Optimization based Method for Circuit Performance Robustness Analysis and Design Automation

Liuxi Qian
The University of Texas at Dallas
Introduction and background

Optimization based method for

- Circuit performance robustness analysis accounting for parameter variations
  - Frequency response bounds of two stage amplifier
  - Worst case phase noise of VCO
  - Data retention voltage of SRAM cell
- Circuit design and technology migration automation
  - Op Amp design automation
  - LDO design automation
  - Buck converter design automation
  - PLL migration automation

Summary
Parameter variations are major sources to deteriorate the robustness of circuit performance

- Design parameters (e.g. $W, L$) and process parameters (e.g. $V_{th}, t_{ox}$)
- Affected by process variations, temperature changes and environmental noises

Predicted $V_{th}$ variation per year [ITRS 2007].
(CD: chemical decomposition)

Shrinking feature sizes introduce larger parameter variations
- Larger variations with scaling technology nodes
  - Feature size shrinks $\rightarrow$ the parameter variations $\uparrow$

Diagram:
- **Fixed Value**
- **Variation Range**
- **Unknown Varied Range**

**Problem Statement**

- **Design, Process Parameters**
- **Parameter Variations**
- **Circuit Performance**
In the presence of parameter variations, circuit performance deviates from its nominal values.

How to **efficiently and accurately** find the upper and lower bounds?

Each sampling point in parameter space corresponds to a point in the concerned performance space.
Monte Carlo Method

- Generates sampling points according to the distribution of each parameter.
- Performs simulation for each sampling point.
- Requires a large number of samplings for reliable results → time-consuming

Corner case based Method

- Several process corners (TT, SS, FF, SF, FS) are evaluated.
- Underestimates the results, since the worst case may not happen at corners.

Other methods:

- Principle component analysis, importance sampling, quasi Monte Carlo method, and etc.
- Will eventually meet the dilemma among analysis accuracy, sampling number and computation cost.
Generally, the behavior of a circuit in the presence of the uncertain parameters can be modeled as:

\[ y = f(x, p) \]

\[ p = \{ p | p_i \in [p_i^-, p_i^+], i = 1, 2, ..., n \} \]

- \( p \): n-dimensional parameter space
- \( x \): a vector variable in either time or frequency domain
- \( y \): the concerned performance function

A nonlinear optimization problem with constraints of the given varied parameter space:

\[ y^- = \min f \quad s.t. \quad p \]
\[ y^+ = \max f \quad s.t. \quad p \]

Whether \( y^+ \) or \( y^- \) is computed, it is determined by the concerned performance.
The circuit performance analysis with respect to parameter variations can be formulated as a non-linear programming (NLP) problem

\[ y^+ = \max y, \quad s.t. \quad p \]
\[ y^- = \min y, \quad s.t. \quad p \]
\[ p = \{ p_i \in p \mid p_i = [p_i^-, p_i^+], \quad i = 1, 2, \ldots, n \} \]

A multi-start global optimization framework is developed to solve the formulated NLP problem.
Features of Our Method

- **SPICE- C/C++ co-simulation platform**

**Input**
- Circuit netlist, Variational parameters

**Circuit performance evaluation**
- SPICE

**Evaluate y**
- Candidate p

**Nonlinear global optimization**
- C/C++

**Output**
- Worst circuit performance, The caused parameter set

**Features**
- **Transistor level analysis** rather than approximated models
- Applicable to any circuit performance and technology
- In both **time and frequency** domain
- Efficient, achieves **multiple orders magnitude of speedup** over many popular methods
  - “region hit” scheme
  - guided searches in parameter space
Features of Our Method

- Two features make it outperform other methods
  - “Region hit” issue vs. “Point hit” issue
  - Guided search vs. random and independent search

The probability for hitting a region is much larger than hitting a point!

None of 200 Monte Carlo sample points exactly hits the global optimum.

Once a start point hits the region containing the global optimum, the global optimum can be found easily by a local optimization search.
Framework of Multi-start Global Optimization

- **Global Phase:**
  - Multiple start points are generated which are \textit{uniformly distributed} in parameter space.
  - Quasi random sequence is used, such as Halton sequence, Sobol sequence and etc.

