [0.18u,5u]; | $l5 = 3.800000e-07 | _l5 = 3.700000e-07 |
| $l7 = [0.18u,5u]; | $l7 = 7.600000e-07 | _l7 = 9.100000e-07 |
| $l9 = [0.18u,5u]; | $l9 = 2.060000e-06 | _l9 = 8.900000e-07 |
| $l11 = [0.18u,5u]; | $l11 = 6.000000e-07 | _l11 = 2.190000e-06 |
| $ib = log[30u,400u]; | $ib = 4.842000e-05 | _ib = 4.851000e-05 |
| $w1 = log[0.24u,200u]; | $w1 = 1.951000e-05 | _w1 = 1.491000e-05 |
| $w4 = log[0.24u,200u]; | $w4 = 3.034000e-05 | _w4 = 6.990000e-06 |
| $w5 = log[0.24u,200u]; | $w5 = 7.131000e-05 | _w5 = 3.678000e-05 |
| $w7 = log[0.24u,200u]; | $w7 = 1.045300e-04 | _w7 = 6.304000e-05 |
| $w9 = log[0.24u,200u]; | $w9 = 6.562000e-05 | _w9 = 3.145000e-05 |
| $w11 = log[0.24u,200u]; | $w11 = 3.080000e-06 | _w11 = 7.320000e-06 |

## 6.6.3 Comparison Results

Table 6.54 shows the GENOM and FRIDGE performance side by side and also depicts the GENOM run-time information in several optimizations points. In order to achieve a computing independent comparison between the tools, the following analysis is based, exclusively, on the number of evaluations "*nEval*" and the main goals, related to the minimization of power and area. Anyway, the time information was not provided with the actual benchmark circuit. GENOM achieved the first solution in 25s approx. using 1110 evaluations and reached a similar performance to FRIDGE in 1461 evaluations, corresponding to an efficiency increase of 41%. One of the best solutions improves simultaneously the power in 17% and 15% in the area as described in Table 6.54 with 2064 evaluations. The GENOM optimization was able to produce 183 new feasible solutions. Fig. 6.29 shows the gain magnitude and phase for typical mean process and 50° C conditions.

**Table 6.54** GENOM benchmarks

| **Target** | **nEval** | **Power (min)** | **Area (min)** | **Time (s)** |
|---|---|---|---|---|
| **FRIDGE**<br>Final results | 2497 | 2.333e-04 | 2.371e+01 | --------- |
| **GENOM** | | | | |
| 1st Feasible Solution | 1110 | 4.0590e-04 | 2.9727e+01 | 25.08 |
| GENOM similar to FRIDGE | 1461 | 2.284e-04 | 2.377e+01 | 32.47 |
| GENOM better than FRIDGE | 2064 | 1.918e-04 | 2.009e+01 | 43.33 |
| Final Results | 2464 | 2.446e-04 | 1.6873e+01 | 53.68 |

