A Statistical Perspective on Nonlinear Model Reduction

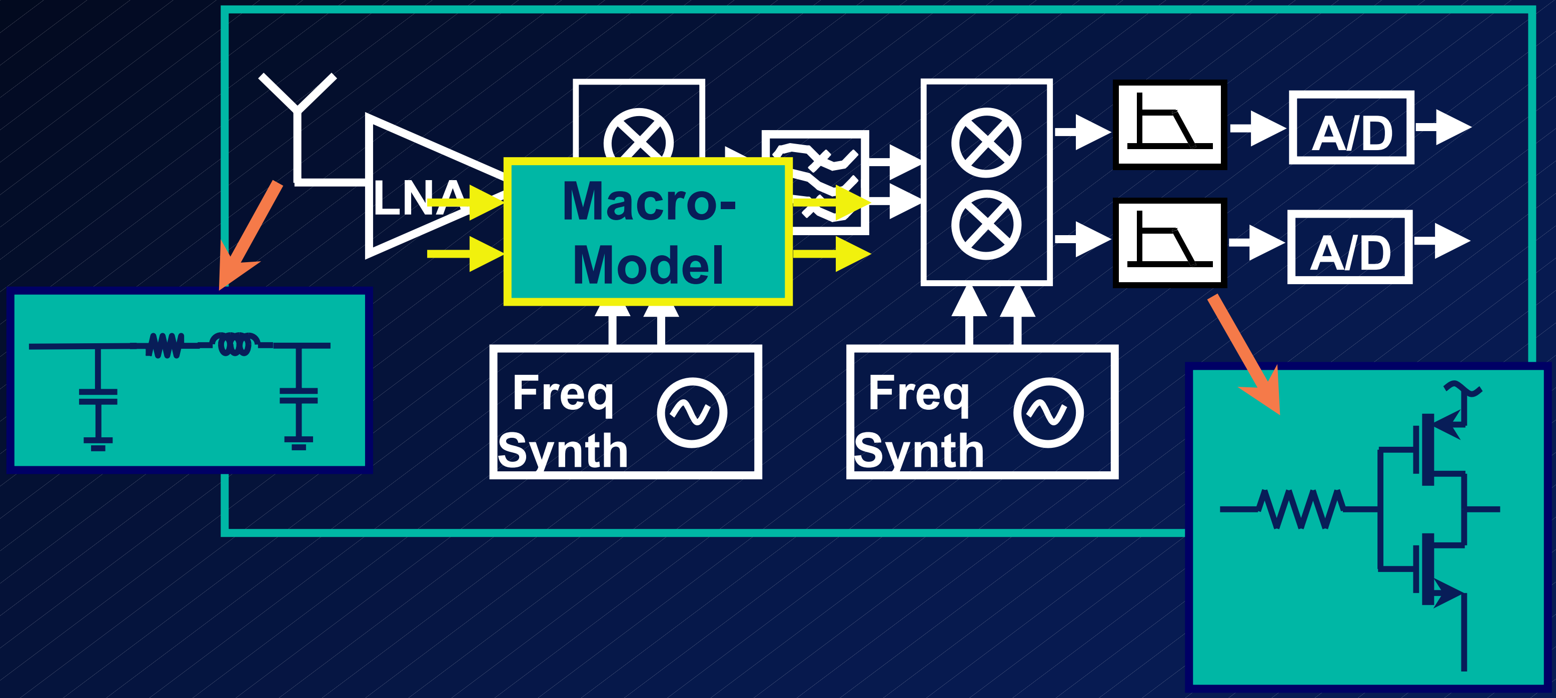
Joel Phillips, Cadence Berkeley Labs

Thanks to: Rutenbar et al @ CMU

cadence

Automated Analog Modeling

- Goal: Automatic algorithm that compresses transistor-level circuit descriptions into macromodels



Main Points

- Fundamental problem: prevent explosion of complexity due to dimensionality
- Statistically motivated thinking can be very powerful
  - Provides a (quantifiable) way to describe complexity
  - Nice way of formulating reasoning based on prior knowledge
    - Connection to regression-type methods (regularization to prevent over-fitting)
- (But most everything can be done from a deterministic viewpoint)

Dimensionality

- Linear systems

$$\frac{dx}{dt} = Ax$$

$O(n^2)$ coefficients
worse-case

- Nonlinear systems

$$\frac{dx}{dt} = f(x)$$

????? coefficients

  - "Nonlinear" model reduction is "non" – takes us into ill-described/unrestricted/undefined world. Need new ways of thinking about this