- **Local Phase:**
  - From each start point, a local nonlinear programming (NLP) solver is invoked.
  - \textbf{Gradient based NLP solver} is used, such as Conjugate method, L-BFGS method and etc.
Comparison with Other Optimization Methods on Test Functions

Eason’s function

Rastrigin’s function

Six-hump camel back’s function

Genetic, simulated annealing and particle swarm methods are using MATLAB build-in functions. The results are based on an average of 10 trials for each method.

Study Case 1: Frequency Response Bounds Estimation

- **Performance concerned**: the magnitude and phase responses.

- **Objective function**: based on transfer function derived from small signal models
  
  \[
  \min/ \max \quad \text{abs}(H(s, p)) \quad s.t. \quad p \\
  \min/ \max \quad \text{arg}(H(s, p)) \quad s.t. \quad p
  \]

- **Parameter space**: circuit parameters \((g_m, g_{ds}, g_{mb}, C_{gs}, C_{ds}, \text{and etc.})\)

- Optimization based approach (blue lines) effectively handles the dependency between coefficients and obtained bounds

- Bounds, red lines, are obtained by Monte Carlo method using circuit parameters
Phase noise is one of the important characteristics indicating the frequency stability of oscillators.

- Ideal Oscillator: An impulse at the oscillation frequency.
- Noisy Oscillator: Phase noise spread to vicinity frequencies when affected by noise.
- Noisy Oscillator: Phase noise affected by process variations.

Study Case 2: Worst Case Oscillator Phase Noise Evaluation
Study Case 2: Worst Case Oscillator Phase Noise Evaluation

- Performance concerned: phase noise of VCO

Experiment Settings:

- Parameter Space: variations in 14 process parameters are considered. (thickness of oxide layer ($t_{ox}$), threshold voltage ($V_{th}$), channel length ($L$), channel width ($W$) and several capacitances ($C_j, C_{jsw}, C_{go}, C_{jswg}$) for both NMOS and PMOS)
- Both flicker and thermal noises in MOSFET are taken into account.

0.18$\mu$m CMOS technology is used.

**Input**

| Circuit netlist | Variational process parameters |

**Oscillator Phase Noise Evaluation**

Evaluate $y$

Candidate $p$

**Multi-start Global Optimization**

C/C++

**Output**

Worst phase noise
By Comparison purpose, “reference” (not the real worst case) is set by 50000 Monte Carlo samplings.

For hitting the ground truth: averagely, 29 simulations from a start point are taken. When more start points are evaluated, the results converge to the “global optimum”.

Speedup: >1700x
• For hitting the ground truth: > **1700x** faster than MC, > **500x faster** than QMC and > **30x** faster than SA

• Faster convergence rate for finding the “global optimum”.

Phase noise is evaluated at 600KHz through HSPICERF
The worst case phase noise is evaluated at different frequencies.

More process variations lead to more deviations in phase noise. As many as 20dBc/Hz deviations are observed when $6\sigma$ parameter space are considered.
All oscillators are in 0.18µm technology, and phase noise at 600Khz is measured.

Average simulation runs and speedups are recorded.

Compared with Monte Carlo method, at least 140x speedup can be achieved.
Study Case 2: SRAM DRV Evaluation

- Stability of SRAM cell in standby mode is affected by:
  - $V_{dd}$ scaling → for reducing static leakage power consumption
  - Process variation, e.g. $V_{th}$ mismatch
  - Temperature

- **Data Retention Voltage (DRV):** lowest supply voltage to maintain the correct data at the presence of variations

Assume Q=1, QB=0; M5 and M6 is off.
Static Noise Margin (SNM) Degradation due to reduced $V_{dd}$

The maximum side length (DC noise)

Decreased $V_{dd} \rightarrow$ shrunked SNM

Decreased $V_{dd} \rightarrow \Delta V$ between $V_{QB}$ and $V_Q$ decreases as well
SRAM SNM Degradation

- SNM Degradation due to $V_{th}$ variation

At a fixed $V_{dd}$
$V_{th}$ variation $\rightarrow$ asymmetrical curve $\rightarrow$ shrinked SNM
The smallest $V_{dd}$ causes an unacceptable failure rate is considered as DRV.

