

SmartSpice Circuit Design Using Local and Global Optimization

Introduction

The SmartSpice optimizer capability performs variable and parameter optimization of circuits. In the context of circuit design, we are interested in achieving specified target goals to some constraints on a set of variables or circuit parameters, typically a lower and upper bound on the design parameters. From a set of initial values and target goals, the general purpose-optimizing engine iterates over the parameter space until the goals have been reached. Circuits can be optimized using any of the available analyses, including steady state, frequency and time-domain.

There are two types of optimization: performance measure optimization and function optimization.

In performance measure optimization, variables and device/model parameters are altered in order to ensure that some electrical characteristics meet specifications. Some electrical characteristic examples include: delay, rise/fall times, trip-point, maximum/minimum current, gain, S-parameters or any additional circuit performance measure that can be calculated by a SmartSpice `.MEASURE` statement.

In function optimization, variables and parameters are altered such that some defined function of the circuit meets specifications. For example, matching calculated curves with some target responses from DC, AC and transient analyses.

Local and Global Methods

Optimization algorithms can be considered as either a local optimization method or a global method. Local optimization methods require a good search direction to make improvements and are generally robust and fast when near a local minimum. Global optimization methods are typically slower since they must avoid becoming trapped at local minima – and may use a stochastic search in order to locate a global minimum.

Common to all optimization methods, is a starting point, representing a reasonable estimate of the solution. This starting set of parameters may be obtained from previous experience, knowledge of the problem or a reasonable assumption given the extent of the design space. A local optimizer will iteratively generate a sequence of points, then terminate when a solution satisfies the given design criteria or when no further progress is possible.

Global optimizers attempt to find the minimum objection function among all local minima. Hence, this is a more complex problem than local optimization, as the method must look beyond a local minimum in searching for a global minimum. This may involve looking in the design space further away from the starting point compared to local optimizers.

There are now a number of optimization algorithms available in SmartSpice. For local optimization, the popular Levenberg-Marquardt method performs well when starting near a solution. The global optimization methods include Hooke-Jeeves, simulated annealing, parallel tempering, genetic algorithm, differential evolution and a hybrid method. Details of these various methods can be found in the SmartSpice manual.

Setup

The syntax to use global optimization is the same as the existing `.MODIF` optimization section. To select one of the various optimization strategies, set the `OPTIMIZER` option as follows:

```
.OPTION OPTIMIZER=val
```

where `val` is the global optimizer type, specified by any of the following:

- **LM** – Levenberg-Marquardt
- **HJ** – Hooke-Jeeves method
- **SA** – Simulated annealing
- **PT** – Parallel tempering
- **GA** – Genetic algorithm
- **DE** – Differential evolution
- **H** – Hybrid optimizer

The default is LM, which corresponds to the local optimizer Levenberg-Marquardt.

Preferences for the optimization algorithms can be set with the Preferences Dialog found under **Edit** → **Preferences...** → **Tools** → **Optimizer**. With this dialog, various optimization related settings can be set including stopping criteria, tolerance, iterations, number of simulation and other optimizer-specific parameters. For example, Fig. 1 shows the optimization settings for the simulated annealing algorithm.

Design #1 – Bandpass Filter

In this example, we are interested in designing a simple bandpass filter having a 0.5 dB equal-ripple response in the passband. The center frequency is 1 GHz, the band-

width is 10%, and the impedance is 50 Ω . The Smart-Spice netlist is shown in Fig. 2. There are 4 optimization variables, L_{13} , L_2 , C_{13} , and C_2 . The starting values are [90 nH, 0.5 nH, 0.1 pF, 25 pF]. The target goals were for $|S_{21}| = -0.5\text{dB}$ in the passband, and -55 dB (500 MHz to 600 MHz) and -60 dB (1.4 GHz to 1.5 GHz) in the lower and upper passband, respectively. After performing Levenberg-Marquardt optimization, the final element values are [150.5 nH, 0.908 nH, 0.167 pF, 27.7 pF]. Fig. 3 shows the results of $|S_{21}|$ before and after optimization.

