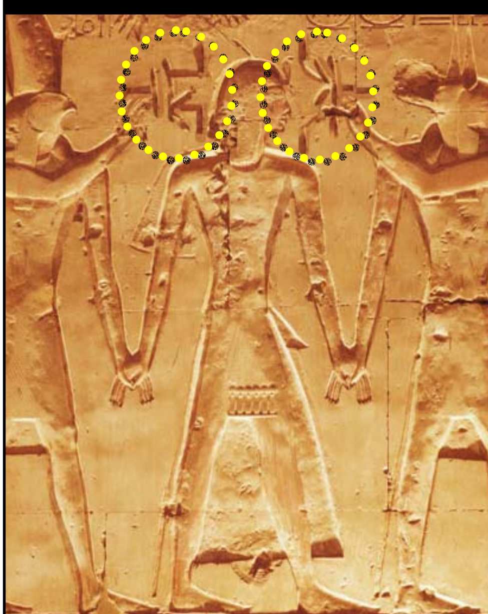
BMAS 2003
The Glorious Future of Macromodeling (In a Nutshell)

Rob A. Rutenbar

Professor, Electrical & Computer Engineering

rutenbar@ece.cmu.edu

The *Past* of Macromodeling…



- Not without its own glories…
- Not always easy to decipher
- Not always easy to use…

The Questions are Ancient, and Profound …

- Archaeologists' reconstruction
- New excavation: Temple of Ductorh

Earliest known attempt at 3D full-wave model equations—sadly, incomplete

 •  = 

Remarkably ahead of its time…

The Essential Themes Recur Thru History…

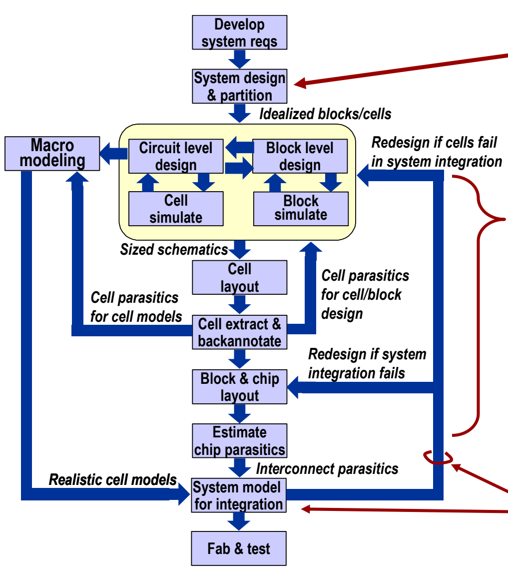
"*The Allegory of St. Darlington,*" from Pane 37, North window, Central vault, Notre Linear

So Why A “Glorious *Future*”…?

1. Exciting *new* ideas

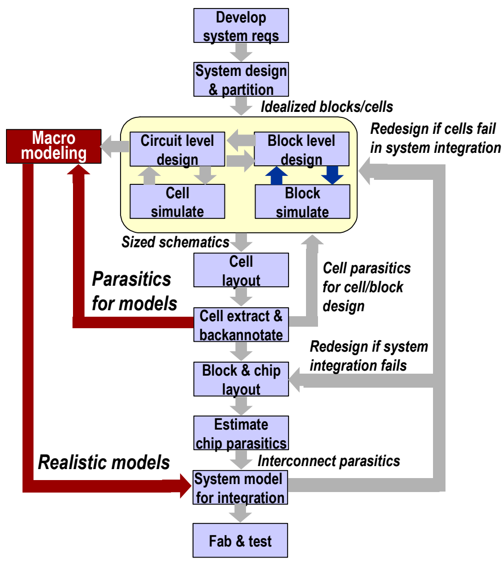
- Projection-based nonlinear MOR
- Volterra-based nonlinear MOR
- Time and frequency modeling
- Noise modeling
- Symbolic modeling
- Design space modeling
- Data mining concepts
- Synthesis applications
- Etc etc etc

No Choice…? Today, Flows Look Like This



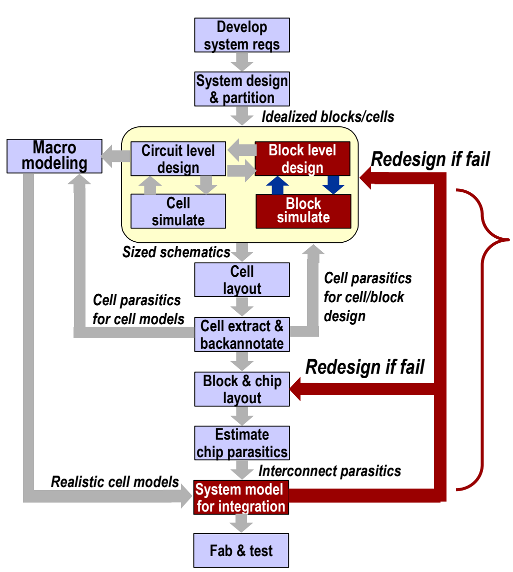
- **We design systems with ideal blocks, and simulate to verify**
- **Take each block down to a circuit**
  - Manually, or with emerging synthesis technologies
- **Replace ideal blocks with better models, integrate, and re-simulate**
  - And, often, it **doesn’t work**…

Model-Building Methods Still Weak, Limited



- Models often still done **by hand**, by experts who understand the circuits and the system

Model-Building Methods Still Weak, Limited

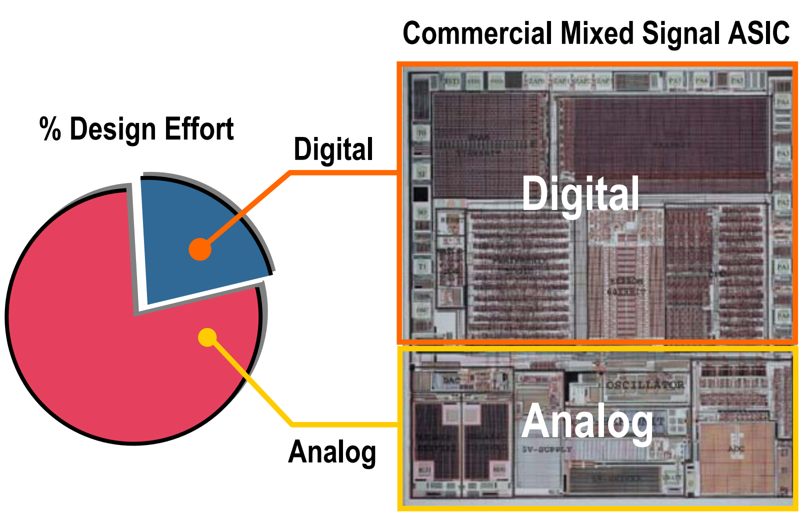


- Especially problematic as we start to rely on macromodels for full-system design **closure**