DRV is a rare event

To guarantee a 99.9% yield for a 1G bit SRAM array, $10^{-12}$ failure rate for a single SRAM cell is estimated!

Hundreds of thousands of samplings are required!!
Performance Concerned: **the DRV of SRAM cell**

Objective function: At each $V_{dd}$, the lower bound $y_{QB}$ for DRV is computed through HSPICE. A few sampling points are adopted on $y_{QB}$ and $S_{QB}$ indicates the sum of these points.

$$\min S_{QB} = \sum_{i=1}^{n} y_{QB(i)}$$

Parameter space: $V_{th}$ for each transistor
Comparison with Monte Carlo method and Importance Sampling method

- A $10^3 \times$ and $10^5 \times$ speedup is achieved compared with IS and MC in 6d space (considering threshold voltage variation of 6 transistors)
- A $10^7 \times$ speedup in 12d space (considering threshold voltage and channel length variation of 6 transistors)
SRAM DRV Evaluation

- DRV at different variation range of $V_{th}$ (3σ - 6σ) for technology from 130nm to 45nm.

- DRV at different temperature (0, 25º, 70º, 100º) for technology from 130nm to 45nm.

The lower technology node suffers more from variation of $V_{th}$.

The temperature has the same impact for different technologies.
The complexity of circuit design and strict time-to-market impose the use of CAD tools for circuit design automation.

20% analog may demand **80%** of total design time.
Optimization based Method for Circuit Design Automation

Behavioral Models using verilog-AMS, systemC, etc.

Mixed-Signal Co-simulation

Transistor level using SPICE

Unsized fixed topology

PDK

Design parameters and ranges

Design specs

Circuit performance

Design Decisions

PVT variations analysis

Post layout parasitics

Start point 1 + local search 1

Start point 2 + local search 2

Start point n + local search n

Parallel Evaluation

Circuit Simulator

Optimization Engine
A multi-start global optimization framework is developed to solve the formulated NLP problem.

A non-linear programming (NLP) problem:

maximize/minimize \( F = f(p) \)
subject to \( p^- \leq p \leq p^+ \)
\( C_j(p) \leq 0, \)

where \( p = \{p_i \in [p_i^-, p_i^+] \}, i = 1, 2, ..., n \)

- Unsized Circuit
- Design parameters and variational ranges
- Process design kit

Generate a start point in the parameter space \( p \)
From each start point, apply a local search solver
Arrive at a local optimum
Converge?
Return the best result obtained so far as the global optimum
### Manual Design

<table>
<thead>
<tr>
<th></th>
<th>TT, 27°C</th>
<th>FF, -40°C</th>
<th>SS, 125°C</th>
<th>σ / Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>GBW (MHz)</td>
<td>≥ 0.92</td>
<td>1.17</td>
<td>0.7</td>
<td>≤ 25.8%</td>
</tr>
<tr>
<td>PM (Degree)</td>
<td>≥ 52.5</td>
<td>51.8</td>
<td>55.5</td>
<td>≤ 3.7%</td>
</tr>
<tr>
<td>GM (dB)</td>
<td>≥ 19.5</td>
<td>21.2</td>
<td>18.5</td>
<td>≤ 6.95%</td>
</tr>
<tr>
<td>SR+ (V/μs)</td>
<td>≥ 0.18</td>
<td>0.26</td>
<td>0.14</td>
<td>≤ 31.6%</td>
</tr>
<tr>
<td>SR- (V/μs)</td>
<td>≥ 0.20</td>
<td>0.26</td>
<td>0.11</td>
<td>≤ 39.7%</td>
</tr>
<tr>
<td>1% Ts+ (μs)</td>
<td>≤ 5.17</td>
<td>4.07</td>
<td>6.78</td>
<td>≤ 25.5%</td>
</tr>
<tr>
<td>1% Ts- (μs)</td>
<td>≤ 5.71</td>
<td>3.80</td>
<td>9.02</td>
<td>≤ 42.7%</td>
</tr>
<tr>
<td>Min IQ (μA)</td>
<td>≤ 69.2</td>
<td>72.1</td>
<td>71.7</td>
<td>≤ 2.2%</td>
</tr>
</tbody>
</table>