Linear System Feature Space

- Linear systems, one dimension: one coefficient

$$f(x) = ax$$

- Linear systems, N dimensions: N coefficients

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \end{bmatrix}$$

$$f(x) = Ax = \begin{bmatrix} \\ a_1 \\ \\ \end{bmatrix} x_1 + \begin{bmatrix} \\ a_2 \\ \\ \end{bmatrix} x_2 + \cdots$$

Nonlinear System Feature Space

- Nonlinear systems, one dimension: m>1 coefficients?

$$f(x) = ax + a_2 x^2 + a_3 x^3 + \cdots$$

- Nonlinear systems, N dimensions

$$f(x) = a_1 x_1 + a_2 x_2 + \cdots a_{11} x_1{}^2 + a_{21} x_1 x_2 + a_{31} x_1 x_3 + \cdots$$

$$+ a_{111} x_1{}^3 + a_{112} x_1{}^2 x_2 + a_{123} x_1 x_2 x_3 + \cdots$$

  - N coefficients?
  - N^m coefficients?

Recall: Volterra like methods

$$\frac{dz}{dt} = \hat{A}_{(1)} z^{(1)} + \hat{A}_{(2)} z^{(2)} + \hat{A}_{(3)} z^{(3)} + \ldots + Bu$$

$$\hat{A}_{(1)} = V^T A_{(1)} V, \qquad \hat{A}_{(2)} = V^T A_{(2)} (V \otimes V),$$

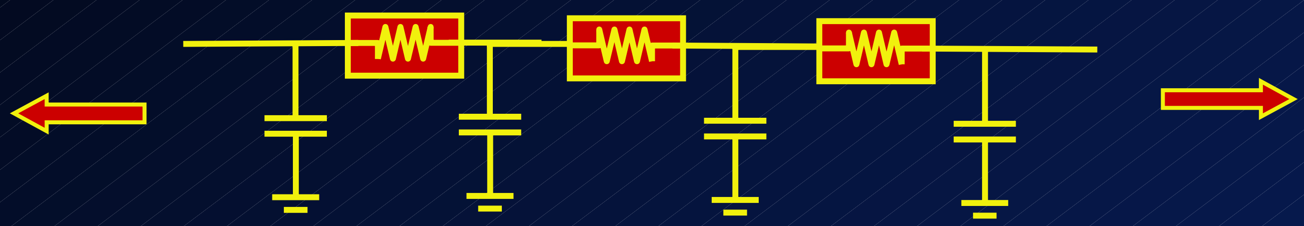
$$\hat{A}_{(3)} = V^T A_{(3)} (V \otimes V \otimes V), \qquad \text{etc.}$$

- Problem: Tensors of $O(m)$ contain $O(q^m)$ elements in a q-state model

- Is this the right way to measure "complexity"?

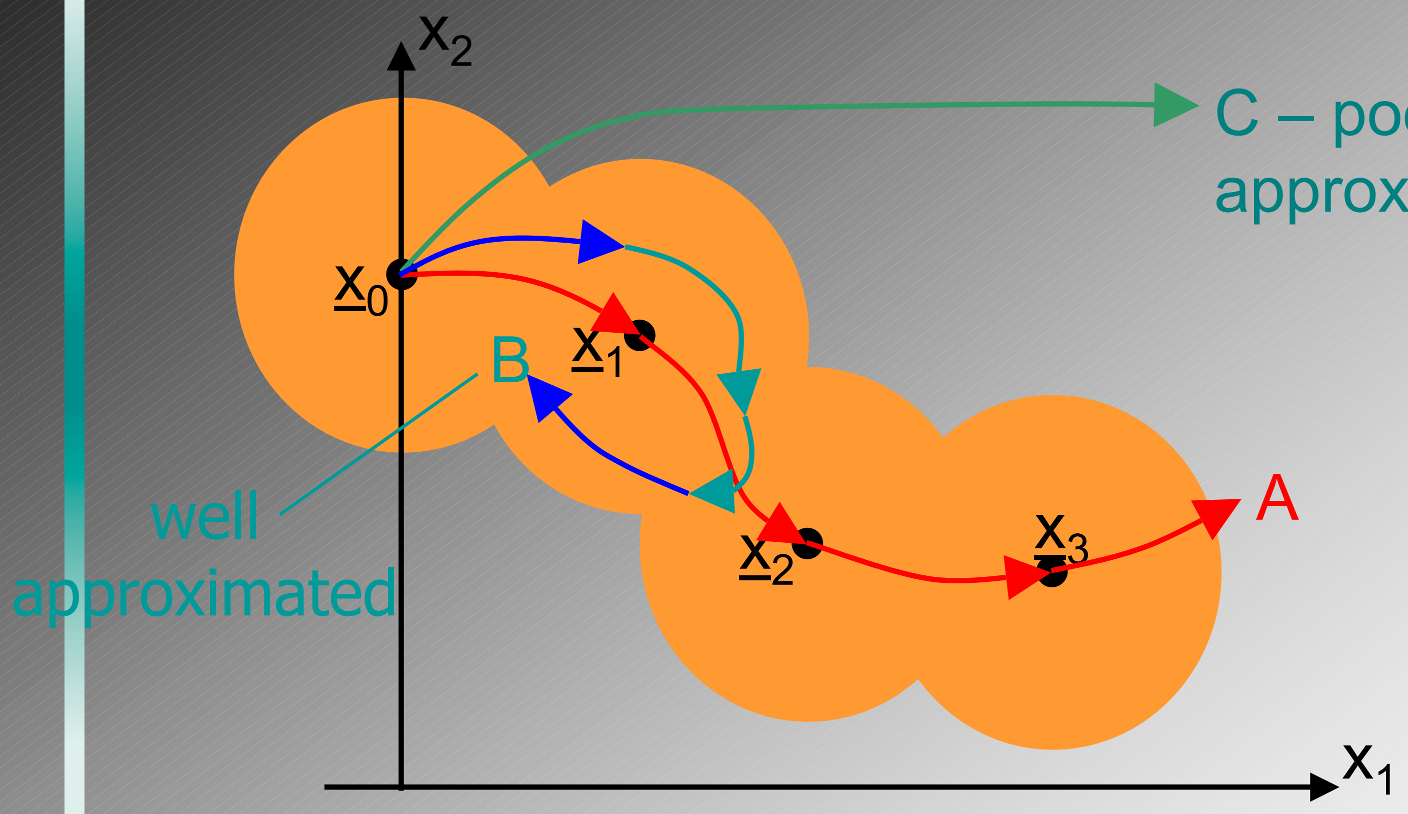
Odd Observation #1: Coefficient Explosion

