Lecture 14. Circuit Optimizers: Facts and Fallacies

Jaeha Kim
Mixed-Signal IC and System Group (MICS)
Seoul National University
jaeha@ieee.org

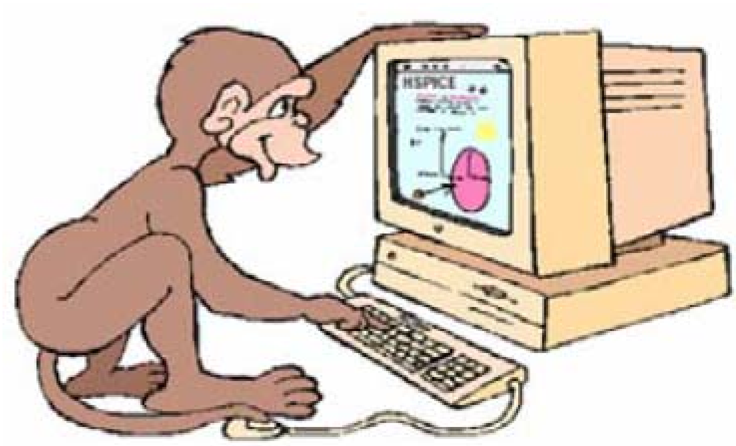
Outlines

- Readings
  - G. Gielen and R. Rutenbar, “Computer-Aided Design of Analog and Mixed-Signal Integrated Circuits,” Proceedings of IEEE, pp. 1825-1852, Dec. 2000.
- Overview
  - There are “circuit optimizers” which claim to automatically size your circuits for the best performance, but this class will describe why they are not your saviors. Despite their decade long history in the circuit field and maturity, they still have not made into the mainstream design flow and there are reasons. This class will examine some misconceptions about analog circuit optimizers and discuss what is the best way to utilize the techniques most effectively for circuit design. It is noteworthy that designer’s understanding of circuits is worth thousands of hours in simulation/optimization time

Circuit Optimizer: The Savior?

- Difficulty in analog circuit design is to realize the design intent in the presence of device non-idealities
  - Channel length modulation, carrier velocity saturation, short-channel effects, DIBL, proximity effects, mismatch/variation, …

- Drives many designers to become “SPICE monkeys”
  - Repeating so many simulations while sweeping parameters in the hope that one day a good design will be found
  - Can one automate this? – an optimizer!



[Courtesy Isaac Martinez]

Optimization in General

- A general optimization problem can be formulated as:

  minimize $f_0(x)$

  subject to $f_i(x) \leq b_i$, $i = 1, . . . , m$

  where

  $x = (x_1, . . . , x_n)$: optimization variables

  $f_0 : R^n \rightarrow R$: objective function

  $f_i : R^n \rightarrow R$, $i = 1, . . . , m$: constraint functions

- Optimal solution $x^\star$ has the smallest value of f_0 among all vectors that satisfy the constraints

Special Optimization Problems

- A certain types of problems have reliable and efficient algorithms to solve
- Least squares:

  minimize $||Ax - b||^2$

- Linear programming:

  minimize $c^T x$
  subject to $a_i^T x \leq b_i, \; i = 1, 2, ..., m$

- Convex optimization (where $f_0(x)$ and $f_i(x)$ are convex functions):

  minimize $f_0(x)$
  subject to $f_i(x) \leq b_i$, $i = 1, . . . , m$

Nonlinear Optimization Problems

- General nonlinear problems are much more difficult and often involve compromises
- Local optimization methods (nonlinear programming)
  - Finds a point that minimizes f_0 among feasible points *near* it
  - Fast, can handle large problems
  - Requires initial guess
  - Provides no information about distance to (global) optimum
- Global optimization methods
  - E.g. simulated annealing, genetic algorithms
  - Finds the (global) solution
  - Worst-case complexity grows exponentially with problem size