- **Performance Concerned:** minimize current consumption
- **Parameter Space:** device dimensions
- **Constraints:** design specifications

---

**Case Study: OPAMP Design Automation**

- **Performance Concerned:** minimize current consumption
- **Parameter Space:** device dimensions
- **Constraints:** design specifications
### Performance improvements at nominal condition

<table>
<thead>
<tr>
<th></th>
<th>Manual design / Automatic design (Bolded are better performance)</th>
<th>Improvement @TT, 27°C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TT @27°C</td>
<td>FF @-40°C</td>
</tr>
<tr>
<td><strong>GBW (MHz)</strong></td>
<td>0.92 / 1.07</td>
<td>1.17 / 1.15</td>
</tr>
<tr>
<td><strong>PM (Degree)</strong></td>
<td>52.5 / 61.7</td>
<td>51.8 / 64.4</td>
</tr>
<tr>
<td><strong>GM (dB)</strong></td>
<td>19.5 / 22.8</td>
<td>21.2 / 23.9</td>
</tr>
<tr>
<td><strong>SR+(V/μs)</strong></td>
<td>0.18 / 0.24</td>
<td>0.26 / 0.31</td>
</tr>
<tr>
<td><strong>SR- (V/μs)</strong></td>
<td>0.20 / 0.46</td>
<td>0.26 / 0.68</td>
</tr>
<tr>
<td><strong>1% Ts+(μs)</strong></td>
<td>5.17 / 3.65</td>
<td>4.07 / 2.84</td>
</tr>
<tr>
<td><strong>1% Ts- (μs)</strong></td>
<td>5.71 / 2.33</td>
<td>3.80 / 1.41</td>
</tr>
<tr>
<td><strong>I_Q (µA)</strong></td>
<td>69.2 / 60.7</td>
<td>72.1 / 58.3</td>
</tr>
</tbody>
</table>

*Case Study: OPAMP Design Automation*
### Performance improvements considering PVT variations

<table>
<thead>
<tr>
<th></th>
<th>Manual design / Automatic design (Bolded are better performance)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TT @27°C</td>
</tr>
<tr>
<td>GBW (MHz)</td>
<td>0.92 / 1.07</td>
</tr>
<tr>
<td>PM (Degree)</td>
<td>52.5 / 61.7</td>
</tr>
<tr>
<td>GM (dB)</td>
<td>19.5 / 22.8</td>
</tr>
<tr>
<td>SR+(V/µs)</td>
<td>0.18 / 0.24</td>
</tr>
<tr>
<td>SR- (V/µs)</td>
<td>0.20 / 0.46</td>
</tr>
<tr>
<td>1% Ts+(µs)</td>
<td>5.17 / 3.65</td>
</tr>
<tr>
<td>1% Ts- (µs)</td>
<td>5.71 / 2.33</td>
</tr>
<tr>
<td>$I_A$ (µA)</td>
<td>69.2 / 60.7</td>
</tr>
</tbody>
</table>
Case Study: LDO Design Automation

- Performance Concerned: load regulation
- Parameter Space: device dimensions
- Constraints: design specifications

