BLACK-BOX MODELLING

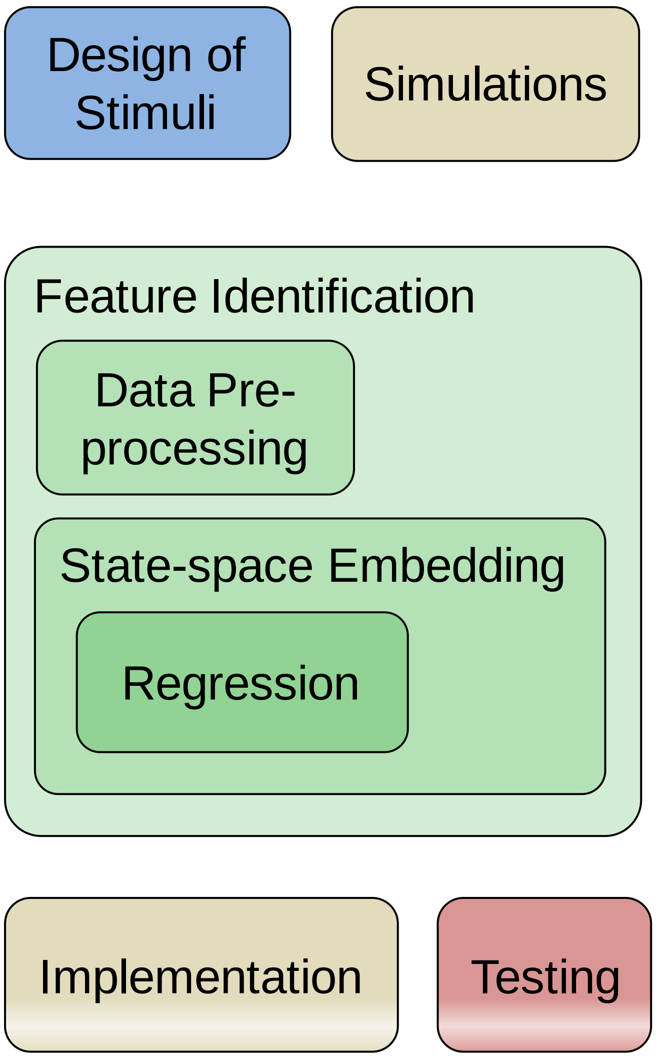
Response surface modelling

V. Ceperic and A. Baric
University of Zagreb

19-Apr-13, Brussels, Belgium

Outline

What's in Black-box Modelling



- Sampling of the input-output behaviour
  - Cover all time-constants / frequencies
  - Cover all relevant amplitudes
  - Cover all meaning input combinations
- Select the training / testing data-sets
- Generate an extensive set of possible input variables (loop to reduce)
- Select the regression engine
- Select the engine's parameters
- Run the regression (model extraction)
- Validate the model

Some Regression Engines

Artificial Neural Networks

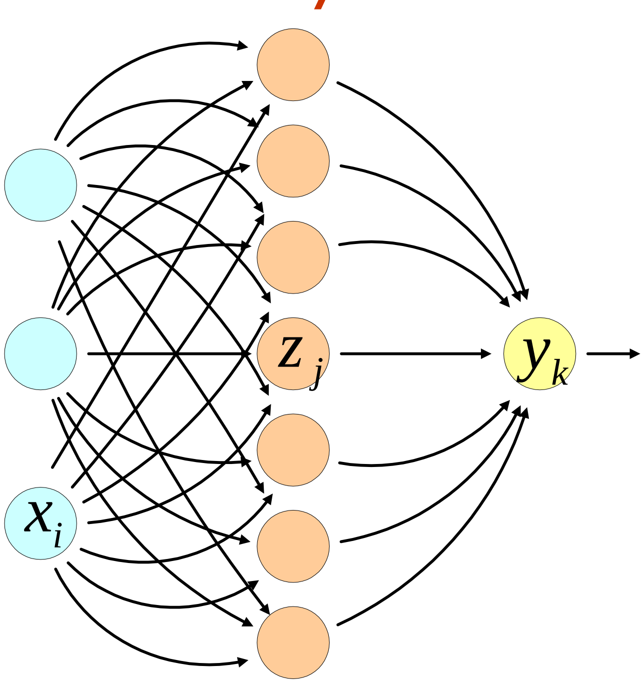
- Historically one of the first tools of machine learning
- Many publications of ANN-based models in electronics
- Known issues :
  - Sensitivity to initial condition
  - Difficult to estimate optimal parameters (eg. # layers)
  - Prone to over-fitting
  - Stability issues when feedback