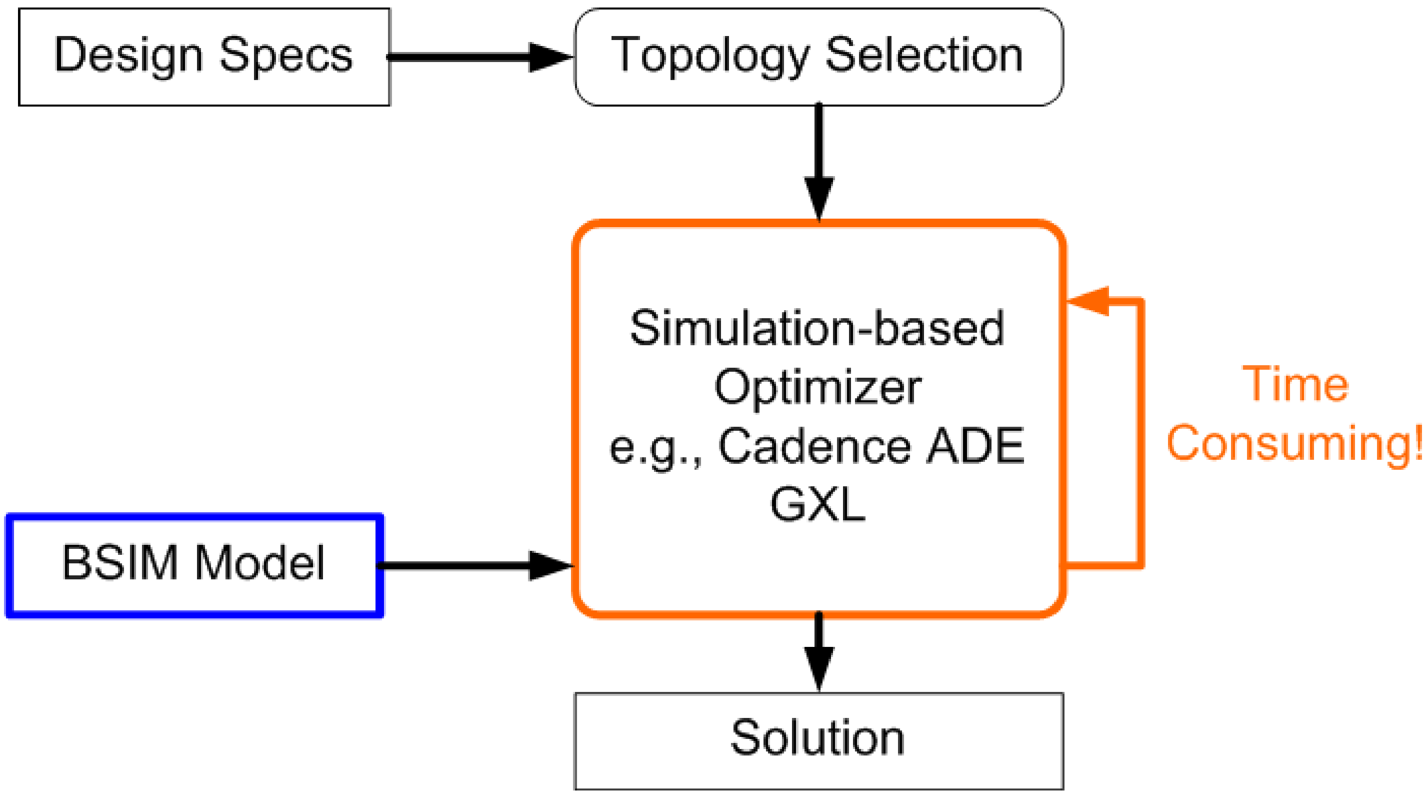
Circuit Optimization Problems

- Most existing optimizers address “sizing”, for example:
  - Design variables: device widths (W) and lengths (L)
  - Minimize power dissipation of a VCO
  - Subject to frequency range, output swing, area constraint, etc.

- Commercial circuit optimizers can be classified largely into two categories:
  - Simulation-based optimizers
  - Equation-based optimizers

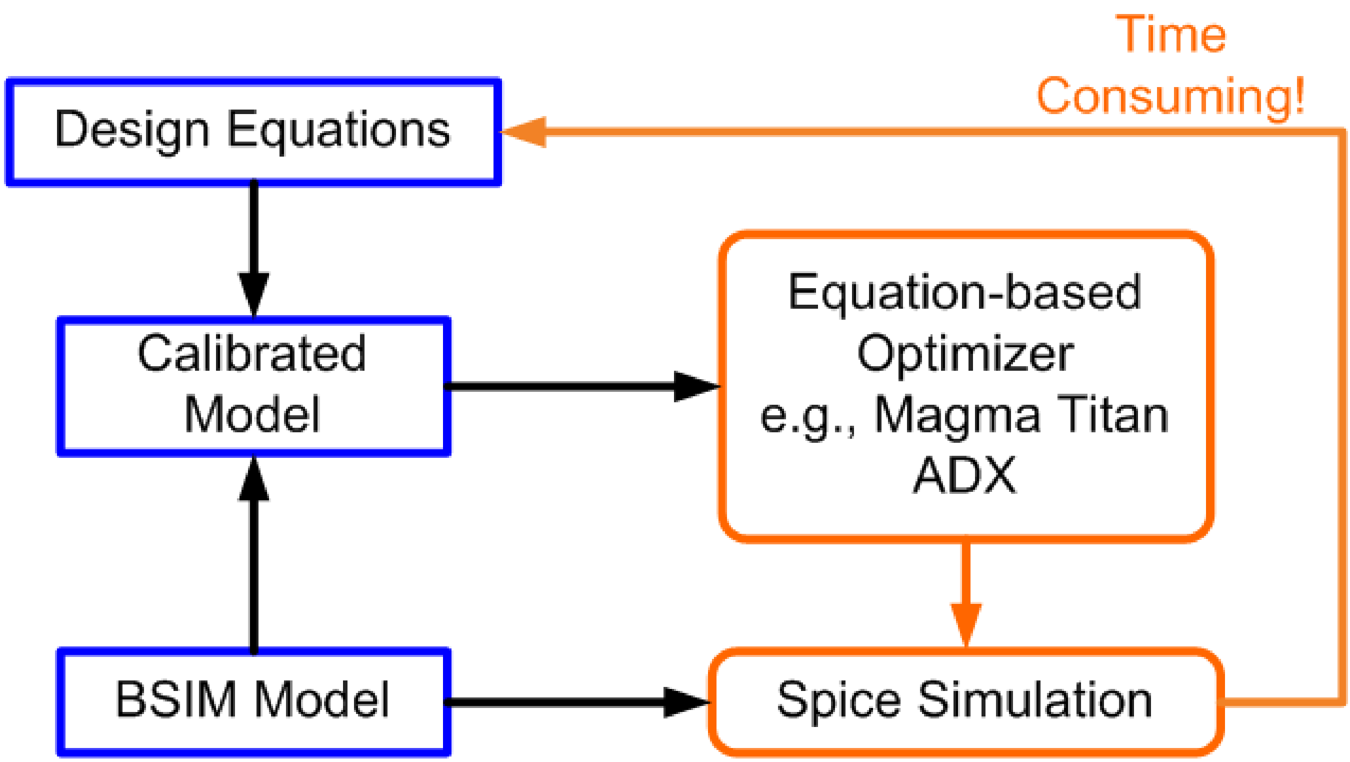
Simulation-Based Circuit Optimizers

- The objective/constraint functions are evaluated by circuit simulators
  - Time consuming; roughness



Equation-Based Circuit Optimizers

- The objective/constraint functions are written as analytical equations; fast only if you can write them in sufficient accuracy



Who Uses Circuit Optimizers?

Almost Nobody!

Then What Do Designers Want?

- Introducing our main characters:

Joe: novice analog circuit designer. Wants to improve his design productivity with automated circuit optimizers but got frustrated after a few trials.

Pat: analog EDA researcher. She has strong backgrounds in numerical and optimization techniques. Also frustrated with slow adoption of circuit optimizers despite their strengths.

Myth 1: Designers Want Faster Optimizers

"I like your optimizer but it took 5 days to finish. I will adopt it if it can finish in 1 day instead…"

Should Pat believe him and build a faster one?

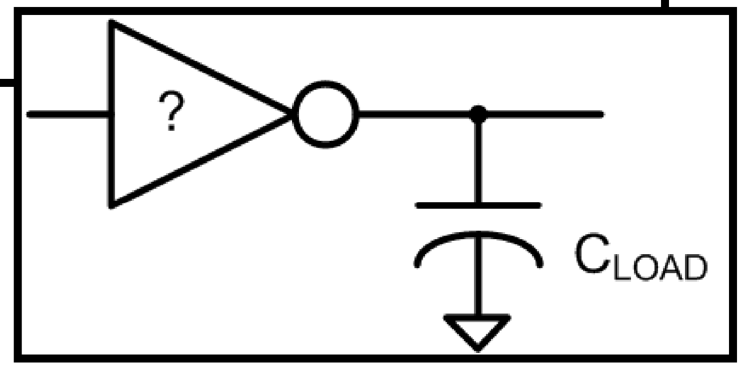
- No – Joe is expecting an almighty optimizer that can find the global solution in no time
  - Something that can't be delivered in the near future
  - And Joe will never be satisfied – you know he is also asking for faster SPICE after 3~4 orders of magnitude improvement!
- You'd help Joe more by not giving him a false hope
  - It's quite likely that Joe is abusing the optimizer – e.g. making simulations overly long, setting the variable ranges too wide, …
  - Teach him how to stop this; otherwise you can't win this race

Truth: Designers Want Quick Turn-Around

- Circuit design is an iterative process
  - All but the last answer will be thrown away
  - Exact values of those answers don't matter as long as we can correctly determine that they are wrong
- So is the process with a circuit optimizer
  - Key to its successful use is to ask the ***right question*!**
  - Otherwise, it is very good at giving you stupid answers
  - Designers keep refining the question until they get the answer they want
  - Don't want the optimizer to waste time finding bogus answers
- Can optimizers help shorten this cycle?