Design Specifications

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specification</th>
<th>Design Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loop gain</td>
<td>&gt;50dB</td>
<td>TT 59.58dB FF 61.63dB SS 56.31dB FS 58.82dB SF 59.68dB</td>
</tr>
<tr>
<td>Phase margin</td>
<td>&gt;45°</td>
<td>TT 45.17 FF 45.19 SS 45.92 FS 45.27 SF 45.67</td>
</tr>
<tr>
<td>Load regulation</td>
<td>&lt;0.02%</td>
<td>TT 0.0095 FF 0.0090 SS 0.01 FS 0.0096 SF 0.0094</td>
</tr>
<tr>
<td>Line regulation</td>
<td>&lt;0.02%</td>
<td>TT 0.0072 FF 0.0075 SS 0.0066 FS 0.007 FS 0.0073</td>
</tr>
<tr>
<td>Psrr</td>
<td>&gt;30dB @10kHz</td>
<td>TT 31.99 FF 33.73 SS 30.12 FS 31.68 SF 32.2</td>
</tr>
<tr>
<td>Overshoot</td>
<td>&lt;50mV</td>
<td>TT 14.6mV FF 17.6mV SS 11.5mV FS 13.5mV SF 15.6mV</td>
</tr>
<tr>
<td>Undershoot</td>
<td>&lt;50mV</td>
<td>TT 18.7mV FF 15.7mV SS 24.3mV FS 20.9mV SF 16.8mV</td>
</tr>
</tbody>
</table>
Case Study: LDO Design Automation

- Performance Concerned: efficiency
- Parameter Space: device dimensions
- Constraints: design specifications

<table>
<thead>
<tr>
<th>Design Specifications</th>
<th>Design Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input voltage</td>
<td>Input voltage</td>
</tr>
<tr>
<td>Output voltage</td>
<td>Output voltage</td>
</tr>
<tr>
<td>Operating frequency</td>
<td>Operating frequency</td>
</tr>
<tr>
<td>Load current</td>
<td>Load current</td>
</tr>
<tr>
<td>Efficiency</td>
<td>Efficiency</td>
</tr>
<tr>
<td>Output ripple</td>
<td>Output ripple</td>
</tr>
<tr>
<td>Output accuracy</td>
<td>Output accuracy</td>
</tr>
<tr>
<td>Input voltage</td>
<td>3V</td>
</tr>
<tr>
<td>Output voltage</td>
<td>1.5V</td>
</tr>
<tr>
<td>Operating frequency</td>
<td>1MHz</td>
</tr>
<tr>
<td>Load current</td>
<td>100mA</td>
</tr>
<tr>
<td>Efficiency</td>
<td>&gt;90%</td>
</tr>
<tr>
<td>Output ripple</td>
<td>&lt;10mV</td>
</tr>
<tr>
<td>Output accuracy</td>
<td>&lt;3%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Design Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inductor value</td>
<td>11.07uH</td>
</tr>
<tr>
<td>Output capacitor value</td>
<td>3.77uF</td>
</tr>
<tr>
<td>Power transistor size</td>
<td>14.6m(pmos)</td>
</tr>
<tr>
<td>ESR of output capacitor</td>
<td>6.3m(nmos)</td>
</tr>
<tr>
<td>Parasitic resistance of inductor</td>
<td>70mΩ</td>
</tr>
</tbody>
</table>
Case Study: PLL Design Migration

Hierarchical Optimization Flow

- Preliminary Optimization
  - BL-TL co-simulation
  - TL optimization

- Secondary Optimization
  - TL simulation and optimization

Block Diagram:

- PFD
- Charge Pump
- Loop Filter
- VCO
- Frequency Divider

Diagrams:

(a) PFD and Clock
(b) Charge Pump
(c) Loop Filter
(d) VCO
Block optimization

Behavioral level model
- Originally abstracted from source design
- Updated from preliminary optimization result

Transistor level model
- Set as optimization object
- Configured in systematical connection
- Constrained by both global and local design specifications and functional conditions
Systematical optimization

- Complete transistor level simulation
- Start from the preliminary optimization result
- Refine the preliminary result
### Case Study: PLL Design Automation