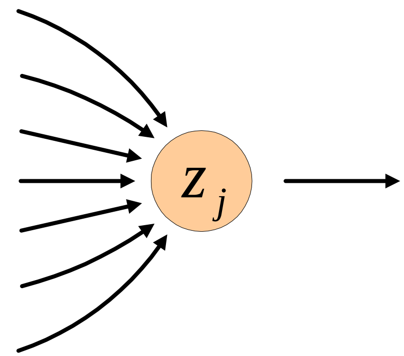
Support-Vector Machines

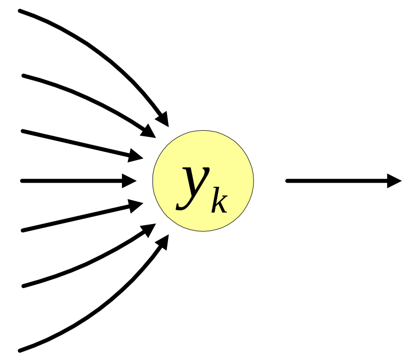
- Developed to overcome limitations of ANN's
- Currently behind many hand-writing recognition and data-base mining algorithms

Artificial Neural Networks

- Weighted sum of inputs
- Every neuron applies a "non-linear" threshold function
- Learning based on concept of back-propagation of errors
- Multi-layer PERCEPTRON :







$$a_j = \theta_j + \sum_i w_{ji} \cdot x_i$$

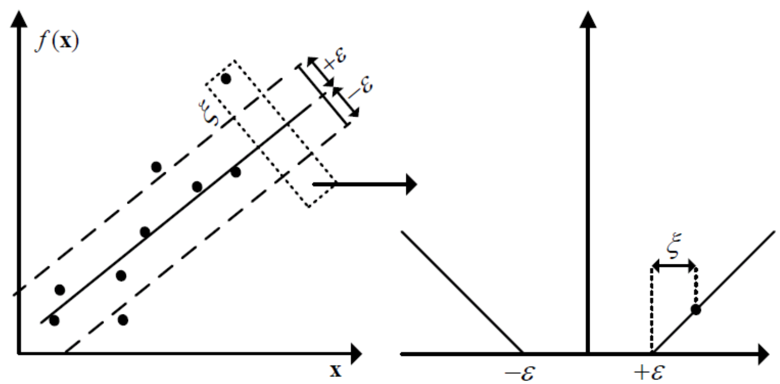
$$y_k = \theta_k + \sum_j w_{kj} \cdot z_j$$

$$z_j = \frac{1}{1+\exp\left(-a_j\right)}$$

Support-Vector Machines

- SVM's realize a kind of "lean regression"
  - Insensitivity tube of width ε
  - Model tends to minimize ξ
  - Model favors lower weights

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b,$$



$$\text{minimise } \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)$$

$$\text{subject to } \begin{cases} y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b & \leqslant & \varepsilon + \xi \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b - y & \leqslant & \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* & \geqslant & 0 \end{cases} .$$

$$y(x) = \sum_{k=1}^{N} \alpha_k \boxed{K(x_k, x)} + b \text{ with unknowns } \alpha \text{ and } b.$$