About This Talk…

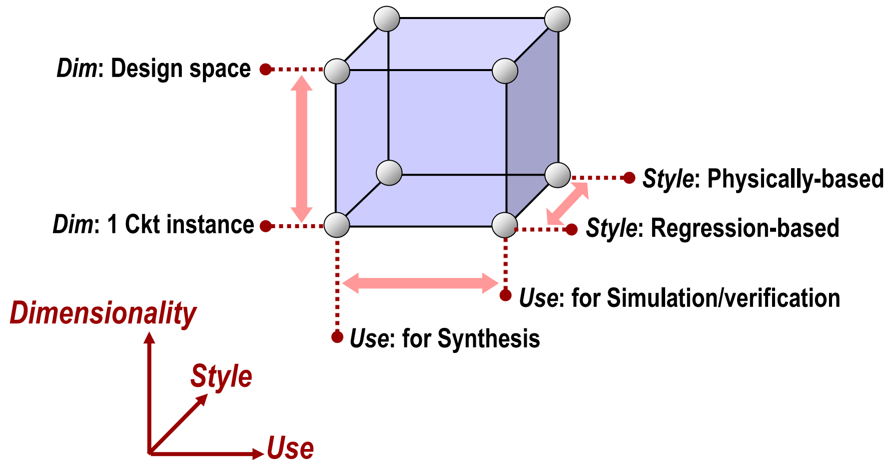
- Pieces of macromodeling puzzle
  - High-level taxonomy of approaches
- New piece of puzzle: *data mining* ideas
  - What they are, where they seem useful

Some Context: My Focus is Mixed-Signal



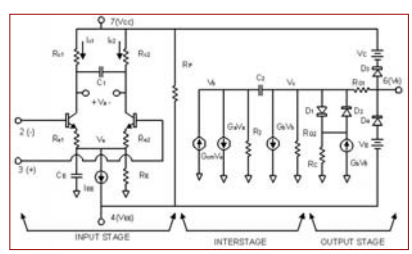
What Kind of “Macromodel” Do We Mean?

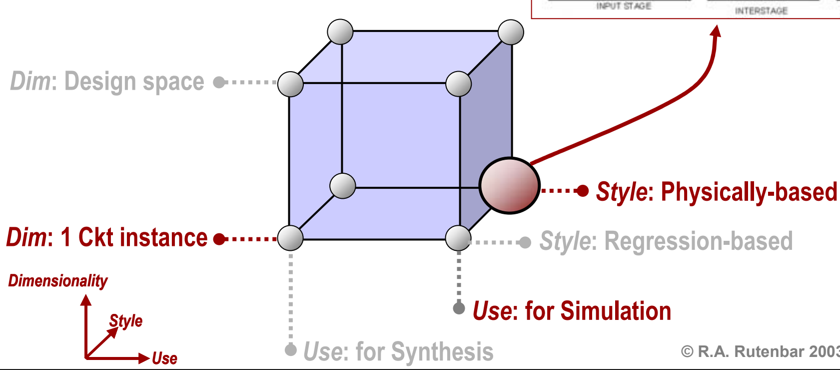
- Term “macromodel” means different things to different people…



Earliest Models: Physical, for Simulation

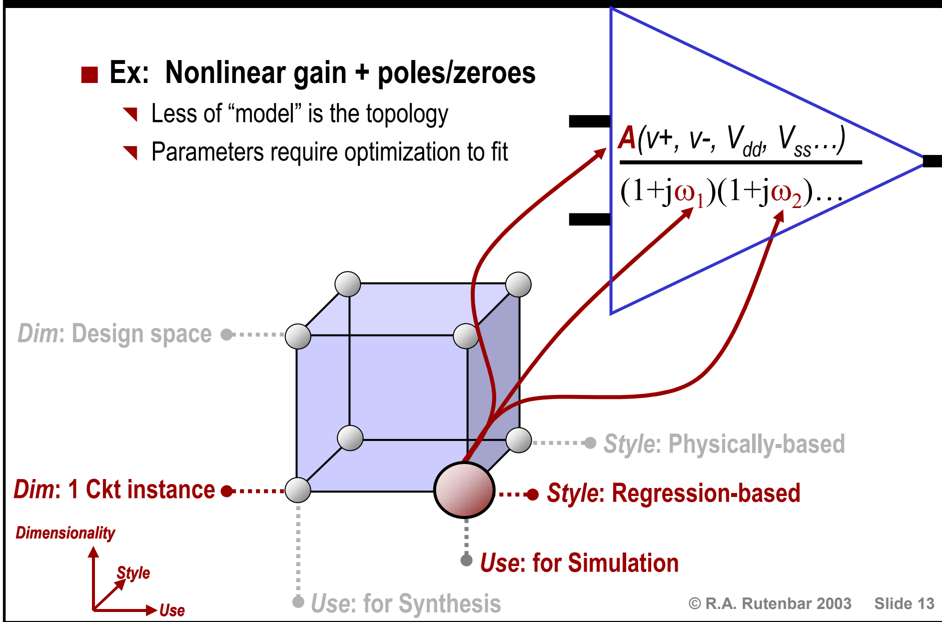
- Ex: Boyle opamp [JSSC Dec’74]
  - Most of the “model” is circuit topology
  - Parameters easy to pick off original ckt





Later Models: More Regression, for Simulation

- **Ex: Nonlinear gain + poles/zeroes**
  - Less of "model" is the topology
  - Parameters require optimization to fit