- Model system: thirty node circuit with two-terminal nonlinearities ("nonlinear delay line")



- Consider approximating each nonlinearity with order-10 polynomial
  - → 300 coefficients to describe
- After "reduction" to 5 state variables (6X reduction in state space size)
  - → 50 million coefficients to describe in tensor product form
  - → 5000 coefficients to describe in non-redundant form
- Oops!

Odd Observation #2: Trajectory Methods



- Why does this work (at all)? Any hope it will work in general?

So how hard, really?

- Goal: quantitatively describe amount of "redundancy" in a nonlinear model

$$\frac{dx}{dt} = f(x) + Bu \longrightarrow \frac{dz}{dt} = \hat{f}(z) + \hat{B}u(t)$$

- When is reduction possible?
  - Only in special cases? Which cases?
  - What information is needed for "good" reduction?
- What is the trade-off between model size and error?

Information Theory

- Consider a random variable X with probability density p(x)
- Entropy:
  - $H(X) = -E_p\{\log p\} = \sum p \log p$
- Interpretation
  - Given a sequence X_1 X_2 X_3 ……..
  - Lower bound of average symbol length of code for sequence

Shannon Coding

- Basic idea – assign codelengths proportional to
  - High probability symbols get short codes
  - Low probability symbols use longer codes

$$l(x) \sim \log \frac{1}{p(x)}$$

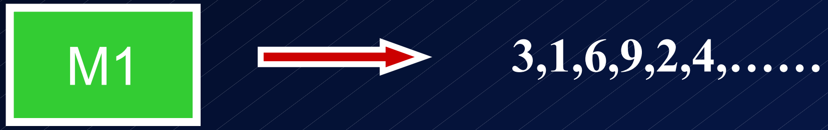
p(x)	Code
1/2	0
1/4	10
1/8	110
1/8	111

- Asymptotically achieves optimal code length (~entropy)

$$E_p\{L(X)\} = \sum p(x) l(x) = \sum p(x) \log \frac{1}{p(x)} < H(X) + 1$$

Example: A Box of Integers

- Simple Probabilistic Model: Box emits a number “between one and ten”



- Over all boxes there is a box with maximal entropy → uniform probability distribution
  - Entropy H(M1) = log 10
  - In the worse case a code can be constructed with average symbol length log 10