$$\left(w = \sum_{k=1}^{N} \alpha_k \varphi(x_k)\right)$$

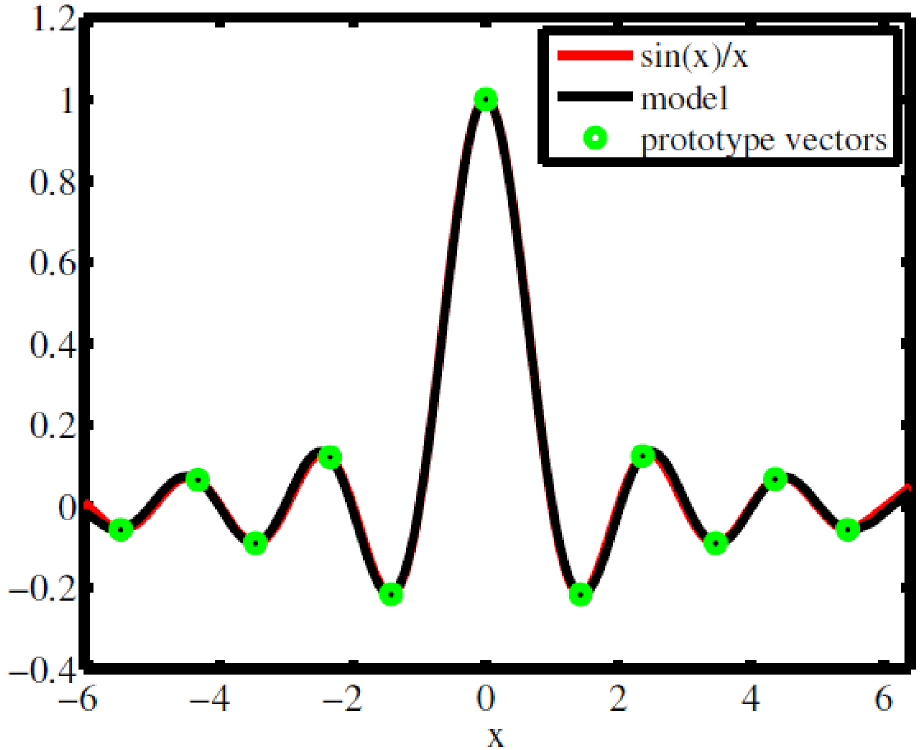
Kernel function enabling high-dimensional nonlinearities

Support-Vector Machines

- Very popular in machine learning :
  - Excellent performance in a variety of learning problems;
  - Theoretical guarantees about their performance;
  - Lower susceptibility to local minima;
  - Deals well with highly dimensional input data.
- However, machine-learning practitioners seem to have an unlimited budget for computational power
  - Usually extraction is run targeting minimal error
  - A large model results (many SV's) which is slow to evaluate
  - Some reduction algorithms were proposed to discard SV's
- Efficient analogue abstractions still require better…

Support-Vector Machine

- Constructive « active learning » :
  Only add support-vectors where absolutely necessary
  (*allows to generate compact models with good accuracy*)



Importance of Data Post-processing

Pre-processing technique	Test data set NRMSE improvement
Scaling (from [-1 to 1])	34.5%
White noise	3.5%
Moving average filters (instead of time delays)	7.1%

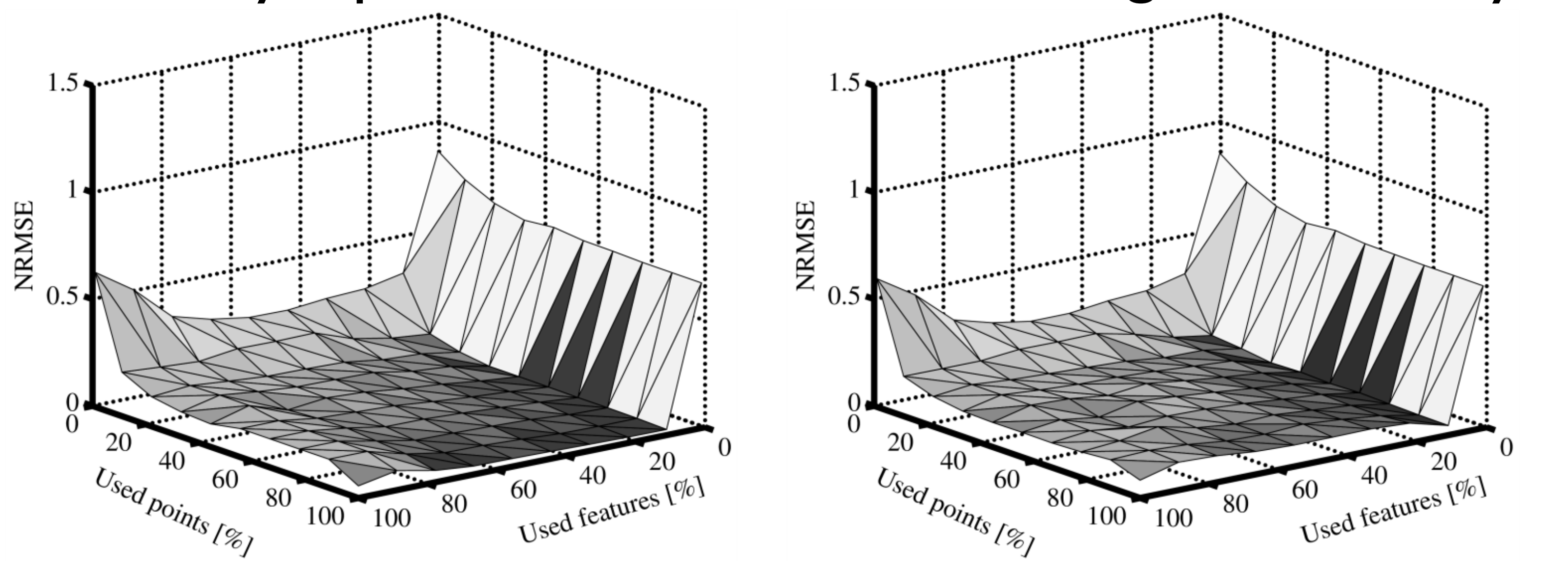
- Scaling / normalizing the data helps both for ANN & SVM's
  - Otherwise there is a risk that significant errors between low-amplitude data-points are just neglected
- Addition of white noise :
  - improves robustness, stability and generalisation ability (Jaeger '02)
  - Avoids the risk that the extraction « focusses » on one input
  - In case of feedback loop, renders the model robust against numerical issues