Essential Problems in All Macromodeling

- **What do you fit *to***
  - What's the template/structure?
- **Options**
  - A simplified circuit
  - A "black box" curve fit
  - A nice mathematical form
    - Pole zero form H(s)
    - Linear ODE
    - Nonlinear ODE
    - Volterra series
    - …

- ***How* do you fit to it?**
  - What's the mathematics of "fit"?
- **Options**
  - By hand (ask someone smart)
  - Nonlinear regression
  - A nice mathematical recipe
    - Linear: **AWE, PVL, Arnoldi, PRIMA…**
    - Nonlinear: **Volterra, posynomial…**
  - Special purpose (common circuits)
    - **Opamp, VCO, mixer, PFD-CP**
    - …

Today's Spectrum of Modeling for Simulation

$$\frac{A(v+, v-, V_{dd}, V_{ss}\ldots)\,(1+j\omega_a)(1+j\omega_b)\ldots}{(1+j\omega_1)(1+j\omega_2)\,(1+j\omega_3)\ldots}$$

Nonlinear Regression Fitting

Transfer Function Projection / nonlin MOR

Today's Spectrum of Modeling for Simulation

$$\frac{A(v+, v-, V_{dd}, V_{ss}\ldots)(1+j\omega_a)(1+j\omega_b)\ldots}{(1+j\omega_1)(1+j\omega_2)\,(1+j\omega_3)\ldots}$$

Pro: handling strong nonlinearities

Con: handling dynamics, accuracy guarantees, training data

Nonlinear Regression Fitting

Transfer Function Projection / nonlin MOR

Today's Spectrum of Modeling for Simulation

$$\frac{A(v+, v-, V_{dd}, V_{ss}\ldots)(1+j\omega_a)(1+j\omega_b)\ldots}{(1+j\omega_1)(1+j\omega_2)\,(1+j\omega_3)\ldots}$$

Pro: handling dynamics, xfer functions, accuracy, less training
Con: handling wide range of strong nonlinearities

Nonlinear Regression
Fitting

Transfer Function
Projection / nonlin MOR

End-Use Diversity Complicates Problem

Intent

Accuracy
vs Speed

Domain

Time
Frequency
Statistical (noise)
…

Fitting

Regression
Projection
Nonlin MOR
…

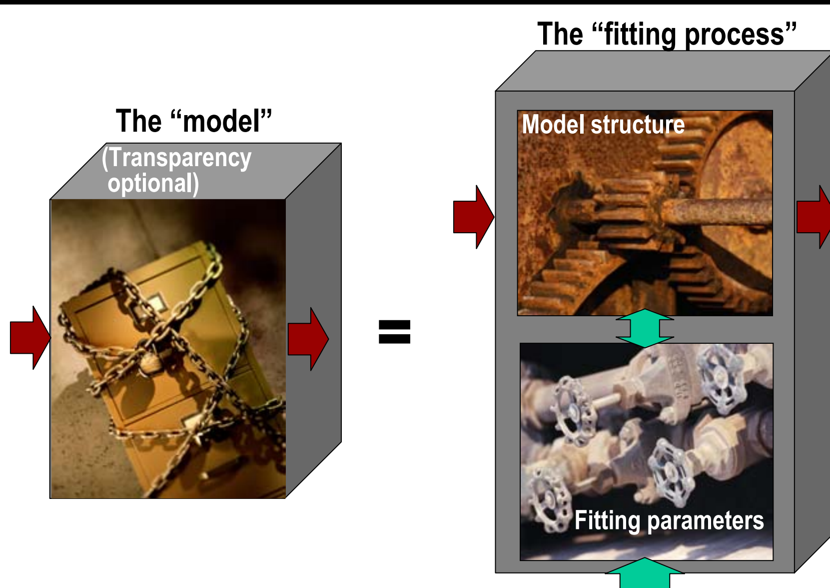
Simulation Style

Signalflow: $Y(s)=H(s)X(s)$
Dynamics: $\dot{g}(x) = f(x) + Bu$
Noise:

…As Do User Expections

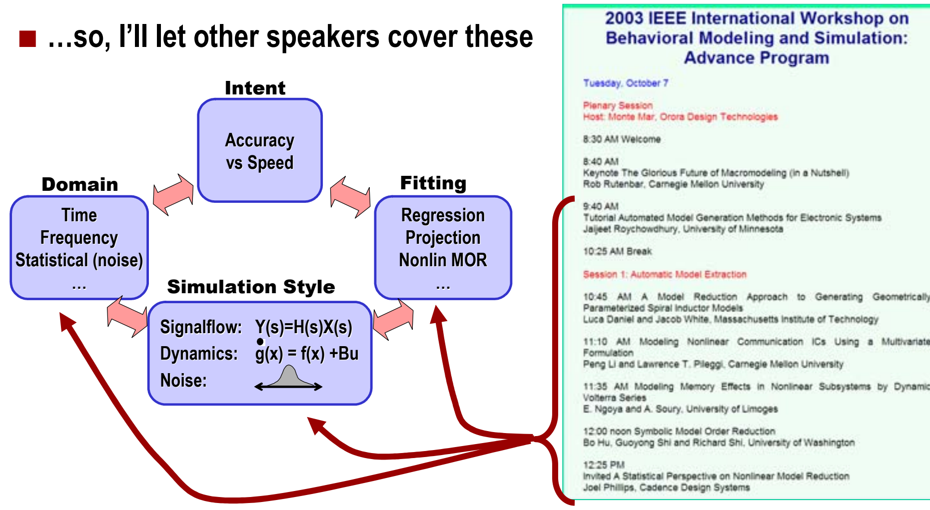
- Like simulator controls, everybody hates having to “twiddle” model details
- This is what every designer *ultimately* wants