Prior Knowledge

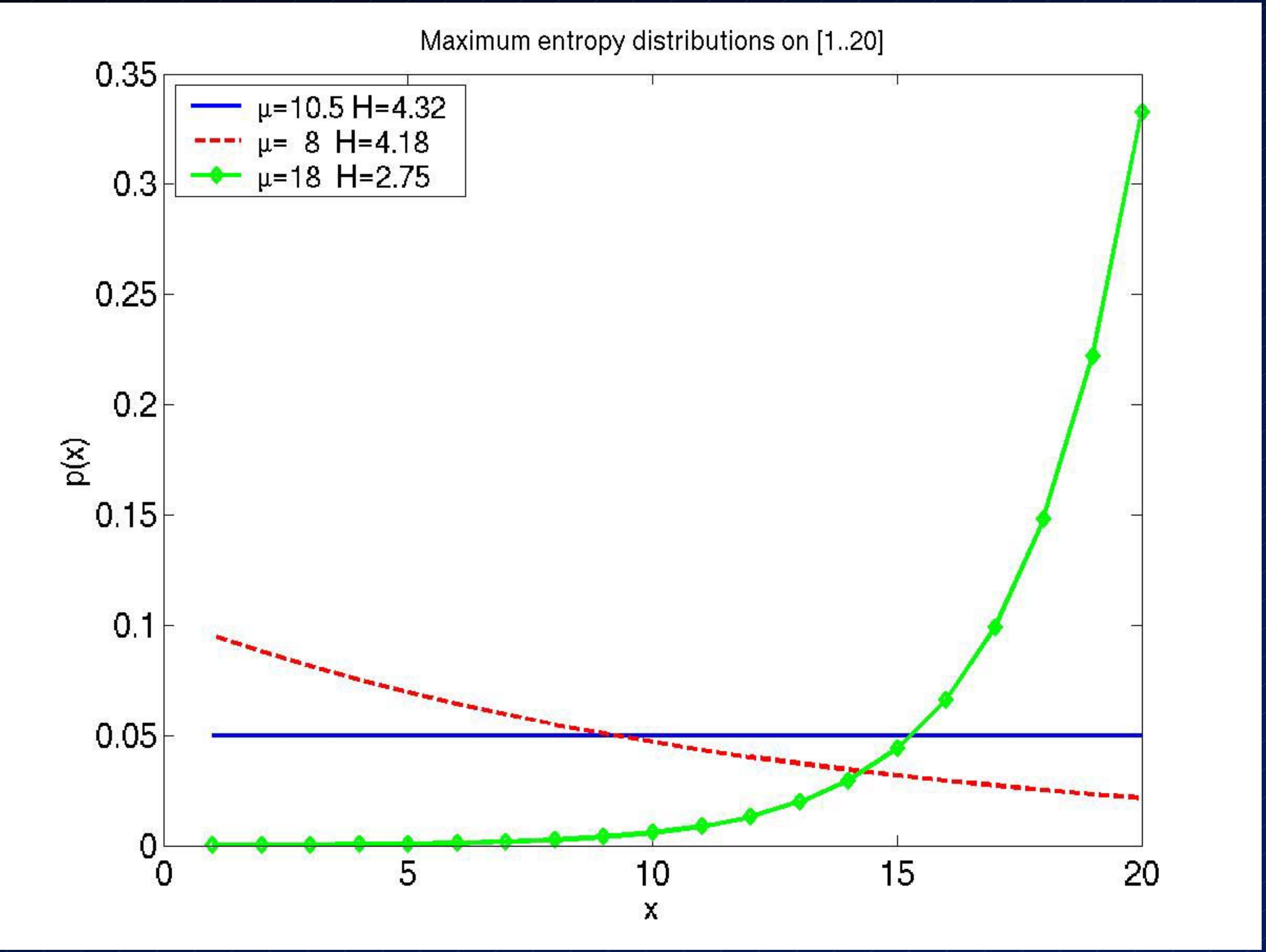
- Conditional entropy
  - $H(X|Y) = E\{ \log p(X|Y)\}$
- Conditioning decreases entropy
  - $H(X|Y) <= H(X)$
    - Equality only if X,Y independent (Y is non-informative wrt X)
- Prior knowledge makes representation easier

Box Example, Continued

- Simple Probabilistic Model: Box emits a number "between one and ten"

- Over all boxes with specified mean there is a box with maximal entropy
  - Entropy H(M2) < log 10
- Additional conditions – e.g. prior knowledge – lowers the entropy → easier to represent ←→ shorter code

Box Example, Continued



What (pray tell) has this to do with dynamic systems modeling?