Normalised root mean square error

- NRMSE = *sqrt*(*mean*((t-y).^2)/*var*(t));
  - *sqrt* = square root, *mean* = mean value, *var* = variance
  - *t* = *target*
  - *y* = *model prediction*
  - 0 - perfect match

Feature Selection Algorithm

- How many inputs do we need to achieve good accuracy ?



(a) SVM model error on the validation data (b) SVM model error on the testing data set

- A sufficient number, but also not too many !
- Good correlation between training and validation data-sets
  - Select features based on other data-set than training set usually improves the model quality

Feature Selection Algorithm

- Feature selection also known as subset selection or variable selection (Guyon, 2006)
  - Wrapper methods
  - Feature filtering
  - Embedded techniques

- Training point selection methods are also actively investigated, e.g.
  - (Wang, 2005) presents training data selection for support vector machines
  - (Patan, 2010) presents selection of training data for locally recurrent ANNs
  - Active learning methods can also be used for training point selection

- Zhang et al. in the recent paper (Zhang, 2012) advocate to do the feature selection and training point selection in the same time.
  - Feature and Training point Selection and Ranking = FTSR

Feature Selection Algorithm

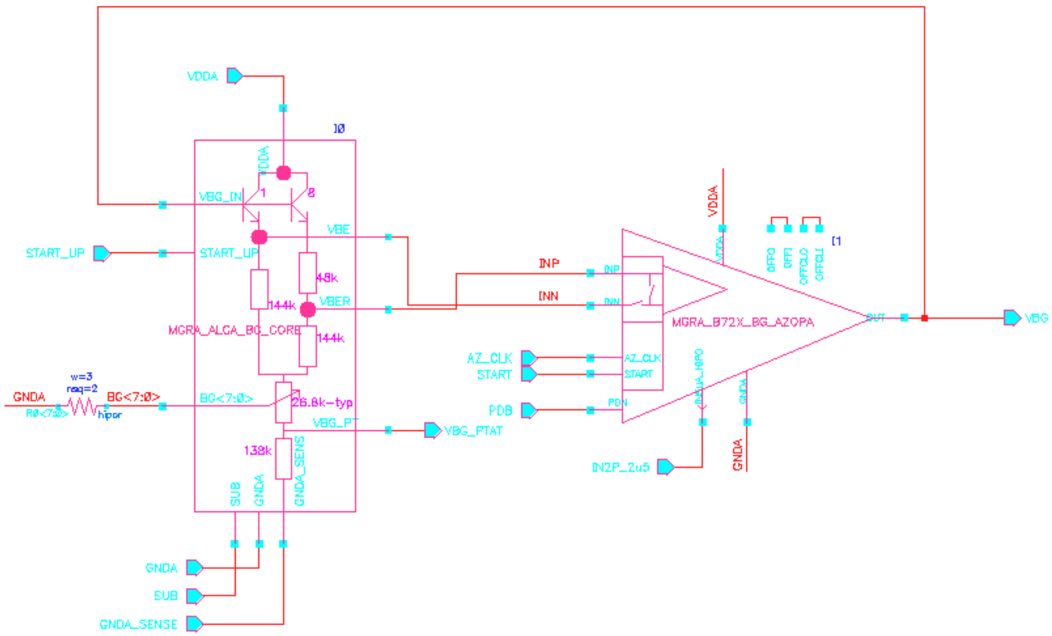
- You can't beat a feature selection algorithm…