Today’s Models Don’t Always Inspire Confidence

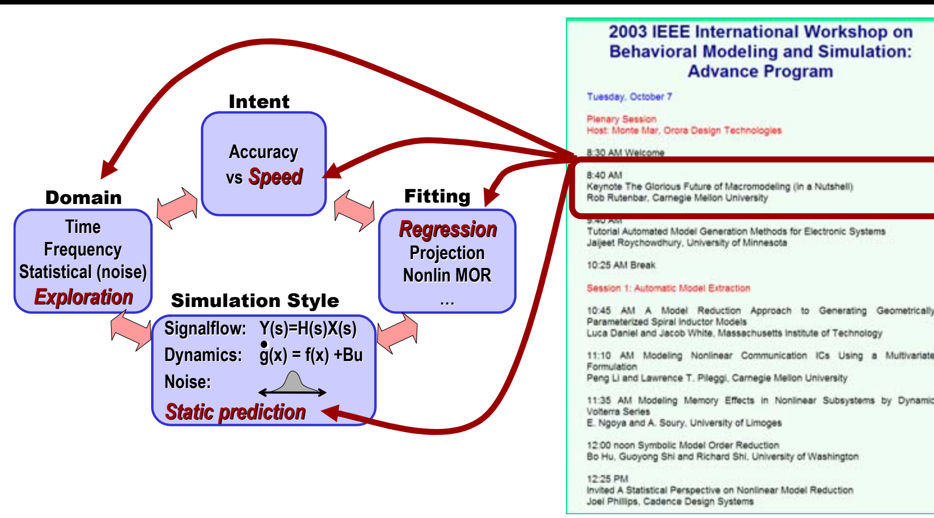


…But, Recent Ideas Offer New Hope

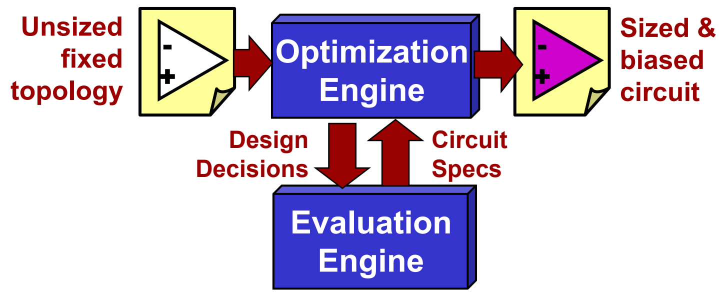
- …so, I'll let other speakers cover these



…and I'll Talk About Something *Different*…



New Motivation: *Large* Analog Design Spaces

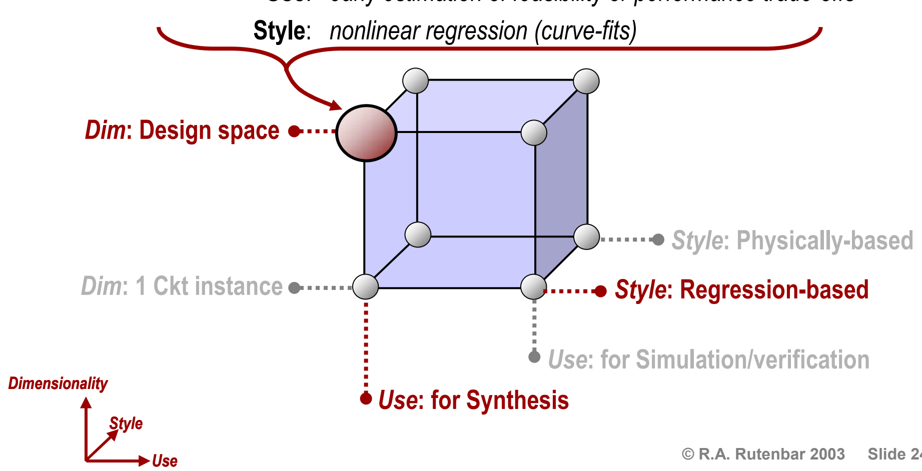


- Emerging commercial infrastructure makes it easy to sample large populations of fully qualified (*simulated*) circuit configs
  - **Optimization engine:** proposes circuit solution candidates
  - **Evaluation engine:** evaluates quality of each candidate
  - **Full SPICE evaluation:** for every spec, for every candidate

Macromodeling in this New Context

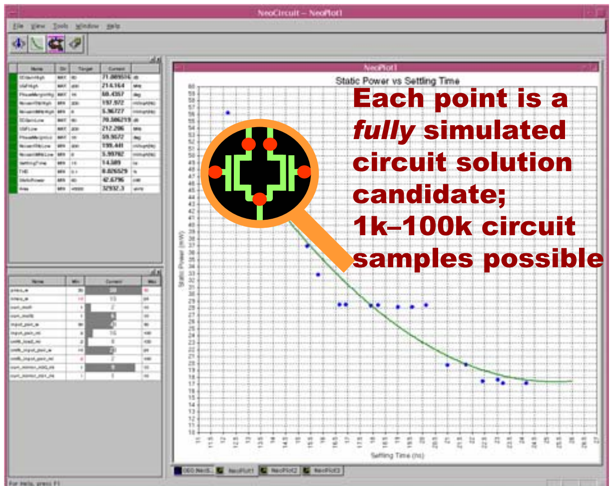
- Modeling analog **design spaces** for synthesis uses

Dimensionality: *populations of circuits, high-dimensional data*
Use: *early estimation of feasibility or performance trade-offs*
Style: *nonlinear regression (curve-fits)*



New Opportunity: *Design Space Modeling*

- Can generate 1K – 100K fully-qualified (simulated) samples of large analog design spaces – what can we do with this data?



Courtesy Neolinear, Inc.

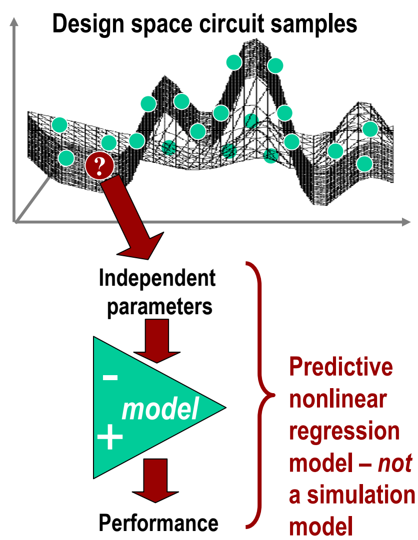
New Idea: Macromodeling as *Data Mining*

- **Problem:** Can we fit these large populations in useful ways?
  - 1K – 100K data points? Conventional, *ad hoc* curve fitting not up to the task
  - Where do we look to do better?