Example: Buffer Chain Optimization

Give me the fastest buffer chain that can drive a large load C_{LOAD}



That's simple –
a very large buffer!

Well – that's not the solution I was looking for. Now I need to figure how to drive this even bigger input load of this inverter…
Oh, I know – ***<u>I forgot to constrain</u>*** the input load of the buffer chain!

Possible Ways Optimizers Can Help

- Return quick-and-dirty estimates on the final answer
  - Based on simpler models or equations
  - e.g. in the buffer example, RC switch model would suffice
  - Help determine if the problem is fully constrained

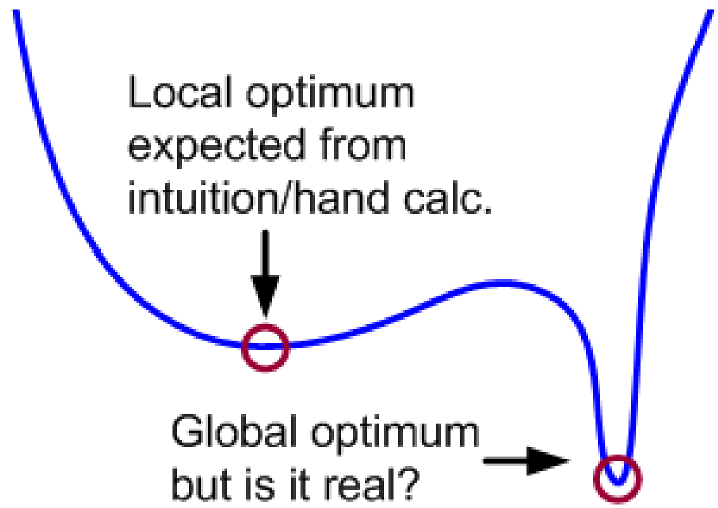
- Provide interactive sessions with designers
  - Show how the optimizer is trying to improve the current design and why
  - e.g. sizing up the buffer since it reduces the delay without being subject to any constraints
  - Check if it is consistent with what a designer would do

Myth 2: Designers Want Global Solutions

"My optimizer guarantees to find globally optimal solution regardless of initial guess. Sometimes, it will surprise you with solutions that no designers have ever found!"

Should Joe Feel Comfortable About This Optimizer?

- Not necessarily – Joe wonders if this surprising solution is the one he'd want and can trust
  - Isn't the optimizer exploiting some unspecified constraints or modeling artifacts?



Truth: Designers Need to Know More

- About the suggested solution and why it is optimal
  - Designers won't accept ones that they cannot understand
  - Esp. if the solution is different from what hand analysis or intuition provides
  - Investigate the discrepancy and refine the hand analysis models/equations until the answers match
  - Visual aids go a long way here
- Designers are worried about:
  - Solutions stuck at local optima
  - Bogus solutions due to missing constraints or wrong assumptions
  - Unrealistic solutions that are too sensitive to variations in parameters or operating conditions

Myth 3: Designers Need Precise Solutions

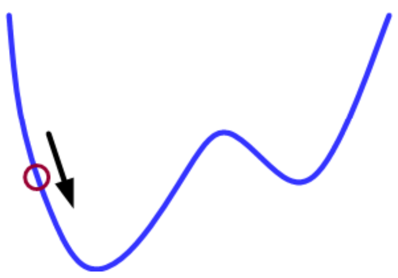
"Here are solutions found by the optimizer: W1=32.147um and W2=19.898um. They were refined to your grid resolution of 0.001um."

"Thanks – I will use W1=32um and W2=20um."

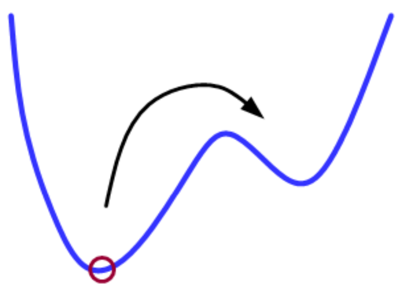
- Designers are aware of model uncertainties and that it doesn't pay off to set the parameter values to their last digits
- Then, why should the optimizer use precise values?

Why Optimizers are Slow

- Modern optimizers are all about:
  - Finding the local optimums
    - Evaluate at finer steps while converging to the local optimum



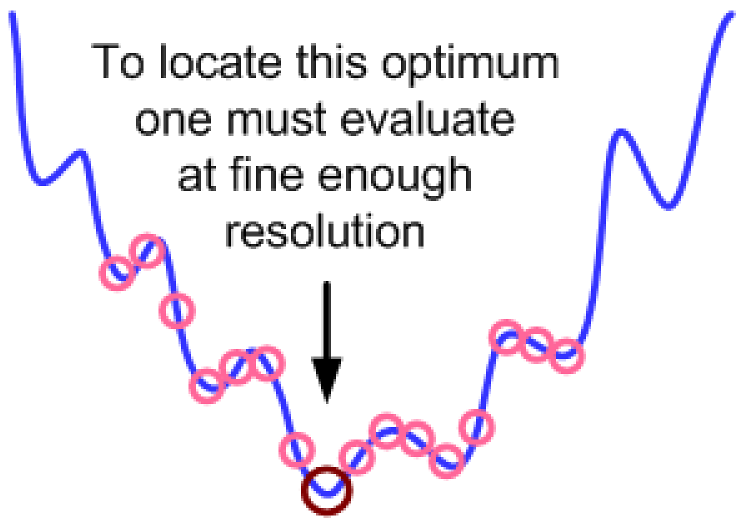
  - Getting out of the local optimums
    - Relying on randomness to try unexplored area and eventually find the global optimum