#### Optimization Method

<table>
<thead>
<tr>
<th>Optimization Method</th>
<th># of iteration</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC</td>
<td>2036</td>
<td>Reference</td>
</tr>
<tr>
<td>PS</td>
<td>1617</td>
<td>1.26x</td>
</tr>
<tr>
<td>DE</td>
<td>1048</td>
<td>1.94x</td>
</tr>
<tr>
<td>GA</td>
<td>1172</td>
<td>1.73x</td>
</tr>
<tr>
<td>MGO</td>
<td>124</td>
<td>16.42x</td>
</tr>
</tbody>
</table>
## Case Study: PLL Design Automation

<table>
<thead>
<tr>
<th>Design Specification</th>
<th>UMC 130nm</th>
<th>UMC 65nm</th>
<th>IBM 65nm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase Noise(dBc/Hz @ 600kHz)</td>
<td>-115.2</td>
<td>-128</td>
<td>-117.2</td>
</tr>
<tr>
<td>Locking time (µs)</td>
<td>4.7</td>
<td>2.6</td>
<td>4.2</td>
</tr>
<tr>
<td>Min/Max VCO frequency (MHz)</td>
<td>5-77</td>
<td>5-83</td>
<td>5-83</td>
</tr>
<tr>
<td>Power consumption(mW@32MHz)</td>
<td>0.142</td>
<td>0.105</td>
<td>0.098</td>
</tr>
<tr>
<td>Phase margin</td>
<td>45</td>
<td>46</td>
<td>45.5</td>
</tr>
<tr>
<td>Area (µm²)</td>
<td>6525</td>
<td>4400</td>
<td>4125</td>
</tr>
</tbody>
</table>

![Bar chart showing comparison between UMC 65nm and IBM 65nm in terms of performance metrics such as phase noise, locking time, and area.](chart.png)
Optimization based method for circuit robustness analysis and design automation

- Realistic, in transistor level
- General, applicable to both linear and nonlinear circuits, in both time and frequency domain
- Efficient, faster than many currently available methods
Thank you!
Design Optimization of a Rail-to-Rail Operational Amplifier

- **Performance concerned:**
  - Input stage: reduce the $g_m$ variation at different process corners.
  - Output stage: improve the overall performance.

- **Objective function:**
  - Input stage: $g_m$ variation of input stage.
  - Output stage: figure of merit: FOM=$U_{GF}*C_L$/Power.

- **Parameter space:** channel width $W$, miller capacitors.

UGF: unity gain frequency
Reduce the $g_m$ variations at different process corners.

$g_m$ variation across different process corners

<table>
<thead>
<tr>
<th>Corners</th>
<th>$\Delta g_m / g_m$ (MANUAL)</th>
<th>$\Delta g_m / g_m$ (OPTIMIZED)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TT</td>
<td>12.8 %</td>
<td>5.9 %</td>
</tr>
<tr>
<td>FF</td>
<td>13.3 %</td>
<td>5.9 %</td>
</tr>
<tr>
<td>SS</td>
<td>11.7 %</td>
<td>6.5 %</td>
</tr>
<tr>
<td>FS</td>
<td>12.7 %</td>
<td>6.1 %</td>
</tr>
<tr>
<td>SF</td>
<td>13.1 %</td>
<td>5.9 %</td>
</tr>
<tr>
<td>ALL</td>
<td>20.2 %</td>
<td>14 %</td>
</tr>
</tbody>
</table>
- Improve overall performance of a rail-to-rail operational amplifier.

![Diagram showing magnitude and phase response](image)

<table>
<thead>
<tr>
<th>Specification</th>
<th>Manual Design</th>
<th>Optimized Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gain &gt; 70 (dB)</td>
<td>78.2</td>
<td>87</td>
</tr>
<tr>
<td>Phase Margin &gt; 60(°)</td>
<td>61.6</td>
<td>60</td>
</tr>
<tr>
<td>Unity-Gain Frequency &gt;25 (MHz)</td>
<td>27</td>
<td>25</td>
</tr>
<tr>
<td>Load Capacitor (pF)</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Power Consumption (mW)</td>
<td>1.10</td>
<td>0.77</td>
</tr>
<tr>
<td>$g_m$ Variation (%)</td>
<td>12.8</td>
<td>5.9</td>
</tr>
<tr>
<td>FOM (MHzpF/mW)</td>
<td>73.6</td>
<td>97.4</td>
</tr>
</tbody>
</table>
An example of developed EDA tool for operational amplifier design optimization.