- **Solution:** Data mining
  - Significant breakthroughs in last 5 years in this community
  - Techniques for extracting (fitting) patterns, predictive formulas, or classifiers to large amounts of high-dimensional, possibly noisy, data

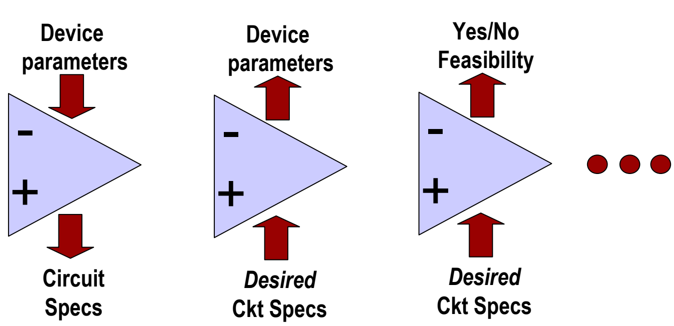
Basic Problems in Nonlinear Regression

- **Data selection**
  - Where does data come from?
  - **Focus: simulation-based search**
- **Model selection/fitting**
  - What functional form to fit populations of high-dim ckts?
  - **Focus: *boosted regressors***
- **Model validation**
  - How do you tell if you "fit well"?
  - **Focus: *rigorous model selection***



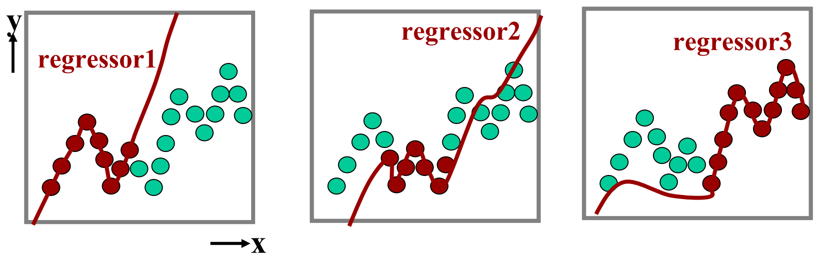
Aside: this is "Black Box" Modeling

- **We want to assume very little about base model template**
  - **Black box** = from data points alone, deduce and parameterize a model that accurately predicts basic behavior



Essential Regression Problem

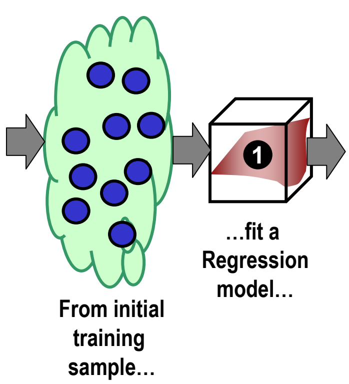
- **Very hard to find one functional form that fits everywhere**
  - Often easy to find regressors that can fit locally, but not globally
  - Problem gets worse with more data, with more dimensions



- **Could we find, then combine, a set of such regressors?**

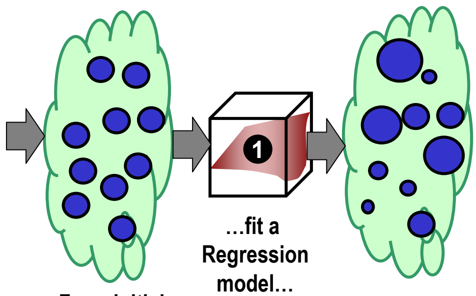
Elegant Strategy: *Boosting* [Freund, Schapire, 97]

- **Build a *sequence* of regression models, then *combine* them**
  - Strategy for sequentially resampling/reweighting data, voting = **boosting**



Ensemble Strategy: Boosting

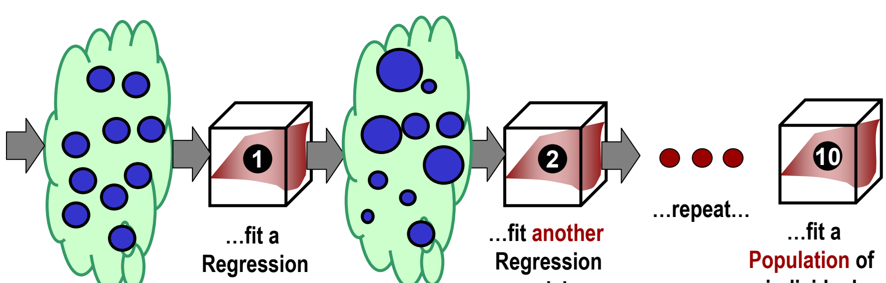
- Build a *sequence* of regression models, then *combine* them
  - Strategy for sequentially resampling/reweighting data, voting = **boosting**



© R.A. Rutenbar 2003 Slide 31

Ensemble Strategy: Boosting

- Build a *sequence* of regression models, then *combine* them
  - Strategy for sequentially resampling/reweighting data, voting = **boosting**



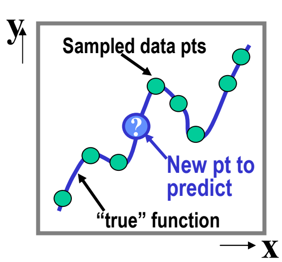
© R.A. Rutenbar 2003 Slide 32

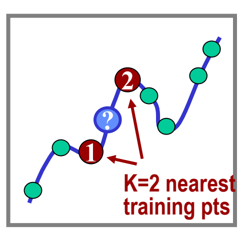
Boosting Specifics

- **Individual regressors**
  - Neural networks in MATLAB. 2-hidden layers, 10 neurons/layer
  - Chosen for ease-of-use mainly
- **Boosting mechanics**
  - Fit **N** networks, in stages, with resampling/reweighting [see our **DAC02** paper]
  - Well-fit data points can **vanish** from the population, over iterations
  - Poorly fit points can **replicate** in the population, over iterations
- **To combine regressors, more data mining: *instance methods***
  - Ask the **K nearest neighbor points** from training data what they think…