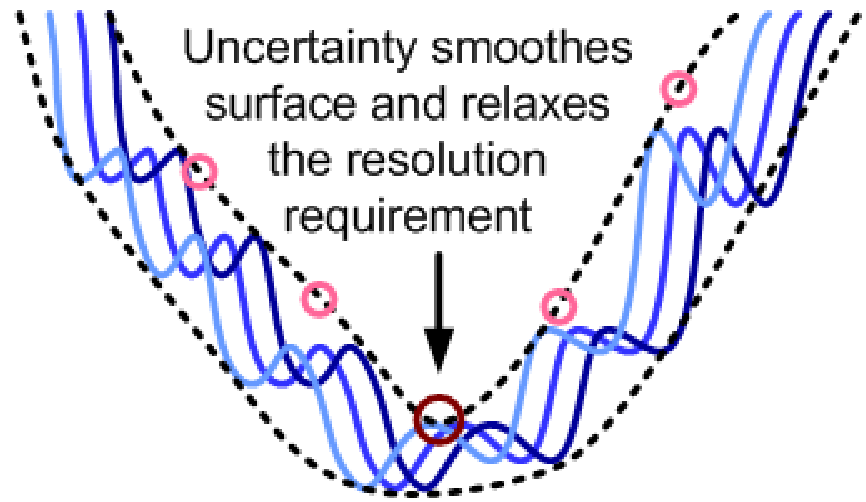


- Can waste computational efforts
  - Refining the solution to the irrelevant precision
  - Following unsuccessful escape attempts

Variability May Ease the Problem

- Model uncertainties and parameter variations effectively smoothe out the objective function
  - It is not possible to have finely-spaced local optima
- Global optimum can perhaps be found via coarse-gridded search and interpolation





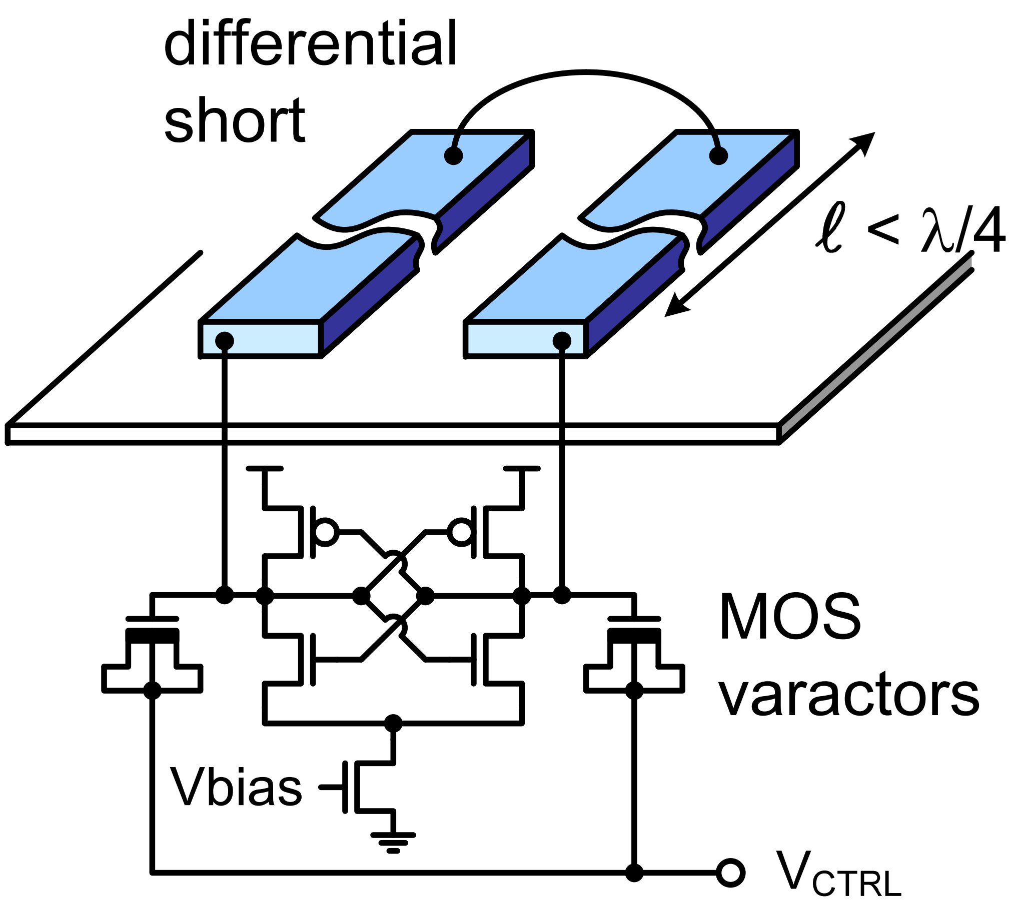
Myth 4: Circuit Problems are Many-D

"Analog circuits involve many design parameters such as device sizes. Therefore, optimizers must be good at solving high-dimensional problems…"

"That sounds logical, but I wonder how on earth people have been designing circuits so far? I bet none of my colleagues can solve many-parameter problems by hand…"

- Designers simplify circuit problems into ones with small number of parameters
  - Only that way, they can visualize trade-offs and build insights
  - Can optimizers help in this regard?

Example: Optimization of 20GHz LC-VCO

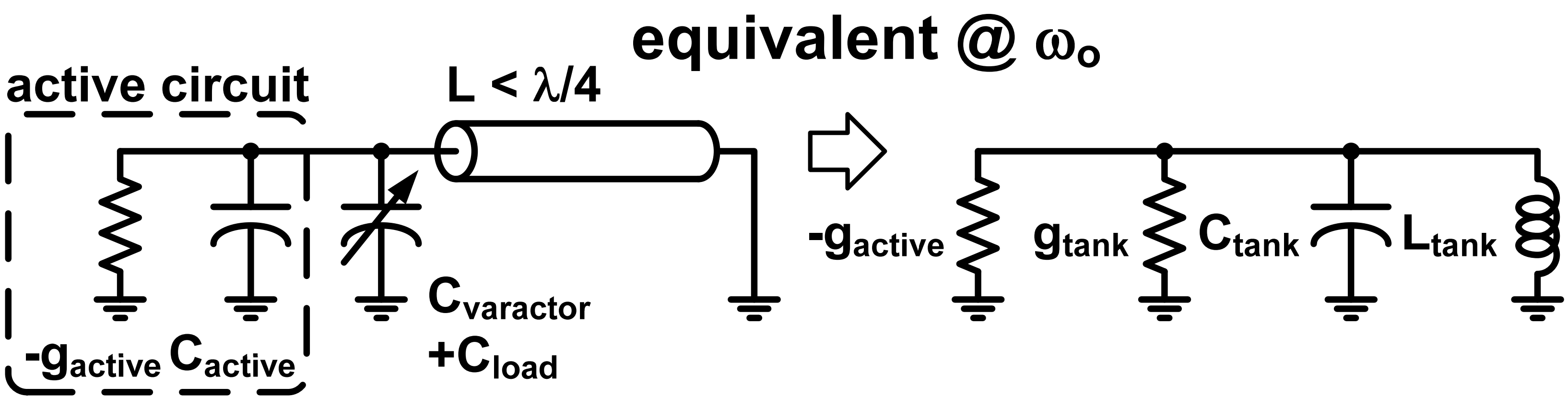


J. Kim, et al., "A 20-GHz Phase-Locked Loop for 40-Gb/s Serializing Transmitter in 0.13-µm CMOS", JSSC 04/2006.

- Negative-R LC-resonant oscillator
  - Used coupled microstrips as inductive element ($\ell<\lambda/4$; short-circuited)
- Design parameters:
  - Active device sizes
  - Varactor sizes
  - Transmission line geometry (e.g. length ℓ)