- User-friendly UI design.
- Single performance and multiple performance optimization.
- Valid for operational amplifier of any topology.
- Design optimization considering satisfying specifics and reducing variations at different process corners at the same time.
Why Non-Monte-Carlo Method?

As the parameter space enlarges, the required Monte Carlo simulation runs increases exponentially.

Necessary an efficient circuit performance estimation method which keeps relatively low computational complexity!

* Data Source: Dr. Ralf Sommer, DATE 2006, COM BTS DAT DF AMF;

Note: $3\sigma$ means that for a 100Mio transistor design 100,000(!!!) transistors may fail!
Parameter variational space conversion

A parameter PDF is truncated at $\pm k\sigma$ values to form an interval parameter.

An $n$ dimensional parameter space is built by $n$ interval parameters.
Local Search iterations:

1. Start with a point
2. Repeat
   • Determine a direction through calculating the gradient
   • Choose a proper step
   • Slide towards to the calculated direction with the proper step
3. Until stopping criterions are satisfied
4. Then the “close” point (local optima) will be returned as a result
Each sampling point in parameter space corresponds to a point in the concerned performance space.
The derived transfer function of the two stage amplifier is:

\[
H(s) = \frac{g_{m1}g_{m5}R_o R_L + s \left[ -R_{o1}R_L \left( C_c g_{m1} + C_{gd5} g_{m1} + C_{gd1} g_{m5} \right) \right] + s^2 \left[ R_{o1}R_L C_{gd1} \left( C_c + C_{gd5} \right) \right]}{1 + s \left[ R_L \left( C_L + C_c + C_{gd5} \right) + R_{o1} \left( C_c + C_{gd1} + C_{gd5} + C_{gs5} + C_{o1} \right) + g_{m5} R_{o1} R_L \left( C_c + C_{gd5} \right) \right]}
\]

**Dependency Problem in Project I**

\[
H(s, a, b) = \frac{x_1(s, a)}{x_2(s, b)} = \frac{\sum_{i=0}^{k} a_i(p) s^i}{\sum_{j=0}^{m} b_j(p) s^j}
\]

- \( a_0 = g_{m1} g_{m5} R_o R_L \)
- \( a_1 = -R_{o1} R_L \left( C_c g_{m1} + C_{gd5} g_{m1} + C_{gd1} g_{m5} \right) \)
- \( a_2 = R_{o1} R_L C_{gd1} \left( C_c + C_{gd5} \right) \)

- \( b_0 = 1 \)
- \( b_1 = R_L \left( C_L + C_c + C_{gd5} \right) + R_{o1} \left( C_c + C_{gd1} + C_{gd5} + C_{gs5} + C_{o1} \right) + g_{m5} R_{o1} R_L \left( C_c + C_{gd5} \right) \)
- \( b_2 = R_{o1} R_L C_L \left( C_c + C_{gd1} + C_{gd5} + C_{gs5} + C_{o1} \right) + R_{o1} R_L C_c \left( C_{gd1} + C_{gs5} + C_{o1} \right) + R_{o1} R_L C_{gd5} \left( C_{gd1} + C_{gs5} + C_{o1} \right) \)

Usually, coefficient \( a \) and \( b \) are related to each other, when considered as independent ones, the predicted bounds are overestimated.
Sufficiently large Monte Carlo samplings are taken to obtain:

- Bounds (green lines) using coefficient parameters of transfer function.

- Bounds (red lines) using circuit parameters \( (g_m, g_{ds}, g_{mb}, C_{gs}, C_{ds}, \text{etc}) \).