Complexity of Model Reduction

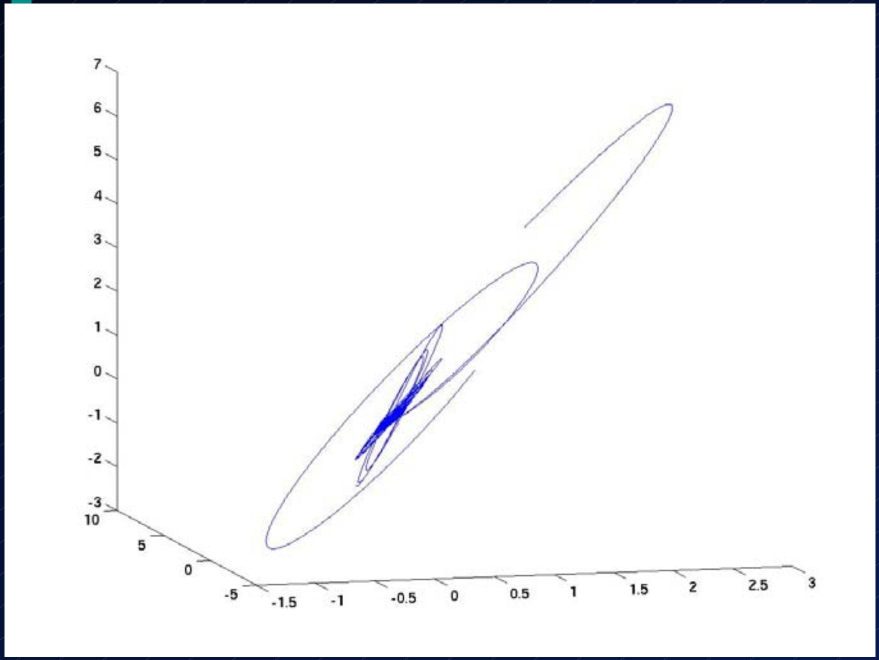
- Recall: model reduction challenge

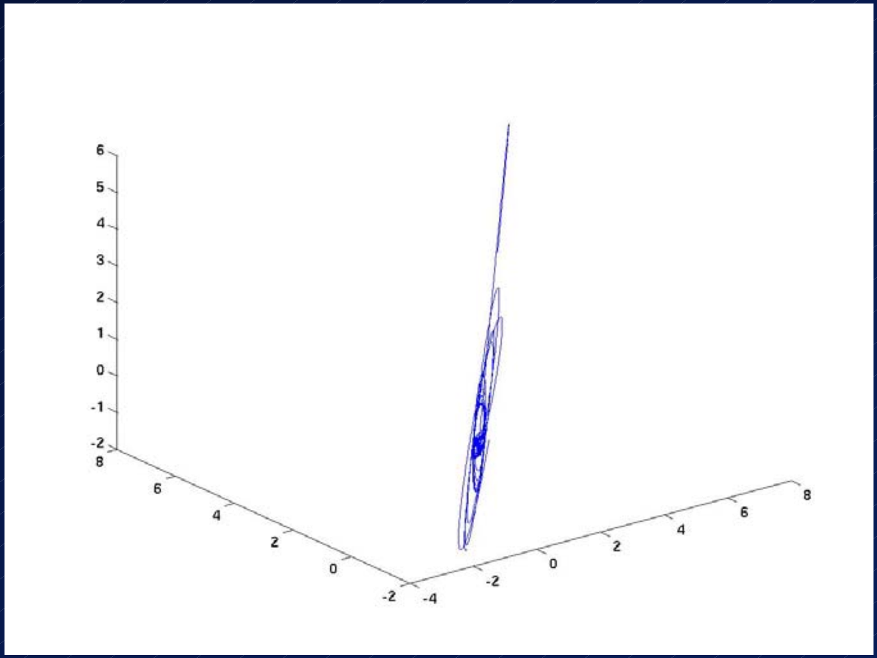
$$\frac{dx}{dt} = f(x) + Bu \longrightarrow \frac{dz}{dt} = \hat{f}(z) + \hat{B}u(t)$$

- Q1: Complexity of representing state.
  - How much “information” is present in the state space?
  - Relates to construction of z
- Q2: Complexity of vector field f(x)
  - How much “information” is present in the state → derivative (feature-space) mapping?
  - Relates to construction of f(z)

Compacting the State Space

- Not all portions of the state space are accessed with high probability → compact representation exists





Example: Statistical Perspective on TBR

- Consider

$$\frac{dx}{dt} = Ax + Bu \qquad y = Cx$$

- Define operator L: u → x(0) maps past inputs to state
- TBR: L_2 optimal approximation of L