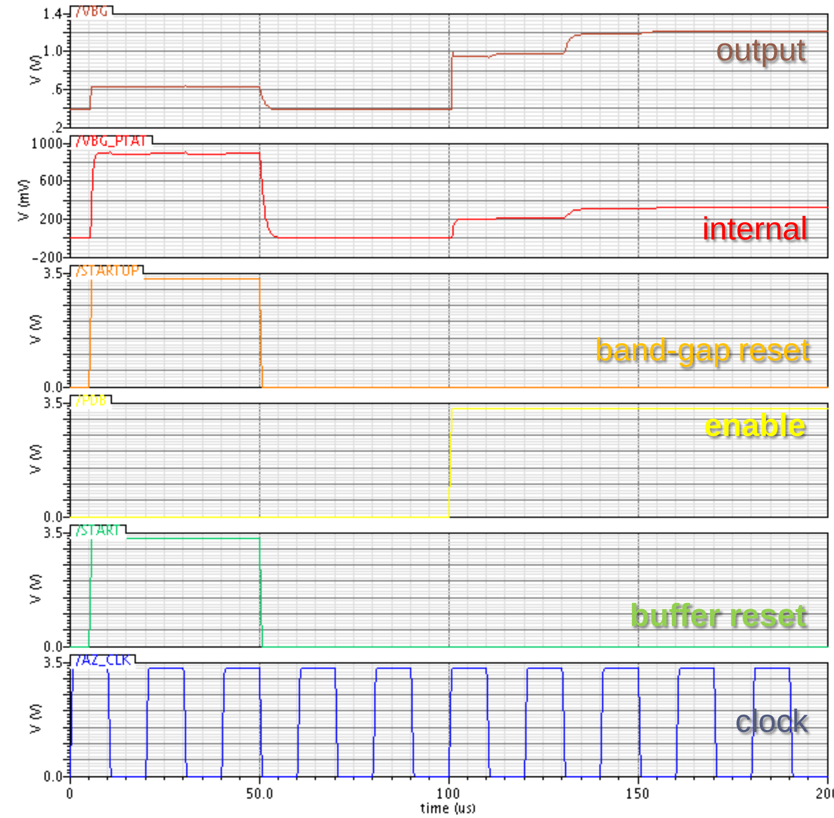
	FTSR	**"Expert" (manual selection)**	**GA**	**PSO**
NRMSE:	0.045	0.18	0.11	0.15

  - FTSR : feature selection
  - GA : genetic algorithm
  - PSO : particle swarm optimization

- Use feature and data-point selection methods !
- FTSR is key to fully automate the model generation