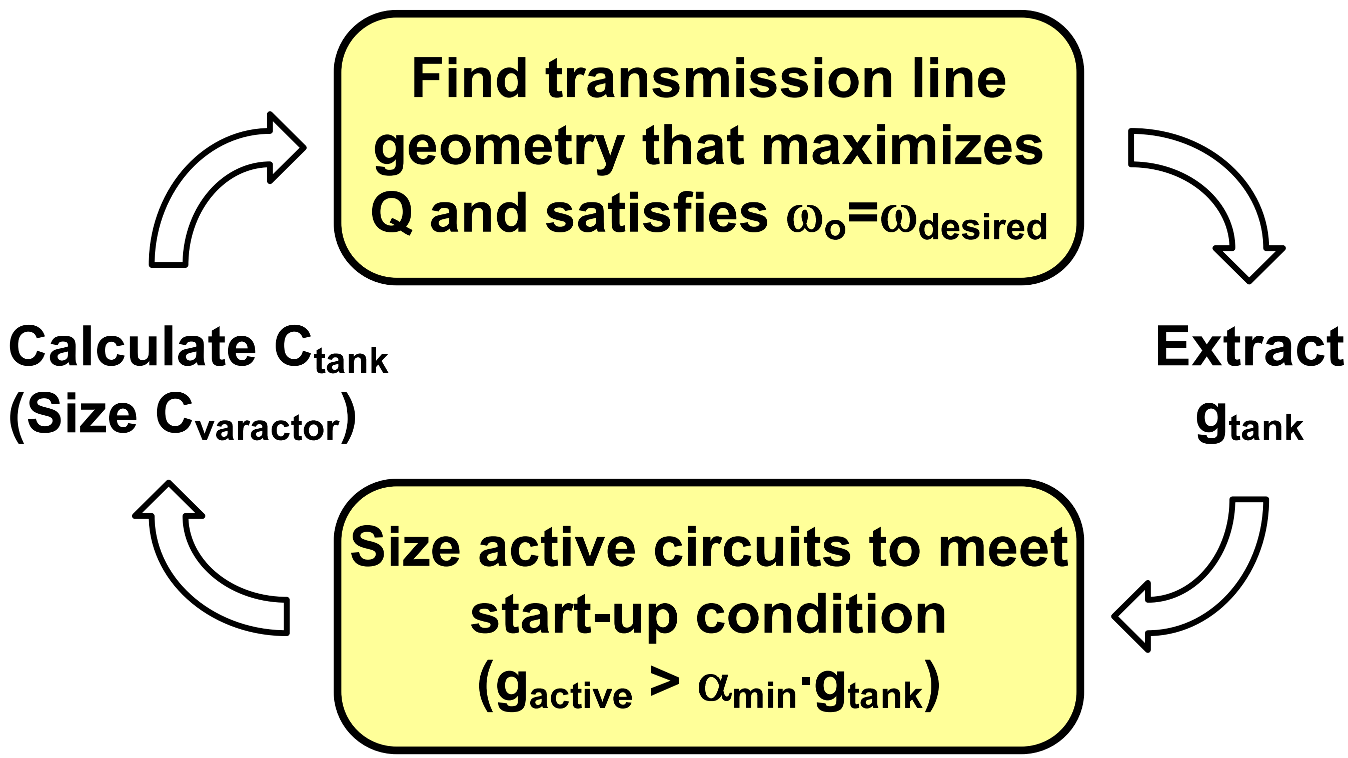
VCO Design Constraints

- Power : I_{bias} ($\leq$10mA)
- Start-up : $g_{active} \geq \alpha_{min} \cdot g_{tank}$ (α_{min} = 3~5)
- Osc freq : $\omega_o^2 = 1/L_{tank}C_{tank}$ (20GHz)
- Tuning range : $C_{varactor}/C_{tank}$ (+/-10%)
- Minimize phase noise