Statistical Perspective on TBR, contd

- Grammian: eigenvectors give principle components of L

$$X_c = LL^H = \int_{-\infty}^{0} e^{At} BB^T e^{A^T t} dt$$

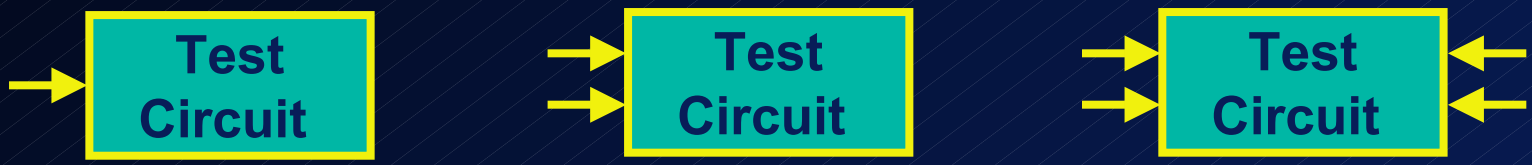
- Statistical interpretation
  - Unit power, white spectrum Gaussian process → covariance matrix of zero-mean Gaussian process

Statistical Perspective on TBR, contd

- Eigenvalues of Grammian
  - = singular values of operator L
  - = variances of N-dimensional Gaussian process
- Entropy = sum of log of singular values
- Small SVs ←→ easy to approximate ←→ low entropy
- Lower entropy means:
  - For a given model order, lower error
  - For a given error, a lower model order is needed

Statistical Perspective on TBR, contd

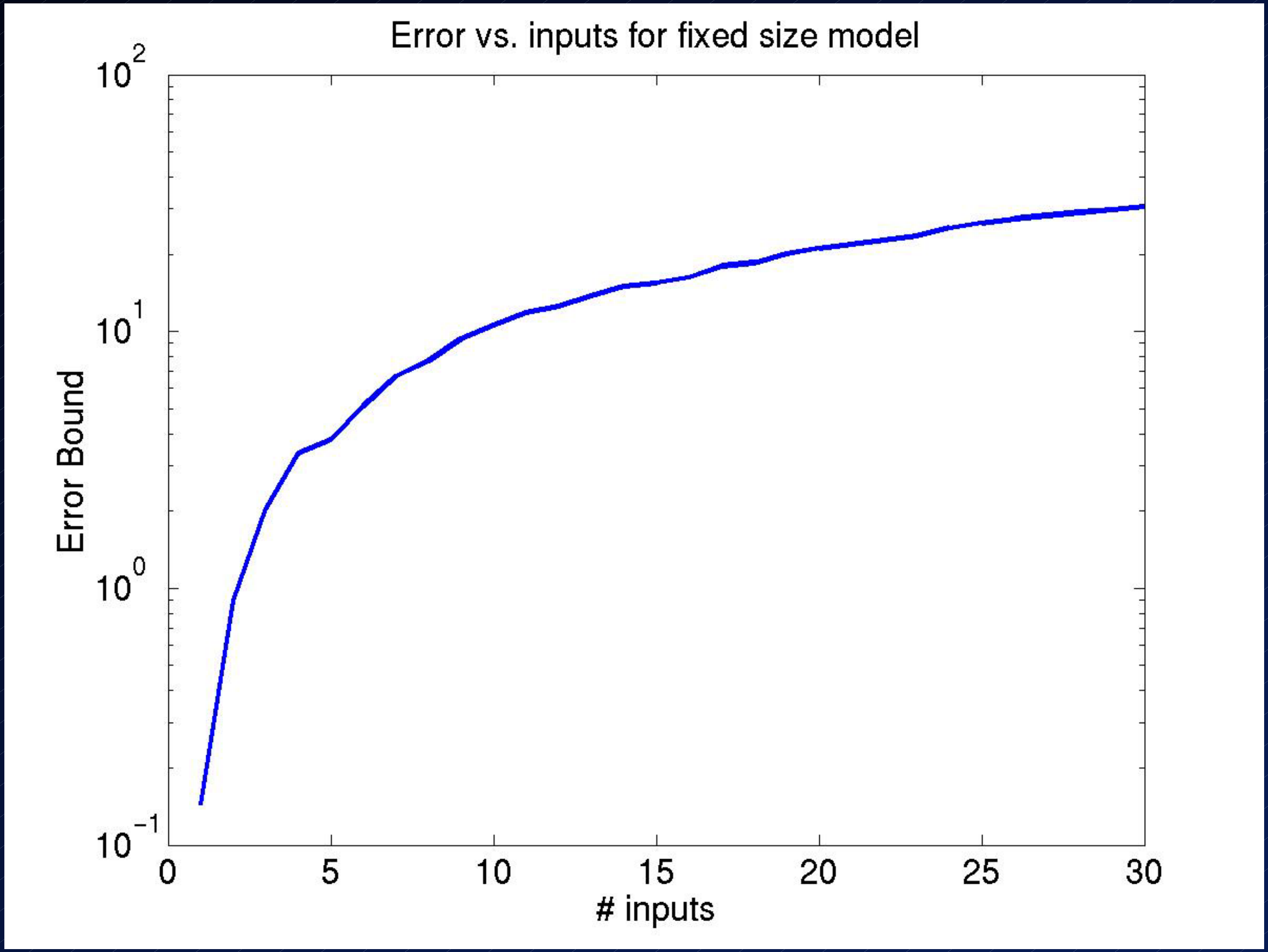
- Input Restriction: What is the impact on entropy *given that the circuit is only driven at certain selected points?*



$$H_1 < H_2 < H_4$$

- The fewer the inputs, the lower the entropy
→ the easier to approximate with low order models
  - Quantifiable
  - Agrees with our intuition
  - Statistical & classical interpretation agree

Statistical Perspective on TBR, contd



What can we say about nonlinear modeling?