Bounds obtained from the coefficient space overestimate the exact bounds, since the dependency between the coefficients of transfer function.
Kharitonov’s theorem based method for circuit robustness analysis accounting for parameter uncertainties
  • Evaluate bounds of circuit frequency responses
  • Based on linearized closed form equations
  • Super efficient

Optimization based method for circuit robustness analysis considering parameter uncertainties
  • Realistic, in transistor level
  • General, applicable to both linear and nonlinear circuits, in both time and frequency domain
  • Efficient, faster than many currently available methods

Optimization based method for circuit design automation
  • Practical, General and Efficient
1. Circuit with parameter variations

2. Linearized circuit network equations through modified nodal analysis.
   \[ A(p)x = Bu, \quad y = C^T x \]

3. Interested output \( x(s, p) \) or relationship of two variables.
   (transfer function)
   \[ H(s, p) = \frac{x_1(s, p)}{x_2(s, p)} = \frac{\sum_{i=0}^{k} a_i(p)s^i}{\sum_{j=0}^{m} b_j(p)s^j} \]

4. Parameter space \( p = \{p | p_i \in [p_i^-, p_i^+], i = 1, 2, \ldots, n\} \) is converted to coefficient space \( a = \{a | a_i \in [a_i^-, a_i^+], i = 1, 2, \ldots, k\} \) and \( b = \{b | b_i \in [b_i^-, b_i^+], i = 1, 2, \ldots, m\} \)

5. Kharitonov’s theorem based performance robustness analysis in frequency domain.
   (The envelop of a system of polynomials with independently varied coefficients can be evaluated with four Kharitonov’s polynomials).
   \[ K_1(s) = q_0^+ + q_1^+ s + q_2^- s^2 + q_3^- s^3 + \ldots \]
   \[ K_2(s) = q_0^+ + q_1^- s + q_2^- s^2 + q_3^+ s^3 + \ldots \]
   \[ K_3(s) = q_0^- + q_1^+ s + q_2^+ s^2 + q_3^- s^3 + \ldots \]
   \[ K_4(s) = q_0^- + q_1^- s + q_2^+ s^2 + q_3^+ s^3 + \ldots \]

q equals a in numerator and q equals b in denominator.
The concerned output: the magnitude and phase responses

The transfer function derived from the linearized small signal model

\[ H(s, p) = \frac{N(s, p)}{D(s, p)} = \sum_{i=0}^{k} a_i(p)s^i \]

\[ = \sum_{j=0}^{m} b_j(p)s^j \]

\[ \text{abs}(H(s, p)) \text{ and } \text{arg}(H(s, p)) \]

Process variations, temperature changes, environmental noises, and etc.

Electrical, dimensional and process parameter variations \((V_{th}, W, L, \text{ and etc.})\)

Small signal model parameter variations \(g_m, g_{ds}, g_{mb}, C_{gs}, C_{ds}, \text{ and etc.}\)

Transfer function coefficient variation

\(a_0, a_1, \ldots, a_n\)

\(b_0, b_1, \ldots, b_m\)
- **Advantage**
  - Evaluate only several closed form Kharitonov-type polynomials.
  - **Obtain Exact bounds (theoretically guaranteed)** when the coefficients are independent.
  - Super-efficient, when comparing with Monte Carlo method.

- **Limitations**
  - Obtain the exact bounds only when the coefficients of transfer function are independent. Otherwise, it overestimates the results.
  - Depend on the approximated small signal model derived from the real circuit.
Study Case 1: Worst Case Oscillator Phase Noise Evaluation

- **Objective Function**: Oscillator is treated as an nonlinear time-varying model* (which is implemented in a commercial Circuit simulation package).

\[
S_v(f_m) = \sum_{k=-\infty}^{+\infty} |H_k(f_m)|^2 S_n(f_m \pm kf_0)
\]

- **Parameter space**: transistor level parameters (\(W, L, V_{th}, T_{ox}\), and etc.).

- Assume a stationary noise process with noise density spectrum \(S_n(f)\).

- Each noise component at \(f_m\) to integer number of \(f_0\) is modulated into phase noise at \(f_m\).

- \(H_k(f)\) is the transfer function from the \(k\)th noise component to the output.

- Process variations affect the operating of oscillator, the noise behaviors, and eventually deviate the phase noise from nominal value.