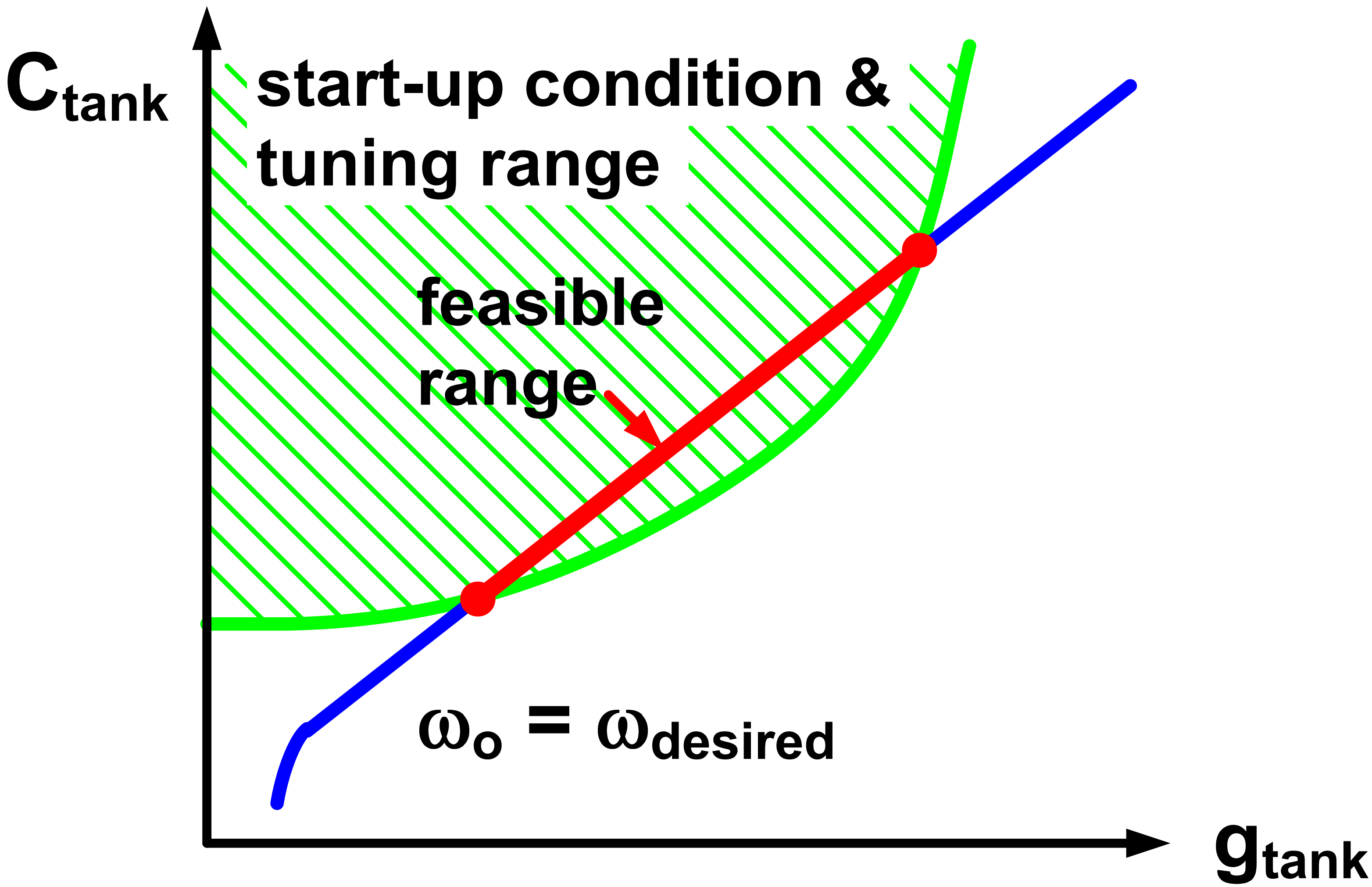


VCO Optimization via Iteration

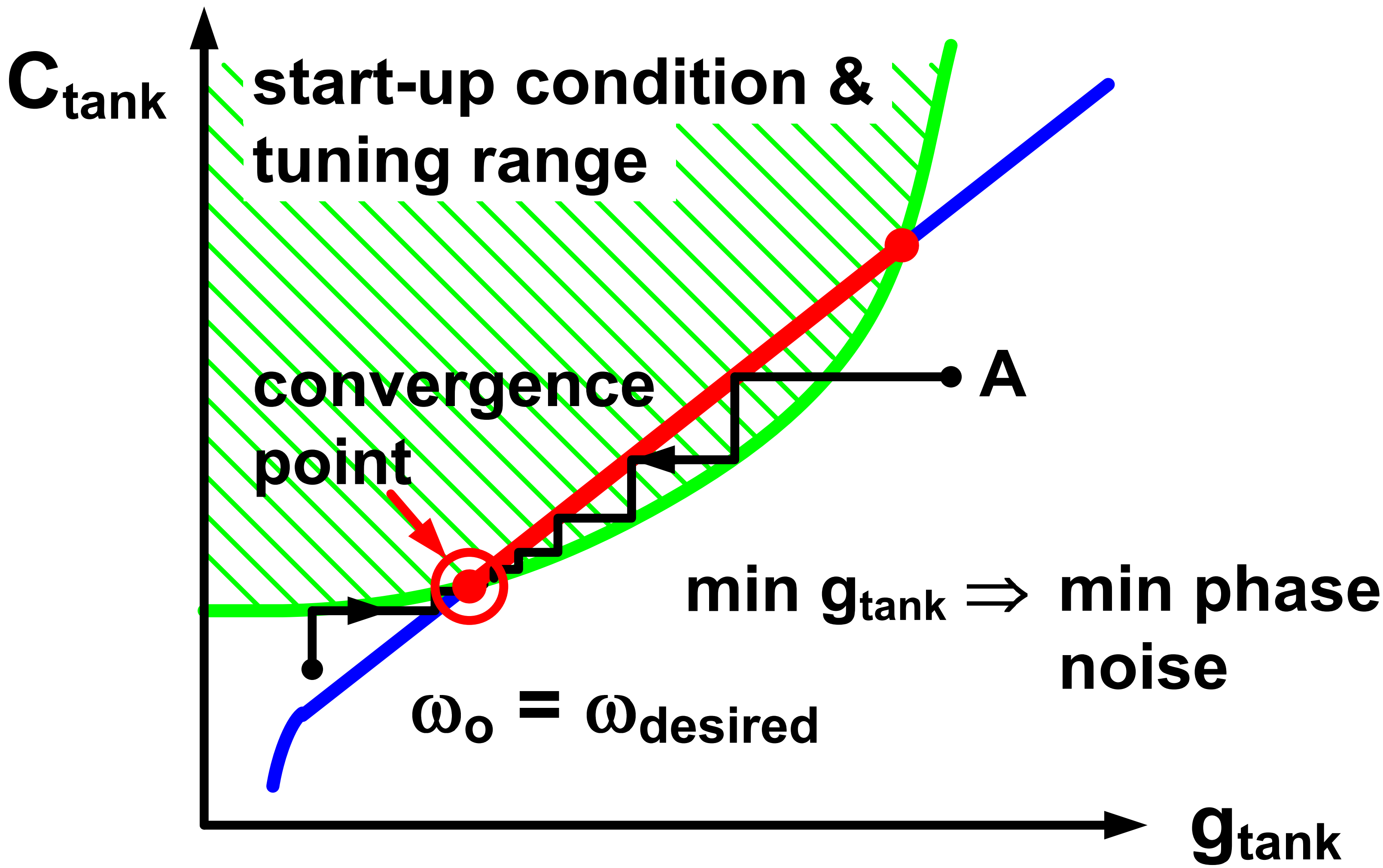
- The problem can broken into two sub-problems
  - With only two independent variables – $\mathbf{C_{tank}}$ and $\mathbf{g_{tank}}$
  - The other variables can be determined from $\mathbf{C_{tank}}$ or $\mathbf{g_{tank}}$



Graphical Interpretation



Iteration Minimizes Phase Noise



Problem Reducer?