The Charge/Current Functions

- Recall: the nonlinear functions may “live” in exponentially large spaces
  - BUT – it may be that much of the space may be accessible only with low probability → May be enough to utilize some subset of the possible functions
- Why would this occur? How can it be exploited?

$$f(x) = a_1 x_1 + a_2 x_2 + \cdots a_{11} x_1^{\,2} + a_{21} x_1 x_2 + a_{31} x_1 x_3 + \cdots$$
$$+ a_{111} x_1^{\,3} + a_{112} x_1^{\,2} x_2 + a_{123} x_1 x_2 x_3 + \cdots$$

$$\Downarrow$$

$$\hat{f}(x) = a_r (x_2 + x_1 x_2 x_3)$$

Need an Implicit Representation!

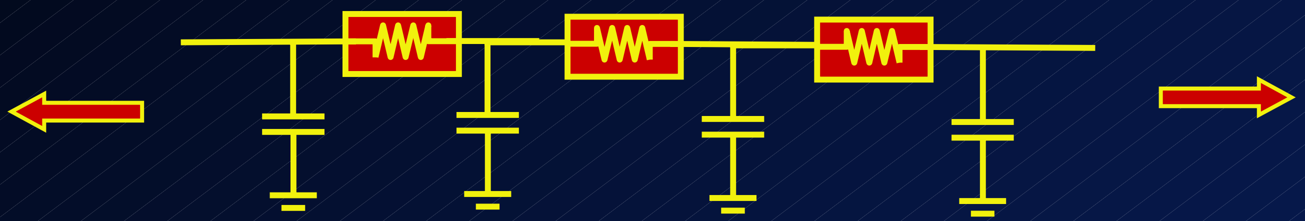
- Need the ability to work in an exponentially large space – but only use function components specifically needed in the problem at hand ("Pay as you go")
- Example: Kernel Hilbert Spaces (Informal Description)
  - Idea: Use a "kernel" K(x,y) as a function space generator

$$f(x) = \sum_k c_k K(x_k, x)$$

  - Each x_k selects some "particular" basis function $K(x_k, x)$
  - Space of all $K(x_k, x)$ is the RKHS

RKHS Example: “Diode Line”