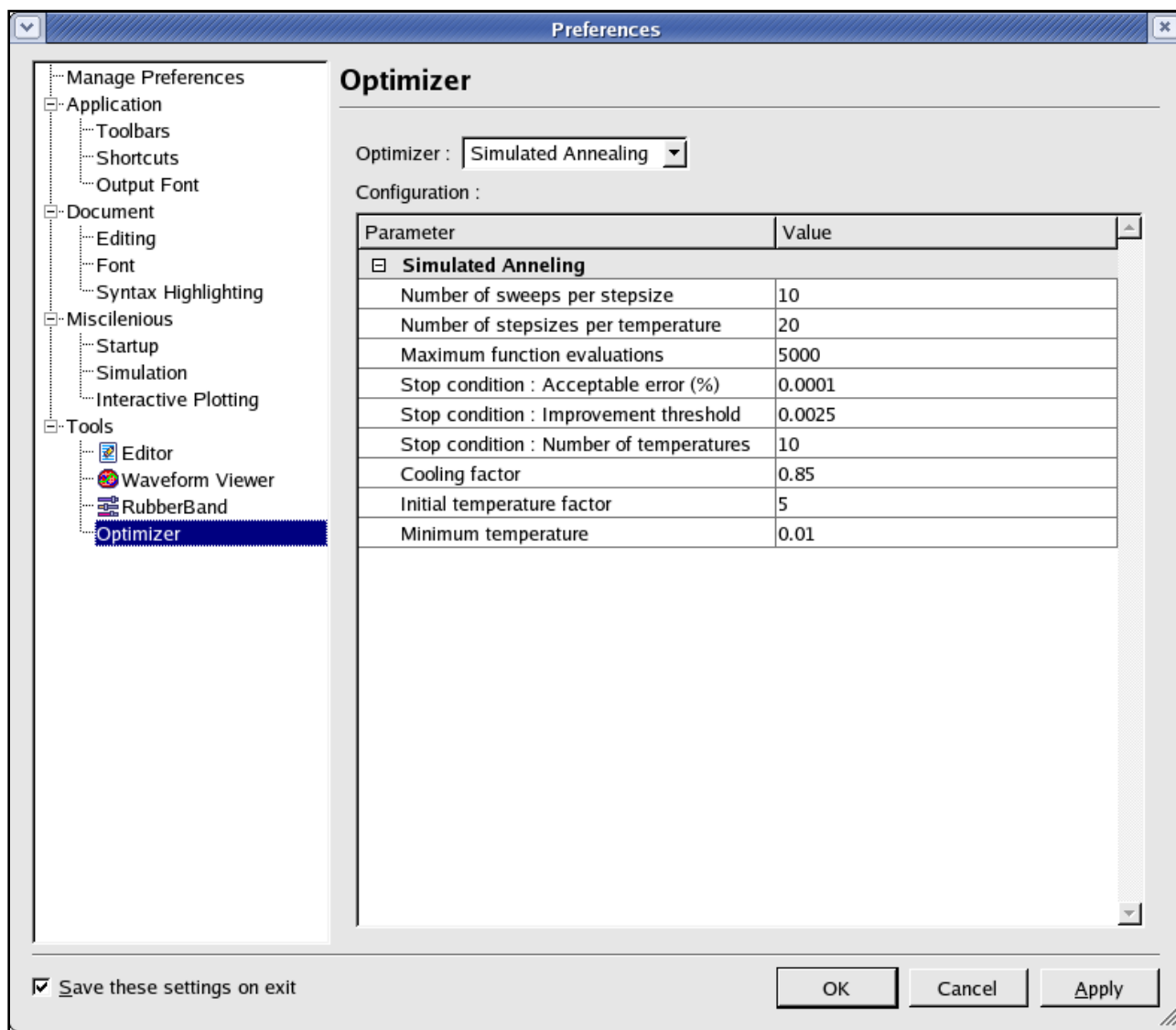


Fig. 1. Optimization preferences dialog.

```

* BAND PASS FILTER with z0=(50,50)
Vin IN 0 ac 1 dc 0
Vout OUT 0 ac 1 dc 0
L1 IN 1 L_13
C1 1 2 C_13
L2 2 0 L_2
C2 2 0 C_2
L3 2 3 L_13
C3 3 OUT C_13

.NET N=2 IN OUT 0 LIN 1001 500Meg 1.5G

.PARAM l_13=154.3nH l_2=0.666nH c_13=0.135pF c_2=39.33pF

.MEASURE NET passband MIN db(sm(out,in)) FROM=950Meg TO=1.05G
.MEASURE NET lower MAX db(sm(out,in)) FROM=500Meg TO=600Meg
.MEASURE NET upper MAX db(sm(out,in)) FROM=1400Meg TO=1500Meg

.MODIF
+ OPTIMIZE L_13=opt(10nH 200nH 90nH)
+ L_2 =opt(0.01nH 10nH 0.5nH)
+ C_13=opt(0.01pF 20pF 0.1pF)
+ C_2 =opt(5pF 300pF 25pF)
+ TARGETS passband=-0.5 lower=-55 upper=-60
+ OPTIONS AVG=1e-8 MEASOFF=1 NUMFUNC=300 NUMITER=300
+ IPLOFF=2 MARQUP=8000

.OPTIONS NOMOD iplot_one
.OPTION OPTIMIZER=LM
.IPLOT db(sm(out,in))
.END
    
```

Fig. 2. SmartSpice netlist of a bandpass filter.