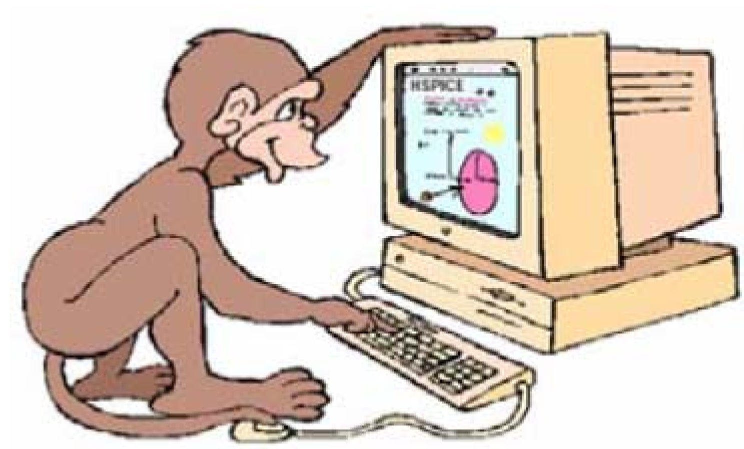
- A helpful tool for designers might be the one that:
  - Takes a circuit problem and makes it simpler
  - Determine the dimensionality of the feasible solution space and identify its key variables
  - Hopefully, # of key variables is only 2~3 so the feasible space can be visualized easily

"Optimizers already do that when solving a problem with constraints – they reduce the search space. There are also many available techniques for dimensionality reduction."

"That's a great news – can designers access some of those information and build insights?"

Intent-Leveraged Circuit Optimization

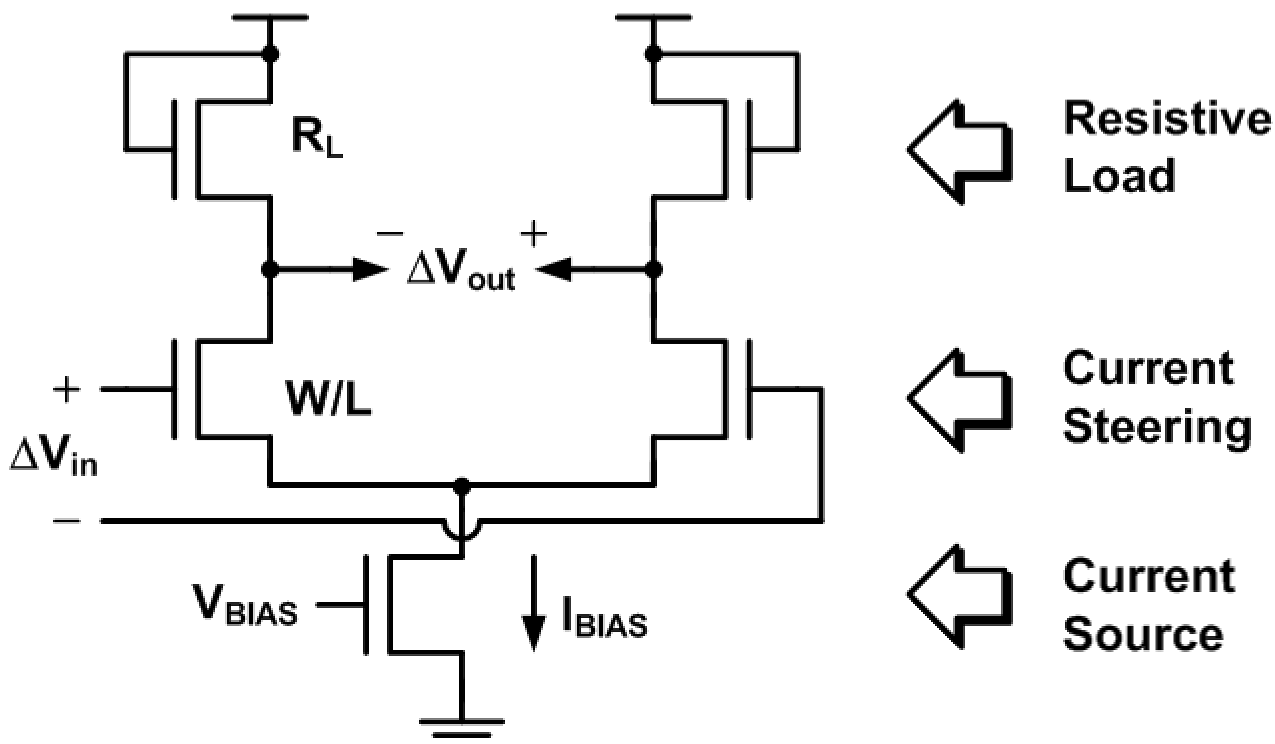
- Main difficulty in analog circuit design is to realize the design intent in the presence of device non-idealities
  - What is the best incarnation of my ideal circuit in this world?
  - Can optimizers help answer this?
- Some designers address this challenge by becoming “SPICE monkeys”
  - Running blind, exhaustive simulations without understanding how circuits really work
  - Most optimizers just help them being more efficient monkeys



[Courtesy Isaac Martinez]

Design Intents of Analog Circuits

- Every device has its own purpose of existence
  - Even though they may all look the same
  - This is the design intent – valuable information itself

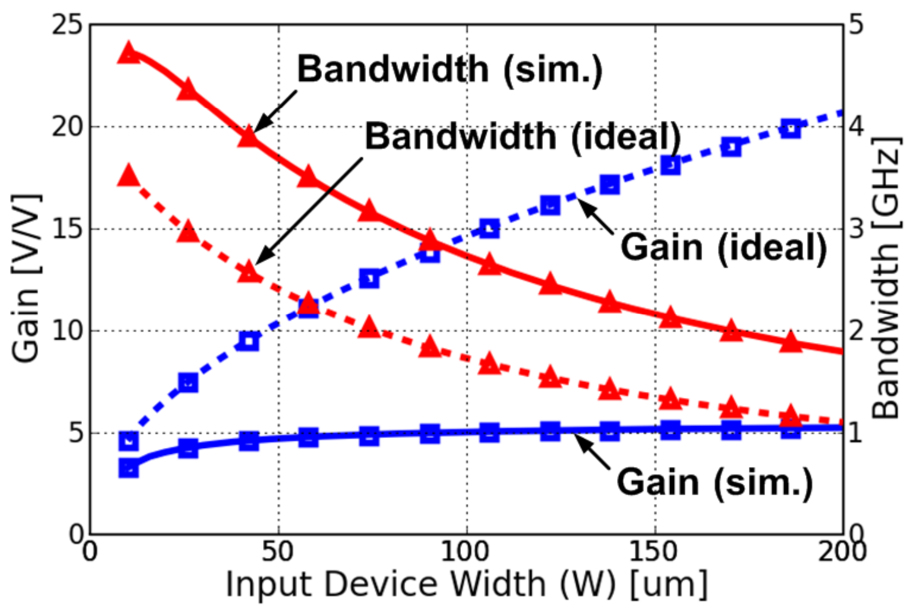


Circuit Design Would Be Easy If...