Instance-Based Voting of Regressors

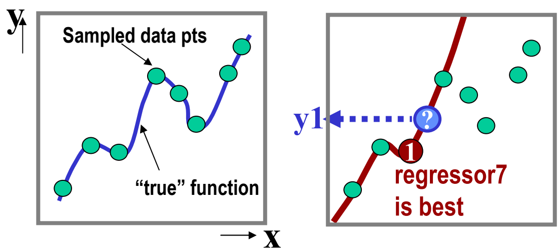
- **Suppose we set K=2 nearest neighbors (K is *arbitrary*)**
  - For any new data point, find **2** nearest training data points, and use the regressor that gave the most accurate prediction at that neighbor point





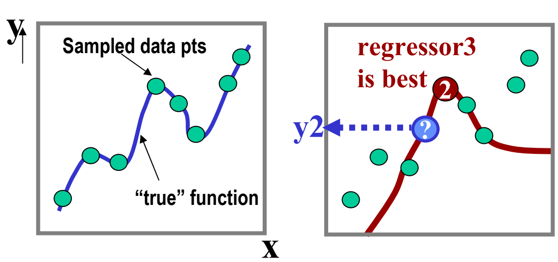
Instance-Based Voting of Regressors

- **Suppose we set K=2 nearest neighbors (K is *arbitrary*)**
  - For any new data point, find **2** nearest training data points, and use the regressor that gave the most accurate prediction at that neighbor point



Instance-Based Voting of Regressors

- **Suppose we set K=2 nearest neighbors (K is *arbitrary*)**
  - For any new data point, find **2** nearest training data points, and use the regressor that gave the most accurate prediction at that neighbor point



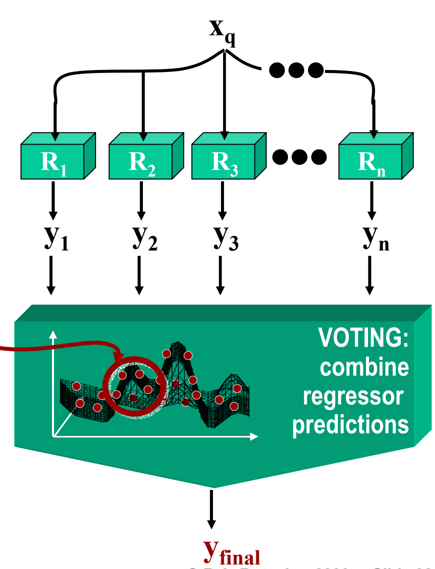
Average y1 and y2 values, weighted inversely by distance to new point

About Boosting

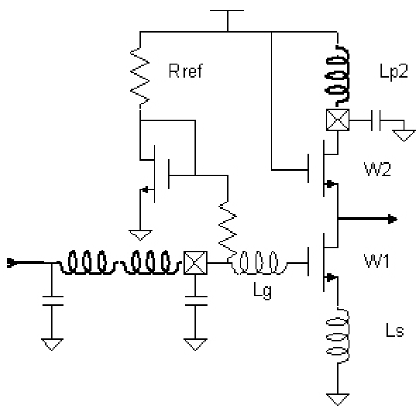
- *Independent* of the regressor (curve fitting) style
  - Use a neural net, a polynomial, posynomial, spline model, **whatever you like**
- *Many* regressors are fit; *few* are evaluated
  - Not uncommon in large problems to boost 100 rounds of regressors
  - But—our instance-based technique only evaluates the **best** regressor on each of the K neighbors **nearest** the evaluation point
- *Standard* data structures to get K nearest neighbors fast
  - High-dimensional nearest neighbor lookup is a **well studied** problem
  - Quad trees, kd-trees, R-trees, S-trees, flat vector scans, etc

What Overall "Boosted" Model Looks Like

- **Basic model setup**
  - ~10-20 boosted regressors $R_i(\,)$
  - Save training data points
- **Basic model eval**
  - Given a new "query point" x_q
  - Select a subset of stored data pts
  - Chooses *which* regressors to eval, *how* to combine (weighted vote)

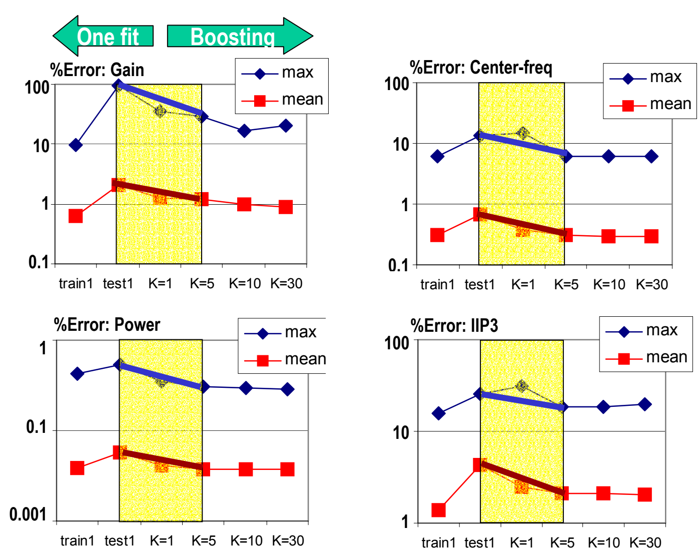


Simple RF Design Space Example

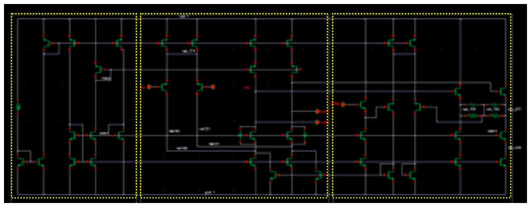


- **Simple RF LNA topology**
  - 5 independent variables
  - ~2000 HSPICE-simulated designs
  - Use half to **train** boosted predictors
  - Use other half to **test** predictors

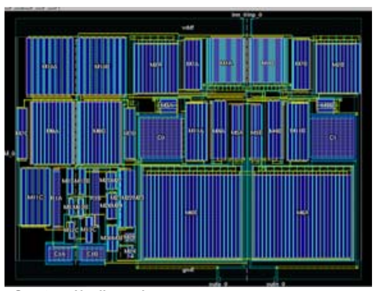
Simple RF Design Space Results



Example Synthesis Data Set



Gain: 70dB
UGF: 460Mhz
Power: 13.5mW
SlewRate: 360 V/us
SettlingTime: 4.75ns
PhaseMargin: 60°
Offset: 2.6uV



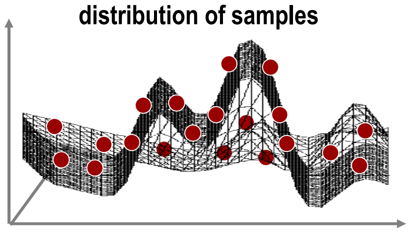
Courtesy Neolinear, Inc.