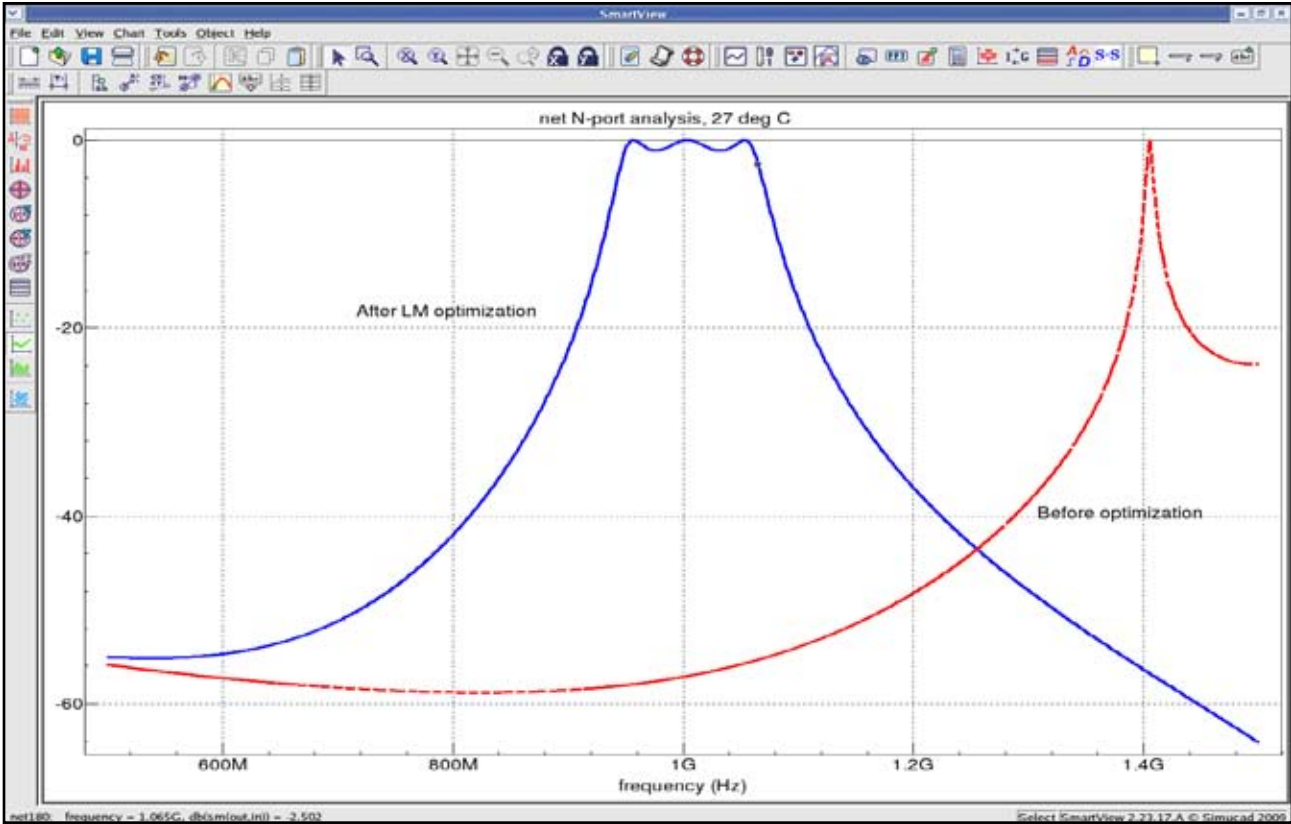


Fig. 3. |S21| results before and after Levenberg-Marquardt optimization.

Design #2 –Wideband Amplifier Gain

In this example, we are interested in optimizing the gain of a wideband amplifier. This circuit uses 120 design parameters – including all of the resistor elements and many of the BJT transistor parameters. We are interested in a final design in which the minimum gain is 119 and the maximum gain is 120, from 1 Hz to 10 MHz. The local optimizer, Levenberg-Marquardt, is unable to arrive at a satisfactory solution. Thus, we employ one of the global optimizers, in particular, simulated annealing, by selecting the option as,

```
.OPTION OPTIMIZER=SA
```

The SmartSpice netlist is shown in Fig. 4. The results before and after performing Simulated Annealing optimization are shown in Fig. 5. The gain is flat at 122 and then drops to 112 at 10 MHz. If we choose, instead, the hybrid optimization strategy, using Simulated Annealing followed by the Levenberg-Marquardt method, the final results at low frequency and 10 MHz are 123 and 114, respectively. With the hybrid optimizer, the second optimizer, Levenberg-Marquardt, adds additional local optimization around the solution found by the first strategy, in this case, Simulated Annealing.

```
EXAMPLE4: RCA3040 WIDEBAND AMPLIFIER

VIN  1  0  DC  0  SIN(0  0.1  50MEG  0.5NS)  AC  1
VEE  3  0  -15
VCC  2  0  15
*
RS2  31  0  1K
RS1  30  1  1K
*
Q1  2  30  5  QNLA
Q2  2  31  6  QNLA
*
Q3  10  5  7  QNLB
Q4  11  6  7  QNLB
*
Q5  14  12  10  QNLC
Q6  15  12  11  QNLC
*
Q7  12  12  13  QNLD
Q8  13  13  0  QNLD
*
Q9  7  8  9  QNL
*
Q10  2  15  16  QNLE
Q11  2  14  17  QNLE
*
R1  5  3  ra
R2  6  3  rb
R3  9  3  rc
R4  8  3  rd
R5  8  0  re
R6  2  14  rf
R7  2  12  rg
R8  2  15  rh
R9  16  0  ri
R10  17  0  rj
*
.MODEL  QNL  NPN(BF=80      RB=100  CCS=2PF  TF=0.3NS  AF=1.0
```