Auto-zero Bandgap Test-case

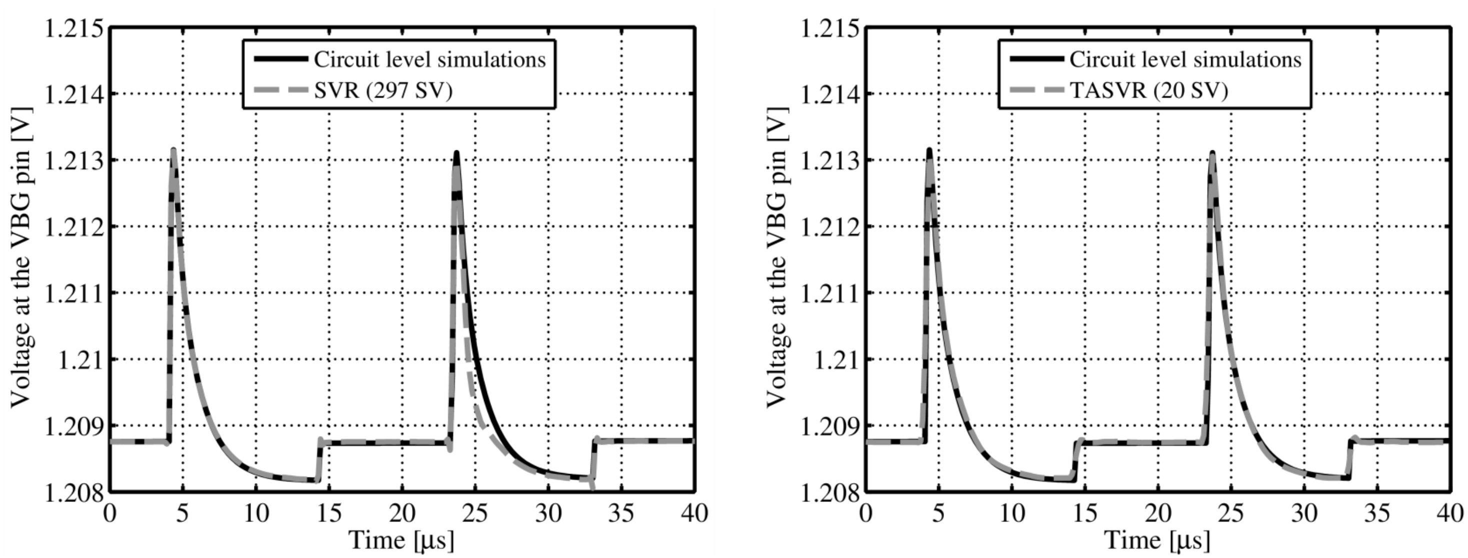




- Stable voltage reference circuit
- Followed by a buffer (to isolate from the load)
- Technology imperfections in the buffer spoil the reference
- Circuit has an auto-calibration routine (internal states)

Band-gap Test-case

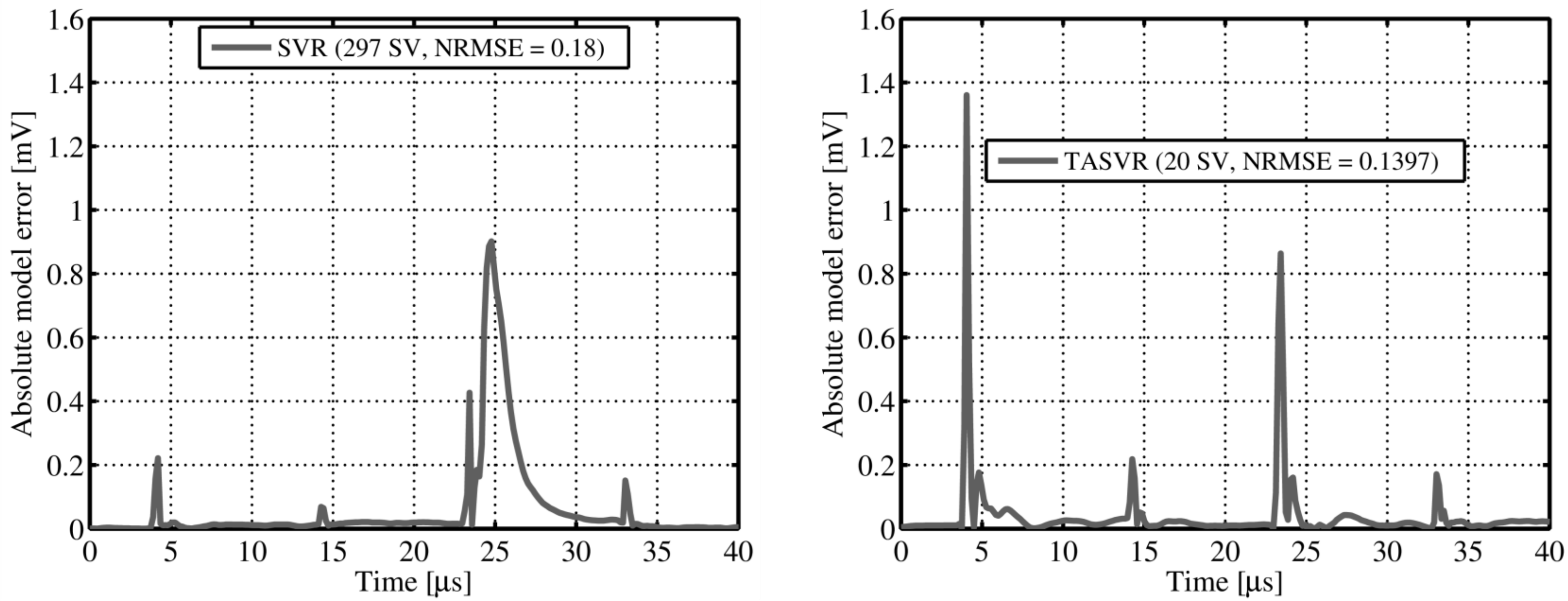
- Voltage bandgap reference with offset compensation
  - ONSEMI test-case with 242 transistors
  - modelling of the VBG voltage in the time-domain
- Comparison of standard SVR and new TASVR