- The devices behave just as intended
  - The current source flows I_{BIAS}, the load has R_L, and the input device has G_m following the long-channel model
  - We can derive simple equations and find optimal solutions
- Of course, the reality bites
  - Errors are likely due to various realistic effects
  - But most intuition still works

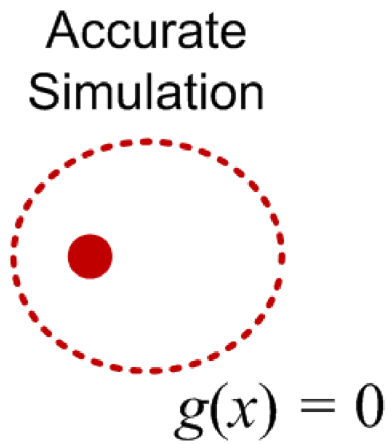
$$Gain = G_m \cdot R_L = \sqrt{\mu C_{ox} \frac{W}{L} I_{BIAS}} \cdot R_L$$

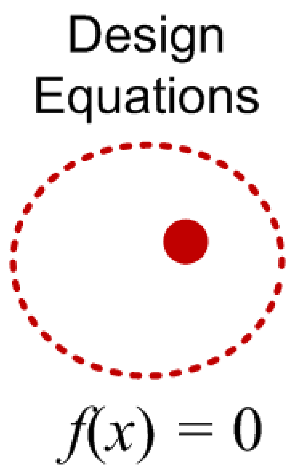
$$Bandwidth = \frac{1}{2\pi \cdot R_L \cdot (C_L + C_j \cdot W)}$$



Circuit Optimization via Homotopy

- How do we leverage the solution from the ideal design equations to find the real solution based on simulation?
  - It may not necessarily serve as a good initial solution

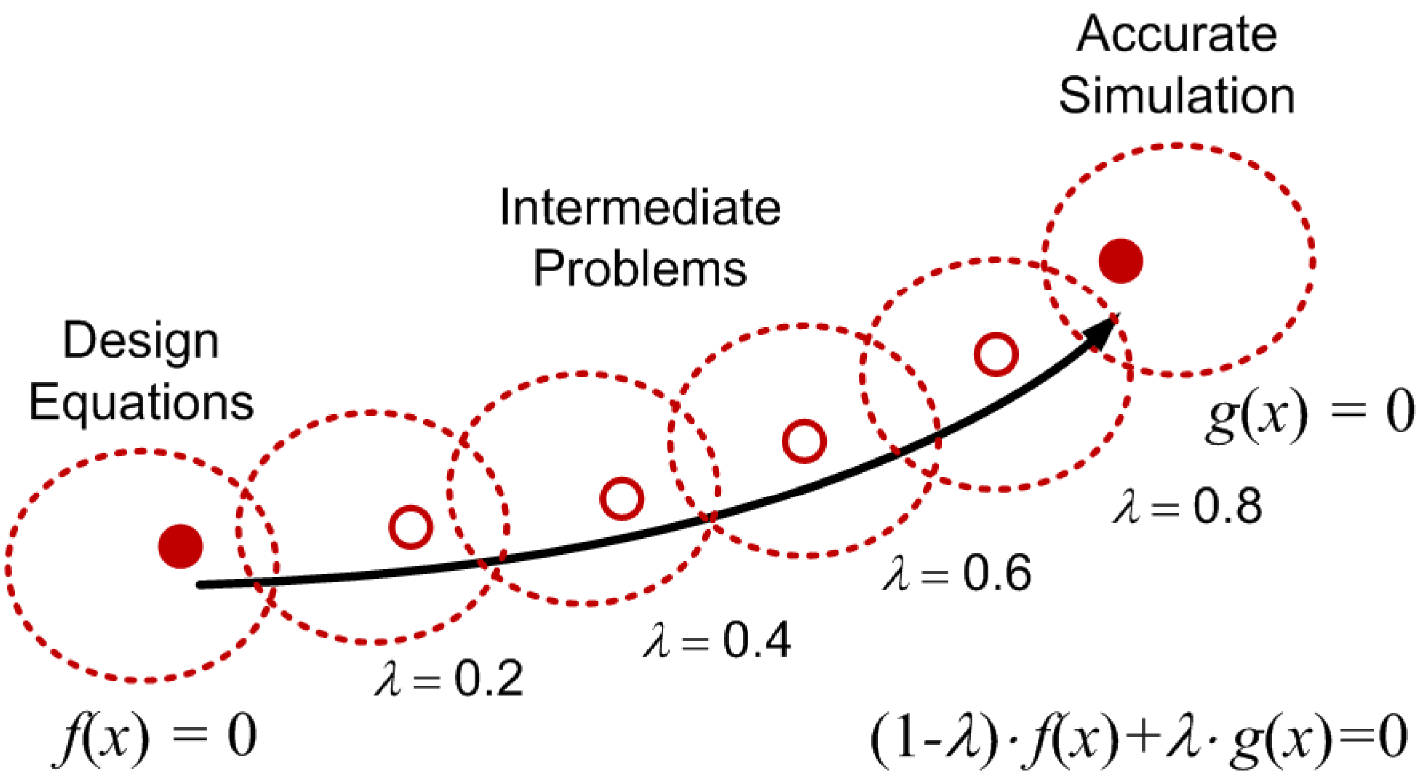




M. Jeeradit, et al., “Intent-Leveraged Optimization of Analog Circuits via Homotopy”, DATE 2010.

Circuit Optimization via Homotopy (2)

- Create a set of intermediate problems in-between and solve them sequentially leveraging previous solutions
  - Quick refinement of the solution at each step



Promising Results

- Intent-based design equations make good guides to the real solutions even when their answers are off

Ckt	# of vars / constrs	# of evaluations		
		Intent-based Homotopy	Fixed-point Homotopy	Sequential Convex
Ring Osc	2 / 3	36	44	26
Supply Regulator	7 / 3	349	1935	681
Clocked Comparator	7 / 2	573	688	Not feasible
2-stage OpAmp	8 / 7	62	149	153
PLL	11 / 4	1051	2463	1183

Summary

- Key to effective use of circuit optimizers is to align the expectations between the users and developers
  - Designers haven’t been expressing well what they want
  - Circuit optimizers are powerful but don’t know how to help
- Knowing the real needs breaks the problem into smaller pieces:
  - How do I check if I setup the right optimization problem?
  - Will my circuit perform well at least in the ideal world?
  - If yes, how do I find its closest incarnation in the real world?
  - Can the optimizer help me understand the circuit better?
    - What did I miss in the design equations?
    - What are the key variables governing the trade-offs?