continued...

```

+          KF=5.4E-16 TR=6NS  CJE=3PF  CJC=2PF  VA = 50 )

.MODEL  QNLA  NPN(BF=80      RB=100    CCS=2PF  TF=0.3NS  AF=1.0
+          NF=1.0      BR=2.38  NR=1.0    VJE=1.44  MJE=0.33
+          VJC=0.99    MJC=0.33  VJS=1.39  MJS=0.5   IS=92.5e-16
+          KF=5.4e-16  TR=6NS   CJE=3PF   CJC=2PF   VA=50)

.MODEL  QNLB  NPN(BF=80      RB=100    CCS=2PF  TF=0.3NS  AF=1.0
+          NF=1.0      BR=2.38  NR=1.0    VJE=1.44  MJE=0.33
+          VJC=0.99    MJC=0.33  VJS=1.39  MJS=0.5   IS=92.5e-16
+          KF=5.4e-16  TR=6NS   CJE=3PF   CJC=2PF   VA=50)

.MODEL  QNLC  NPN(BF=80      RB=100    CCS=2PF  TF=0.3NS  AF=1.0
+          NF=1.0      BR=2.38  NR=1.0    VJE=1.44  MJE=0.33
+          VJC=0.99    MJC=0.33  VJS=1.39  MJS=0.5   IS=92.5e-16
+          KF=5.4e-16  TR=6NS   CJE=3PF   CJC=2PF   VA=50)

.MODEL  QNLD  NPN(BF=80      RB=100    CCS=2PF  TF=0.3NS  AF=1.0
+          NF=1.0      BR=2.38  NR=1.0    VJE=1.44  MJE=0.33
+          VJC=0.99    MJC=0.33  VJS=1.39  MJS=0.5   IS=92.5e-16
+          KF=5.4e-16  TR=6NS   CJE=3PF   CJC=2PF   VA=50)

.MODEL  QNLE  NPN(BF=80      RB=100    CCS=2PF  TF=0.3NS  AF=1.0
+          NF=1.0      BR=2.38  NR=1.0    VJE=1.44  MJE=0.33
+          VJC=0.99    MJC=0.33  VJS=1.39  MJS=0.5   IS=92.5e-16
+          KF=5.4e-16  TR=6NS   CJE=3PF   CJC=2PF   VA=50)
*
**** Parameter labels
.PARAM  ra=4.8K  rb=4.8K  rc=811  rd=2.17K  re=820
+       rf=1.32K  rg=4.5K  rh=1.32K  ri=5.25K  rj=5.25K
*
**** MEASURE
.MEASURE AC gain_min MIN vm(16) FROM=1 TO=10e6
.MEASURE AC gain_max MAX vm(16) FROM=1 TO=10e6

.AC  DEC  5 1 10GHZ

.OPTIONS NOMOD iplot_one

**** Optimization specification
.MODIF
+   OPTIMIZE  ra=opt(1K  10K  2K)
+           rb=opt(1K  10K  2K)
+           rc=opt(100  2K  200)
+           rd=opt(500  5K  1K)
+           re=opt(100  2K  500)
+           rf=opt(500  5K  1K)
+           rg=opt(1K  10K  2K)
+           rh=opt(500  5K  12K)
+           ri=opt(1K  10K  2K)
+           rj=opt(1K  10K  2K)