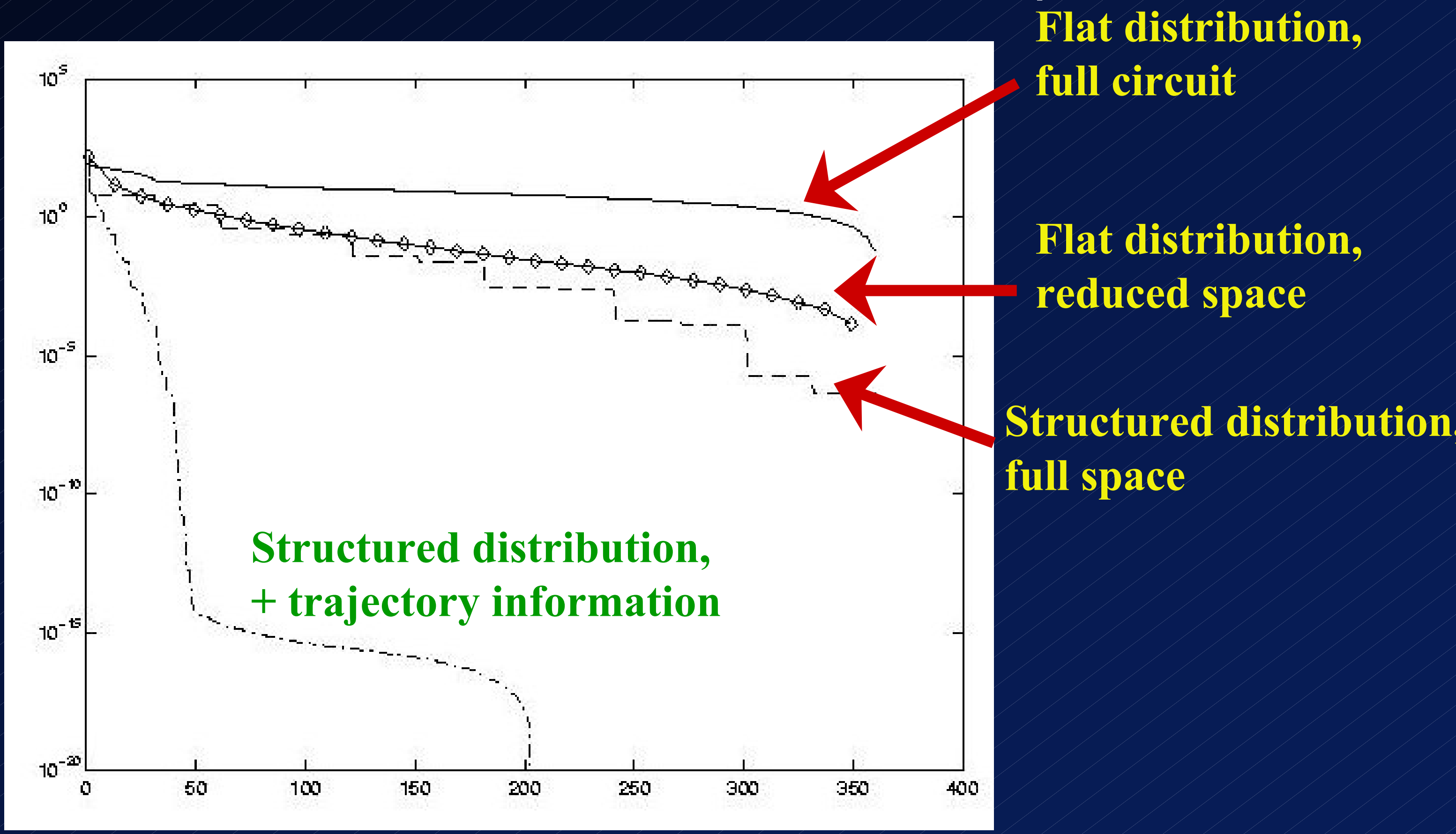
- Space of all order 10 polynomials is of dimension 30^{10}



- Polynomial kernel
  - K(x,y) = (1 + <x,y>)^d
  - Only need M=300 x_k (“support vectors”) to describe the line (30 * 10 = 300)
  - Number does not grow with projection to reduced model

The more prior knowledge, the better

- Continuing diode example…..now drive at one end with ensemble of waveforms, do SVD in feature space:



Interpreting Classical Volterra Techniques

$$\frac{dz}{dt} = \hat{A}_{(1)} z^{(1)} + \hat{A}_{(2)} z^{(2)} + \hat{A}_{(3)} z^{(3)} + \ldots + Bu$$

$$\hat{A}_{(1)} = V^T A_{(1)} V, \qquad \hat{A}_{(2)} = V^T A_{(2)} (V \otimes V),$$

$$\hat{A}_{(3)} = V^T A_{(3)} (V \otimes V \otimes V),$$ etc.

- Explicit enumeration of basis functions
  - "Equal cost" associated with each basis function
  - Each is equally important
- Interpretation: no prior knowledge → flat statistical distribution → maximally sized model
  - Problem with Volterra is mechanical, not intrinsic

Explicit Techniques Are Bad?

- Once N^m functions are written down….game is over
- Moment matching is probably not a good metric in higher dimensions

Is Everything Reducible?

- Information that can reduce “circuit entropy”
  - Device properties (2 vs. 3 vs. 4 –terminal, IV curve shapes)
  - Constraints due to connectivity
    - E.g.: voltage decreases along line when driven at one end
  - Reduced set of inputs
  - Finite bandwidth inputs
- Always occurs in practical circuits
- Implies: Models are always reducible (in some sense) without loss of accuracy

Trajectories, Sensitivity, Generalization

- Conjecture: Exploiting low entropy state & feature spaces
  - Small entropy ←→ small "volume"
  - Not sensitive to samples → have covered dominant portion of volume
  - Implies good generalization error
    - Methods should work well for all other inputs in the probability class
- *Not* due to piecewise nature of representation

Summary: Case for Statistical Thinking

- Quantifiable way to describe “compressibility” of analog circuit models
- Quantifiable way of determining amount of “compressibility” introduced by structure, restrictive assumptions, restrictive inputs.
  - → Connection to “regression” viewpoint
    - Structural information introduces extremely strong constraints into the circuit modeling problem…“black-box” techniques do not work nearly as well!
    - “Overfitting” tamed by regularization – circuit’s own internal behavior/structure is the best regularizer