- **Industrial synthesis result**
  - CMOS amplifier ~50 devices
  - 27 design inputs (independent vars)
  - 12 outputs (performance specs)
- **We have…**
  - **~40,000, 27-dimensional data points**

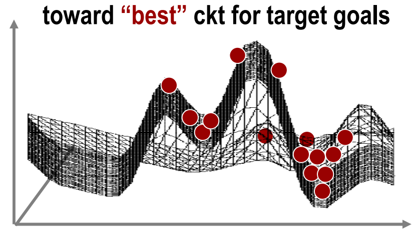
Data Points from Circuit Synthesis

- **New problems**
  - Synthesis visits many fully-qualified (simulated) samples of space…
  - …but synthesis may **not** sample in ways optimized for modeling
  - To illustrate this, we fit **first** and **last** **~20%** of time-ordered synthesis samples here; first samples have wider variance; last samples have less.

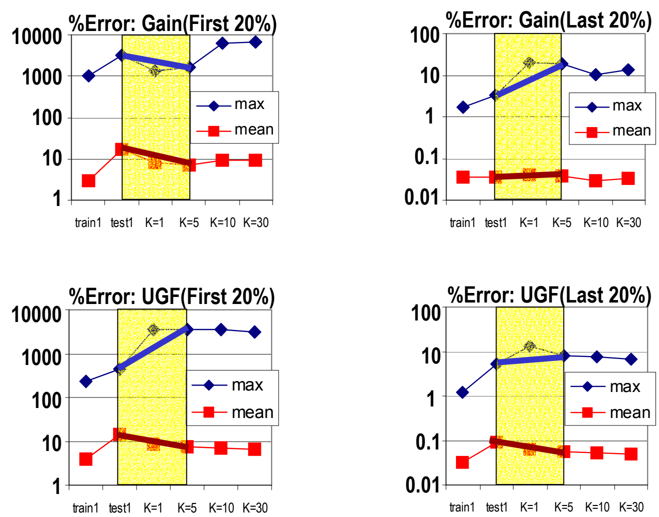
For modeling, want widest distribution of samples



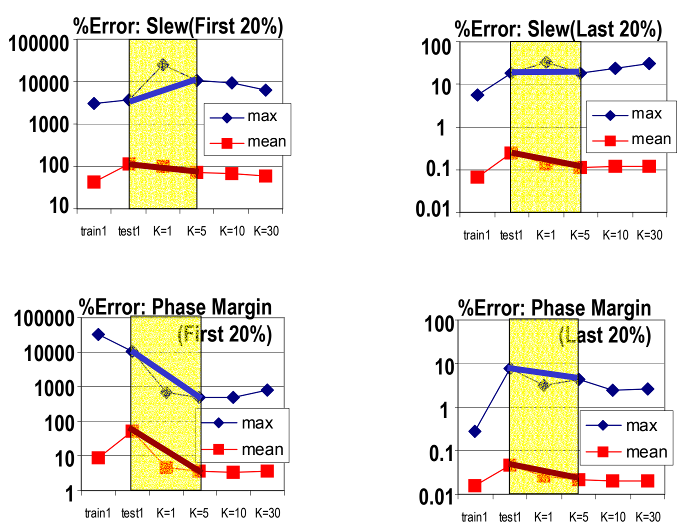
Synthesis focuses sampling toward "best" ckt for target goals



Boosted Results from Synthesis

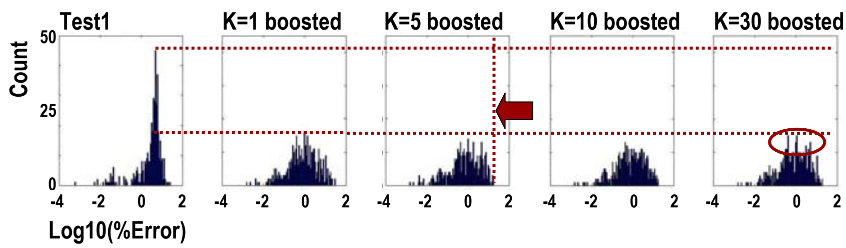


Boosted Results from Synthesis



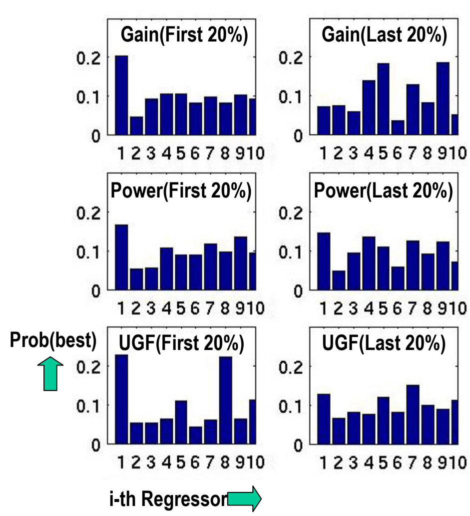
Qualitative Effects on Prediction Error

- Often, better worst-case error, and better mean error
  - Test1 is a single, conventional neural net regressor
  - K=1 to 30 show 10 boosted regressors, with up to 30 neighbors voted
  - K=small can help both max and mean; but you can also over-boost