```

continued...

```

+           QNL(BF)=OPT( 10  200  80 )
+           QNL(RB)=OPT( 10  500  100 )
+           QNL(CCS)=OPT( 0.5pF  10pF  2pF )
+           QNL(TF)=OPT( 0.1NS  1NS  0.3NS )
+           QNL(AF)=OPT( 0.2  5.0  1.0 )
+           QNL(KF)=OPT( 1e-18  1e-14  5.4e-16 )
+           QNL(TR)=OPT( 2ns  15ns  6ns )
+           QNL(CJE)=OPT( 0.5pF  10pF  3pF )
+           QNL(CJC)=OPT( 0.5pF  10pF  2pF )
+           QNL(VA)=OPT( 10  100  50 )

+           QNLA(BF)=OPT( 10  200  80 )
+           QNLA(RB)=OPT( 10  500  100 )
+           QNLA(CCS)=OPT( 0.5pF  10pF  2pF )
+           QNLA(TF)=OPT( 0.1NS  1NS  0.3NS )
+           QNLA(AF)=OPT( 0.2  5.0  1.0 )
+           QNLA(NF)=OPT( 0.1  3  1.0 )
+           QNLA(BR)=OPT( 0.2  3  2.38 )
+           QNLA(NR)=OPT( 0.5  3  1.0 )
+           QNLA(VJE)=OPT( 0.1  1.5  1.44 )
+           QNLA(MJE)=OPT( 0.1  1.5  0.33 )
+           QNLA(VJC)=OPT( 0.25  2  0.99 )
+           QNLA(MJC)=OPT( 0.1  1  0.33 )
+           QNLA(VJS)=OPT( 0.2  1.5  1.39 )
+           QNLA(MJS)=OPT( 0.1  2  0.5 )
+           QNLA(IS)=OPT( 1e-18  1e-14  92.51e-16 )
+           QNLA(KF)=OPT( 1e-18  1e-14  5.4e-16 )
+           QNLA(TR)=OPT( 2ns  15ns  6NS )
+           QNLA(CJE)=OPT( 0.5pF  10pF  3PF )
+           QNLA(CJC)=OPT( 0.5pF  10pF  2PF )
+           QNLA(VA)=OPT( 10  100  50 )

+           QNLB(BF)=OPT( 10  200  80 )
+           QNLB(RB)=OPT( 10  500  100 )
+           QNLB(CCS)=OPT( 0.5pF  10pF  2pF )
+           QNLB(TF)=OPT( 0.1NS  1NS  0.3NS )
+           QNLB(AF)=OPT( 0.2  5.0  1.0 )
+           QNLB(NF)=OPT( 0.1  3  1.0 )
+           QNLB(BR)=OPT( 0.2  3  2.38 )
+           QNLB(NR)=OPT( 0.5  3  1.0 )
+           QNLB(VJE)=OPT( 0.1  1.5  1.44 )
+           QNLB(MJE)=OPT( 0.1  1.5  0.33 )
+           QNLB(VJC)=OPT( 0.25  2  0.99 )
+           QNLB(MJC)=OPT( 0.1  1  0.33 )
+           QNLB(VJS)=OPT( 0.2  1.5  1.39 )
+           QNLB(MJS)=OPT( 0.1  2  0.5 )
+           QNLB(IS)=OPT( 1e-18  1e-14  92.51e-16 )
+           QNLB(KF)=OPT( 1e-18  1e-14  5.4e-16 )
+           QNLB(TR)=OPT( 2ns  15ns  6NS )
+           QNLB(CJE)=OPT( 0.5pF  10pF  3PF )
+           QNLB(CJC)=OPT( 0.5pF  10pF  2PF )

```

continued...

```

+           QNLB(VA)=OPT( 10 100 50 )

+           QNLC(BF)=OPT( 10 200 80 )
+           QNLC(RB)=OPT( 10 500 100 )
+           QNLC(CCS)=OPT( 0.5pF 10pF 2pF )
+           QNLC(TF)=OPT( 0.1NS 1NS 0.3NS )
+           QNLC(AF)=OPT( 0.2 5.0 1.0 )
+           QNLC(NF)=OPT( 0.1 3 1.0 )
+           QNLC(BR)=OPT( 0.2 3 2.38 )
+           QNLC(NR)=OPT( 0.5 3 1.0 )
+           QNLC(VJE)=OPT( 0.1 1.5 1.44 )
+           QNLC(MJE)=OPT( 0.1 1.5 0.33 )
+           QNLC(VJC)=OPT( 0.25 2 0.99 )
+           QNLC(MJC)=OPT( 0.1 1 0.33 )
+           QNLC(VJS)=OPT( 0.2 1.5 1.39 )
+           QNLC(MJS)=OPT( 0.1 2 0.5 )
+           QNLC(IS)=OPT( 1e-18 1e-14 92.51e-16 )
+           QNLC(KF)=OPT( 1e-18 1e-14 5.4e-16 )
+           QNLC(TR)=OPT( 2ns 15ns 6NS )
+           QNLC(CJE)=OPT( 0.5pF 10pF 3PF )
+           QNLC(CJC)=OPT( 0.5pF 10pF 2PF )
+           QNLC(VA)=OPT( 10 100 50 )

+           QNLD(BF)=OPT( 10 200 80 )
+           QNLD(RB)=OPT( 10 500 100 )
+           QNLD(CCS)=OPT( 0.5pF 10pF 2pF )
+           QNLD(TF)=OPT( 0.1NS 1NS 0.3NS )
+           QNLD(AF)=OPT( 0.2 5.0 1.0 )
+           QNLD(NF)=OPT( 0.1 3 1.0 )
+           QNLD(BR)=OPT( 0.2 3 2.38 )
+           QNLD(NR)=OPT( 0.5 3 1.0 )
+           QNLD(VJE)=OPT( 0.1 1.5 1.44 )
+           QNLD(MJE)=OPT( 0.1 1.5 0.33 )
+           QNLD(VJC)=OPT( 0.25 2 0.99 )
+           QNLD(MJC)=OPT( 0.1 1 0.33 )
+           QNLD(VJS)=OPT( 0.2 1.5 1.39 )
+           QNLD(MJS)=OPT( 0.1 2 0.5 )
+           QNLD(IS)=OPT( 1e-18 1e-14 92.51e-16 )
+           QNLD(KF)=OPT( 1e-18 1e-14 5.4e-16 )
+           QNLD(TR)=OPT( 2ns 15ns 6NS )
+           QNLD(CJE)=OPT( 0.5pF 10pF 3PF )
+           QNLD(CJC)=OPT( 0.5pF 10pF 2PF )
+           QNLD(VA)=OPT( 10 100 50 )

+           QNLE(BF)=OPT( 10 200 80 )
+           QNLE(RB)=OPT( 10 500 100 )
+           QNLE(CCS)=OPT( 0.5pF 10pF 2pF )
+           QNLE(TF)=OPT( 0.1NS 1NS 0.3NS )
+           QNLE(AF)=OPT( 0.2 5.0 1.0 )
+           QNLE(NF)=OPT( 0.1 3 1.0 )
+           QNLE(BR)=OPT( 0.2 3 2.38 )