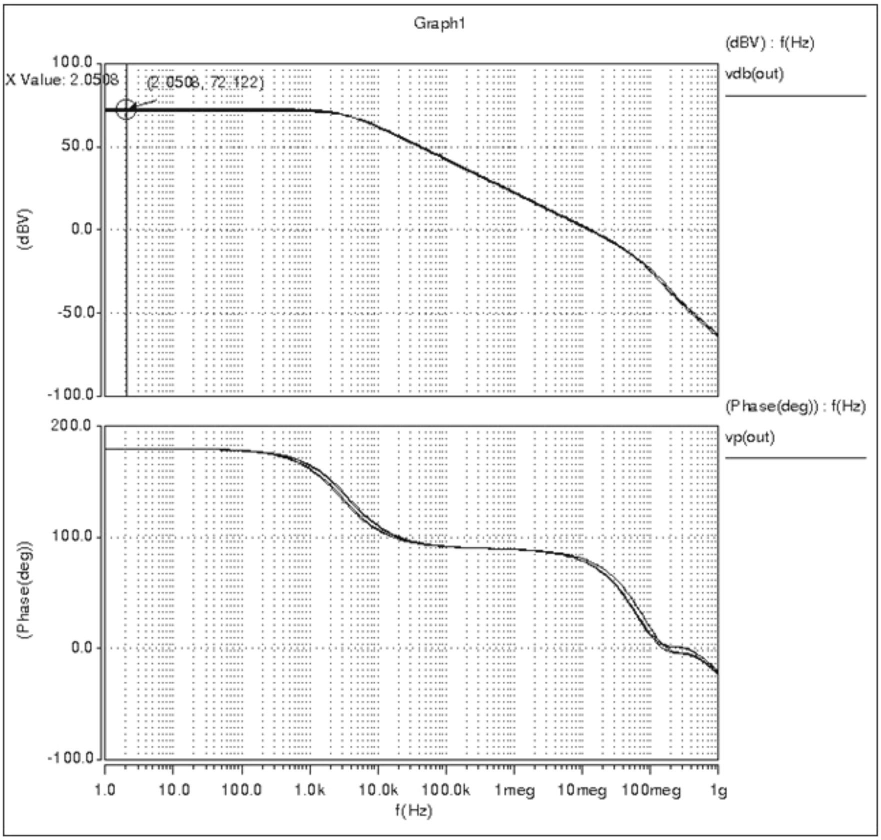


**Fig. 6.29** Gain magnitude and phase for typical conditions

### 6.6.4 Corners Optimization with FRIDGE Circuit

Although there is no available benchmark information about the corner optimization for the FRIDGE benchmark circuit, the next experiment tests the GENOM performance for this type of optimization. However, there was the need to relax one specification, maintaining the others intact, in order to allow the corner optimization. This situation may occur when the performance specification is defined with a value that will not meet the worst-case corner point. The identification of this problematic specification was relatively easy to detect. First, it was verified that after several runs, the final solution always fulfils all constraints except one in a particular corner point. After identifying the problematic constraint, a new optimization was executed, assigning a high weight to this corner. However, the final solution did not improve, so this is probably the case where a specification was defined with a value that is not able to satisfy all corner points at the same time.

Expression (6.4) reflects the small modification introduced to the original FRIDGE specs.

Before:

$$(a0 > 70.0\ \text{dB}) \quad \rightarrow \ \text{New value: } (a0 > 67.0\ \text{dB}) \qquad (6.4)$$

The specifications must be satisfied for the corner points of Table 6.55. Table 6.56 shows the GENOM performance and depicts the run-time information for the first

**Table 6.55** Corners analysis data

| **Conditions** | **Variation points** | | |
|---|---|---|---|
| MOS worst case parameters | SNFP | TT | FNSP |
| Temperature Range (° C) | -40° C | +50° C | +120°C |

**Table 6.56** Design performance and final results for corners analysis

| **Target** | **GENOM Results** | **Optimization Var.** | **GENOM Results** |
|---|---|---|---|
| Gb > 1.20e+07 | gb = 1.845000e+07 | $cn = [-0.4,0] | _cn = -1.971000e-01 |
| a0 > 6.70e+01 | a0 = 6.871930e+01 | $cp = [0.0,0.4] | _cp = 6.300000e-03 |
| pm > 5.50e+01 | pm = 7.435350e+01 | $l1 = [0.18u,5u] | _l1 = 2.110000e-06 |
| sr > 1.00e+07 | sr = 2.103000e+07 | $l4 = [0.18u,5u] | _l4 = 1.270000e-06 |
| dm2 > 1.20e+00 | dm2 = 8.352600e+00 | $l5 = [0.18u,5u] | _l5 = 4.100000e-07 |
| dm4 > 1.20e+00 | dm4 = 2.588000e+00 | $l7 = [0.18u,5u] | _l7 = 8.100000e-07 |
| dm5 > 1.20e+00 | dm5 = 1.803000e+00 | $l9 = [0.18u,5u] | _l9 = 1.150000e-06 |
| dm7 > 1.20e+00 | dm7 = 1.008620e+01 | $l11 = [0.18u,5u] | _l11 = 2.420000e-06 |
| dm9 > 1.20e+00 | dm9 = 2.828500e+00 | $ib = log[30u,400u] | _ib = 6.644000e-05 |
| dm11 > 1.20e+00 | dm11 = 1.695600e+00 | $w1 = log[0.24u,200u] | _w1 = 2.496000e-05 |
| onm2 > 1.00e-01 | onm2 = 1.311000e-01 | $w4 = log[0.24u,200u] | _w4 = 1.935000e-05 |
| onm4 > 3.00e-02 | onm4 = 1.695000e-01 | $w5 = log[0.24u,200u] | _w5 = 4.813000e-05 |
| onm5 > 3.00e-02 | onm5 = 1.129000e-01 | $w7 = log[0.24u,200u] | _w7 = 1.022000e-04 |
| onm7 > 3.00e-02 | onm7 = 6.762000e-02 | $w9 = log[0.24u,200u] | _w9 = 4.983000e-05 |
| onm9 > 3.00e-02 | onm9 = 4.708000e-02 | $w11 = log[0.24u,200u] | _w11 = 3.123000e-05 |
| onm11 > 3.00e-02 | onm11 = 1.362000e-01 | | |
| osp > 5.00e-01 | osp = 5.752000e-01 | | |
| osn < -5.00e-01 | osn = -6.185000e-01 | | |
| Areas (min) | 2.450920e+01 | | |
| Power (min) | 3.286000e-04 | | |
| Cost | 1.145283e-01 | | |
| Iteration | 20281 | | |
| Time (s) | 411.18 | | |