Fraction of Points Choosing Each Regressor

- 10 regressors boosted.
- For training points, what fraction "believes" the i^{th} regressor to be best, locally?
- This is an informal measure of the utility of boosting to adapt an ensemble of models
- Note, not always true that 1st regressor dominate all others



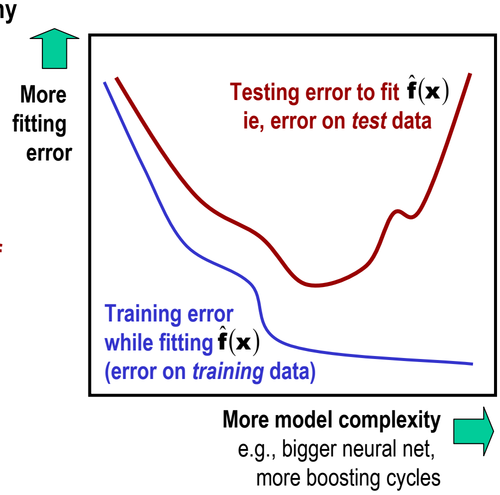
Issue: How Much "Model" is "Enough Model"?

- **Can always add complexity to any individual regression model**
- **Problem: Overfitting**
  - Easy to build more model than the data can really support
- **Classical Bias-Variance Tradeoff**
  - With more complexity…
  - Bias error **grows**:

$$\left[\mathbf{E}\left[\hat{\mathbf{f}}(\mathbf{x})\right] - \mathbf{E}\left[\mathbf{f}(\mathbf{x})\right]\right]^2$$

  - But variance error **shrinks**:

$$\mathbf{E}\left[\left(\hat{\mathbf{f}}(\mathbf{x}) - \mathbf{E}\left[\hat{\mathbf{f}}(\mathbf{x})\right]\right)^2\right]$$



Solution: Rigorous Model *Selection*

- **Generic model selection**
  - ***Divide*** the data into training set and test set.
  - ***Increase*** model complexity step by step
  - ***Fit*** each of these models
  - ***Choose*** the right model based on the test error
- **Can be applied to our macromodeling approach**
  - Base regressor selection: what is **best** single regressor ?
  - Boosting complexity selection: given right base regressor, how much boosting, etc., is **best**?

Solution: Rigorous Model *Selection*

- Generic model selection
  - *Divide* the data into training set and test set.
  - *Increase* model complexity step by step
  - *Fit* each of these models
  - *Choose* the right model based on the test error
- Can be applied to our macromodeling approach
  - Base regressor selection: what is **best** single regressor ?
  - **Boosting complexity selection: given right base regressor, how much boosting, etc., is best?**

How Much Boosting & How Many Neighbors?

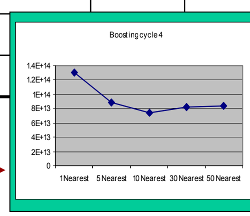
- **Experiment**
  - Run 2-16 boost cycles, and try 1-50 nearest neighbors to vote regressors
  - Pick the **best** model we see in this set of experiments

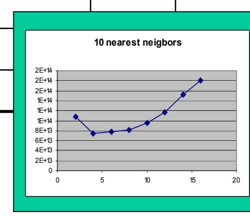
Model Selection	Boost 2 cycles	Boost 4 cycles	Boost 6 cycles	Boost 8 cycles	Boost 10 cycles	Boost 12 cycles	Boost 14 cycles	Boost 16 cycles
K=1 nearest neighbors								
K=5								
K=10								
K=30								
K=50								

Model Selection: Center Freq Model for RF LNA

- **Don't always need a lot of boosting/neighbors**
  - Mean Square Error **-18%;** Max error rate **-21%;** Mean error rate **~same**

Model Selection	Boost 2 cycles	Boost 4 cycles	Boost 6 cycles	Boost 8 cycles	Boost 10 cycles	Boost 12 cycles	Boost 14 cycles	Boost 16 cycles
K=1 nearest neighbors								
K=5		best						
K=10								
K=30								
K=50								



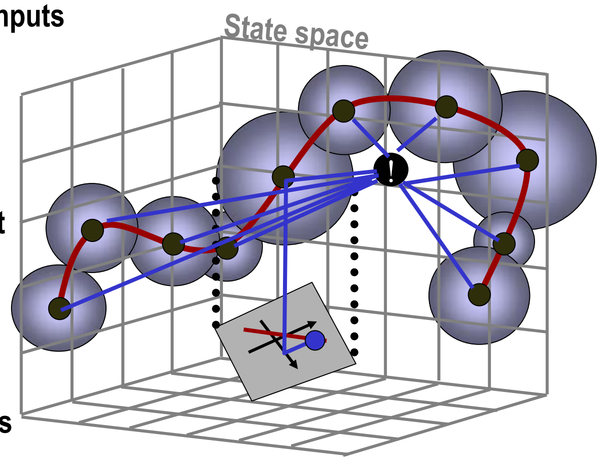


Observations

- **First rigorous application of data-mining in ckt modeling**
  - We can build usefully accurate predictive models of lots of high-dim data
  - Boosting is an elegant, robust fitting framework for exploration here
- **Rigorous model selection makes models easier to use**
  - Most users are not very familiar with regression jargon or parameters
  - Sensible application of selection / validation lets us auto-pick a 'best' model
- **Interesting connections to recent "simulation-directed" models**

Recall How Trajectory-Based Methods Work

- Simulate training inputs in state space
- Choose center pts on this trajectory
- Build simple local model near each pt projecting down to a smaller state vec
- Distance-weighted voting to compute predicted dynamics



Multiple regressors, locally weighted, with saved training data…

Summary

- **Macromodeling: Future looks pretty good (OK, *glorious*…)**
  - Because we don't have a choice, to do the systems we want to design
  - Because of many recent modeling innovations
- **Data mining: It's not just for dot-coms anymore**
  - Smart ways to handle large amounts of high-dimensional data
  - Smart ways to build and vote and validate ensembles of regressors
- **Simulation-based circuit optimization infrastructure**
  - Lots of companies making it possible to sample widely, cheaply, quickly
  - This will fundamentally change the way we get data to build macromodels