```

continued...

```

+           QNLE(NR)=OPT( 0.5  3  1.0 )
+           QNLE(VJE)=OPT( 0.1  1.5  1.44 )
+           QNLE(MJE)=OPT( 0.1  1.5  0.33 )
+           QNLE(VJC)=OPT( 0.25  2  0.99 )
+           QNLE(MJC)=OPT( 0.1  1  0.33 )
+           QNLE(VJS)=OPT( 0.2  1.5  1.39 )
+           QNLE(MJS)=OPT( 0.1  2  0.5 )
+           QNLE(IS)=OPT( 1e-18  1e-14  92.51e-16 )
+           QNLE(KF)=OPT( 1e-18  1e-14  5.4e-16 )
+           QNLE(TR)=OPT( 2ns  15ns  6NS )
+           QNLE(CJE)=OPT( 0.5pF  10pF  3PF )
+           QNLE(CJC)=OPT( 0.5pF  10pF  2PF )
+           QNLE(VA)=OPT( 10  100  50 )

+   TARGETS  gain_min=119 gain_max=121
+   OPTIONS  AVG=1e-6 NUMFUNC=500 NUMITER=300
+           MEASOFF=1 IPLOFF=1 MARQUP=5000
+
+   ***** Select the optimizer
+   .OPTION OPTIMIZER=SA
+   .IPLOT VM(16)
+   .END

```

Fig. 4. SmartSpice netlist of the wideband amplifier with 120 design variables.