and final solution. This optimization produces 135 generations and executes 20281 electrical evaluations and creates 165 new solutions satisfying all design specs and functional constraints in all corners points.

Where, SNFP, TT and FNSP mean the Slow/Fast, Typical/Typical and Fast/Slow process, respectively.

Table 6.57 presents the final results for the present optimization problem.

**Table 6.57** GENOM corner optimization

| Performance Constr. | nEval | Power (min) | Area (min) | Time (s) |
|---|---|---|---|---|
| 1st Solution in GENOM | 9193 | 3.68E-004 | 3.31E+001 | 186.98 |
| Final evaluation | 20281 | 3.29E-004 | 2.45E+001 | 411.18 |

## 6.7 Conclusions

This chapter presented a set of experiments which test the GENOM’s performance to design high-performance and novel circuit topologies. The above simulations have shown that the circuits designed by the GENOM tool conform to the synthesis objectives with efficiency and accuracy. Particularly, GENOM was able to achieve an efficiency increase of about 40% and a significant increase in performance when compared with one of the synthesis tool of reference.

The use of corners analysis and embedded designer rules methodology in every optimization run increases the value and trust in the final product, although the inclusion of corners analysis in the optimization scheme slows down the execution times considerably. This option produces a more robust design to parameter and process variations and in a certain way avoids the undesired circuits with high sensibility which causes big variations at the output in response to a small deviation in one of the parameters.

The great majority of the presented results are based on a 0.35μm CMOS technology because of the good availability of these models, although the GENOM tool has also been tested with success for a 0.18μm technology models in the telescopic and the FRIDGE OpAmp case studies. Since the technological process is independent from the optimization algorithm, virtually any technological process, including the more recent ones, can be supported by this tool.

With a proper configuration, the present optimization tool is able to synthesize a broad range of analog ICs beyond the class of circuits presented in this research.

## References

[1] Wu, C.: Analog integrated circuits – lecture notes. IEE 6703, National Chiao Tung University (2006), http://www.cc.nctu.edu.tw (Accessed March 2009)

[2] Baker, R.J.: CMOS, circuit design, layout and simulation, 2nd edn. IEEE Press, John Wiley & Sons, Inc., (2005)

[3] Martin, K., Johns, D.: Analog integrated circuit design. John Wiley & Sons Inc., Chichester (1996)

[4] Milenova, B.L., Yarmus, J.S., Campos, M.M.: SVM in oracle database 10g: Removing the barriers to widespread adoption of support vector machines. In: Proc. 31st International Conference on Very Large Data Bases, pp. 1152–1163 (2005)

[5] Boardman, M., Trappenberg, T.: A heuristic for free parameter optimization with support vector machines. In: Proc. International Joint Conference on Neural Networks, pp. 610–617 (2006)

[6] Imbault, F., Lebart, K.: A stochastic optimization approach for parameter tuning of support vector machines. In: Proc. 17th International Conference on Pattern Recognition, vol. 4, pp. 597–600 (2004)

[7] Chang, C., Lin, C.: LIBSVM: A library for support vector machines (2001), `http://www.csie.ntu.edu.tw/~cjlin/libsvm` (Accessed March 2009)

[8] Medeiro, F., et al.: A Statistical optimization-based approach for automated sizing of analog cells. In: Proc. ACM/IEEE Int. Conf. Computer-Aided Design, pp. 594–597 (1994)

[9] Horta, N.C.: Analogue and mixed-signal systems topologies exploration using symbolic methods. In: Proc. Analog Integrated Circuits and Signal Processing, vol. 31(2), pp. 161–176 (2002)