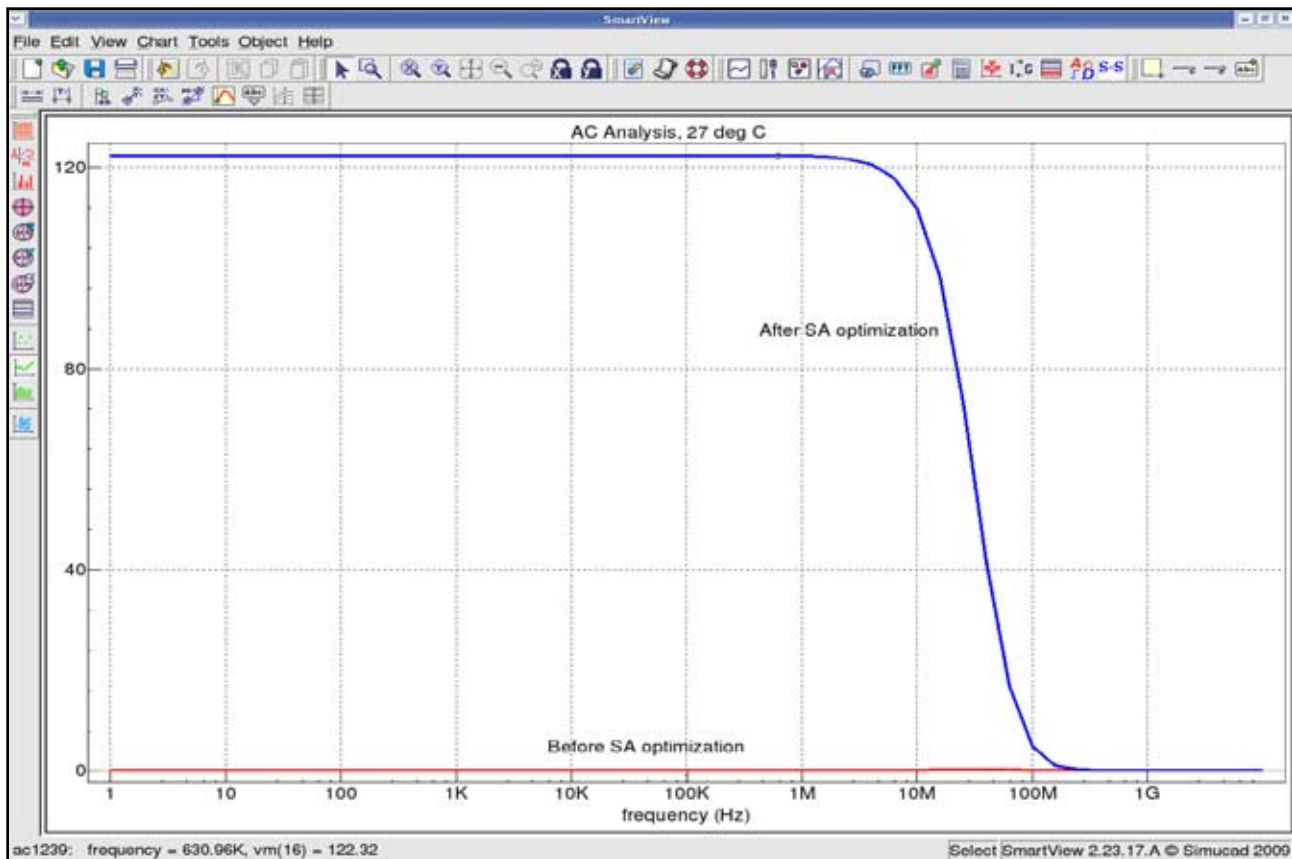


Fig. 5. Plot of amplifier gain before and after Simulated Annealing optimization.

Design #3 –Digital Inverter

In this example, we are interested in minimizing the supply power of a digital inverter, shown in Fig. 6, the same time as meeting the constraint of having an output voltage of 1.95 V.

For this example, the design variable is the power supply voltage, and the target goals are a small value for the calculated power supplied by the DC source and an output voltage of 1.95 V. The netlist is shown in Fig. 7. From a supply voltage (V_{SUPPLY}) starting point of 2.5V, the Levenberg-Marquardt optimizer results in a local minimum.

So, instead, we use the hybrid optimizer, obtained by selecting the option as,

.OPTION OPTIMIZER=H

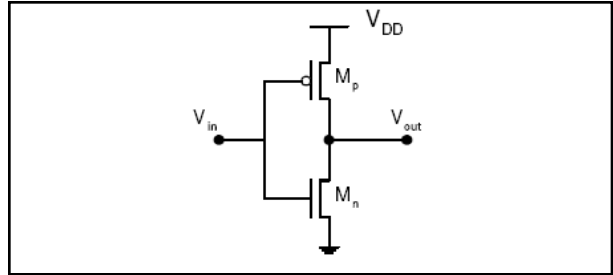


Fig. 6. Digital inverter.

With Parallel Tempering and Levenberg-Marquardt selected as the first and second optimizer, respectively, in the optimizer section of the Preferences dialog, a satisfactory solution is found which meets our target goals. The transient results showing the output voltage before and after optimization are shown in Fig. 8.

```

* Digital inverter

***** SOME PARAMETERS
.PARAM INP_FREQ    = 850MegHz
.PARAM INP_PERIOD  = 1/INP_FREQ
.PARAM NO_PERIODS  = 4
.PARAM TMEAS_START = '(NO_PERIODS-1)*INP_PERIOD'
.PARAM TMEAS_STOP  = NO_PERIODS*INP_PERIOD
.PARAM T1          = 'TMEAS_STOP - 1*INP_PERIOD/4'
.PARAM T2          = 'TMEAS_STOP - 0.75*INP_PERIOD/4'

***** OPTIMIZATION PARAMETERS
.PARAM V_SUPPLY    = 2V
.PARAM WP          = 1000u
.PARAM LMIN       = 0.35u

***** SUPPLY VOLTAGES
VDS (VDD 0) DC V_SUPPLY

***** INPUT SIGNAL
VSIG (IN 0) PULSE(V_SUPPLY 0
+          INP_PERIOD/2
+          INP_PERIOD/1000 INP_PERIOD/1000
+          INP_PERIOD/2 INP_PERIOD)

***** OUTPUT STAGE
MP (OUT IN VDD VDD) pm W=WP L=LMIN
MN (OUT IN 0 0) nm W=WP/3 L=LMIN

***** Models
.MODEL pm PMOS ( level=3 tox=.02e-6
+ phi=0.576 gamma=0 vto=0 alpha=0 kappa=0 is=0)
.model nm NMOS ( level=3 tox=.02e-6
+ phi=0.576 gamma=0 vto=0 alpha=0 kappa=0 is=0)

CL (OUT 0) 10pF
    
```

continued...

```

**** Analysis statement
.TRAN INP_PERIOD/250 NO_PERIODS*INP_PERIOD CALLV SAVEV

.OPTIONS iplot_one nomod
.IPLOT v(OUT)

**** Measure statements
.MEASURE TRAN VHIGH MAX v(OUT) FROM=T1 TO=T2
.MEASURE TRAN PSUPPLY MAX '(v(VDD)*i(VDS))' FROM=T1 TO=T2

.MODIF
+ OPTIMIZE V_SUPPLY=opt(1 5 2.5)
+ TARGETS PSUPPLY=1e-6 VHIGH=1.95
* VLOW=0.001
+ OPTIONS AVG=1e-6 MEASOFF=1 NUMFUNC=50 NUMITER=50
+ IPLOFF=2

**** Select the optimizer
.OPTION OPTIMIZER=H

.END

```

Fig. 7. SmartSpice netlist of the digital inverter.

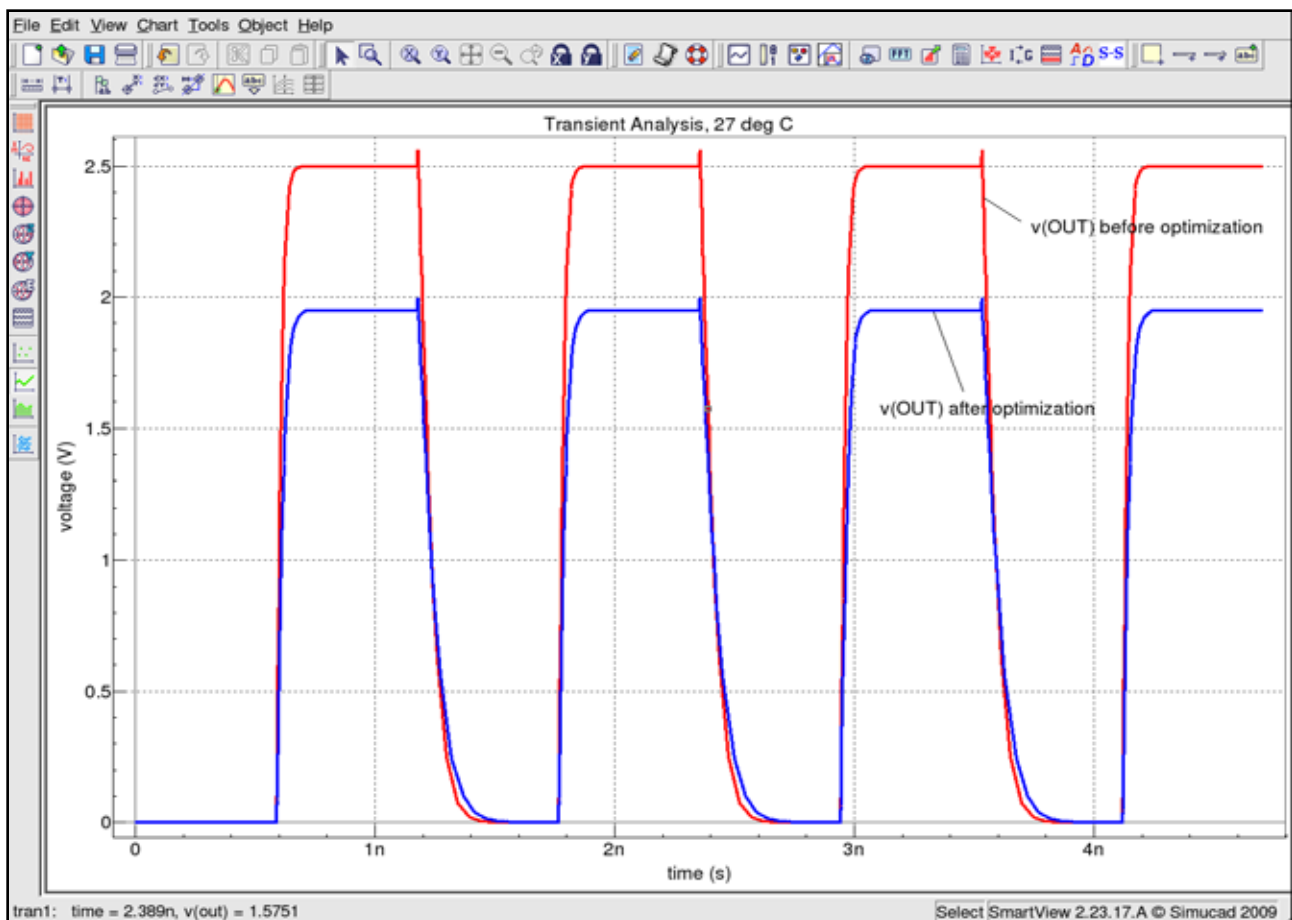


Fig. 8. Plot of the digital inverter output voltage before and after optimization.

Conclusion

The SmartSpice optimization features enable circuit designers to refine their designs in order to meet or improve given specifications. From a set of initial values and specified target goals, the optimizer iterates over the parameter space until the optimal solution is attained. Both local and global optimizers are available. Local optimizers are useful when the starting point is near a local minimum, and generally they are fast. Global opti-

mizers enable a wider parameter search allowing for the possibility of arriving at a global solution. Circuits can be optimized using any of the available analyses, including steady state, frequency, and time-domain. With the addition of global optimization in SmartSpice, designs can now be optimized with large numbers of parameters giving designers more flexibility.