(a) SVR **(297 SVs)** vs. circuit level simulations (b) TASVR **(20 SVs)** vs. circuit level simulations

Band-gap Test-case (continued)

- Comparison of standard SVR and new TASVR : <u>Absolute model Errors</u>



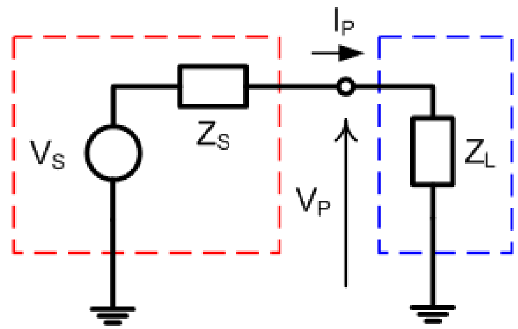
(a) SVR **(297 SVs)** vs. circuit level simulations

(b) TASVR **(20 SVs)** vs. circuit level simulations

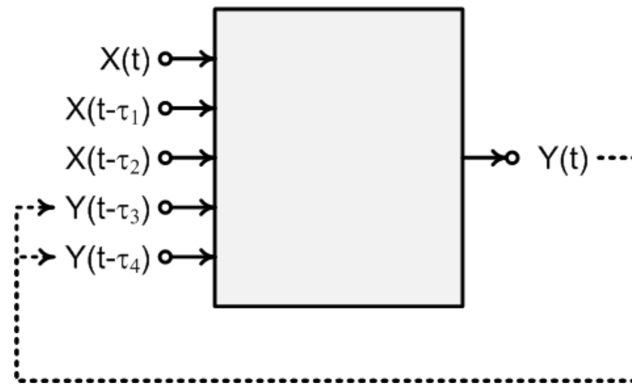
- Same level of errors but TASVR >10x faster

The Problem of Ubiquitous Feedback

- Analogue circuits ≡ feedback (*everywhere you don't want it*)



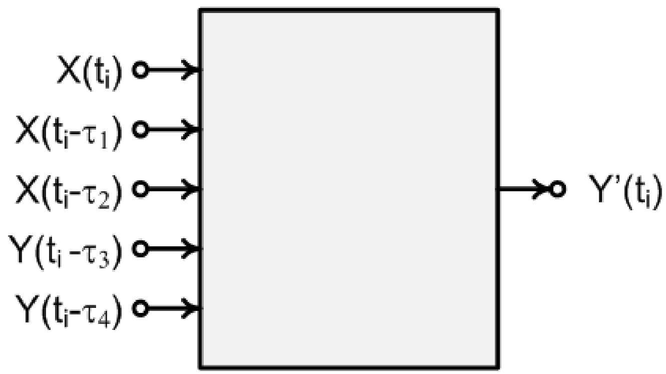
- Must generate behavioural models with multiple feedbacks : eg. *delayed replica's of output voltage*

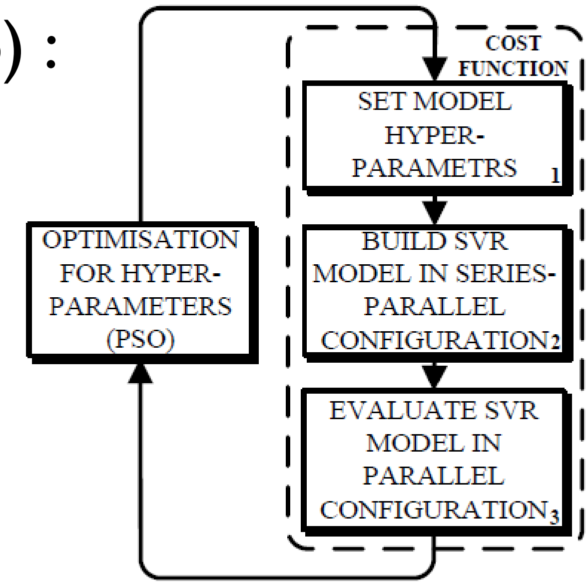


- Training of recurrent behavioural models is a problem
  - You can sample the output of the target system
  - *But you can not sample the output of a non-existent model…*
  - Risk that small modelling errors get amplified

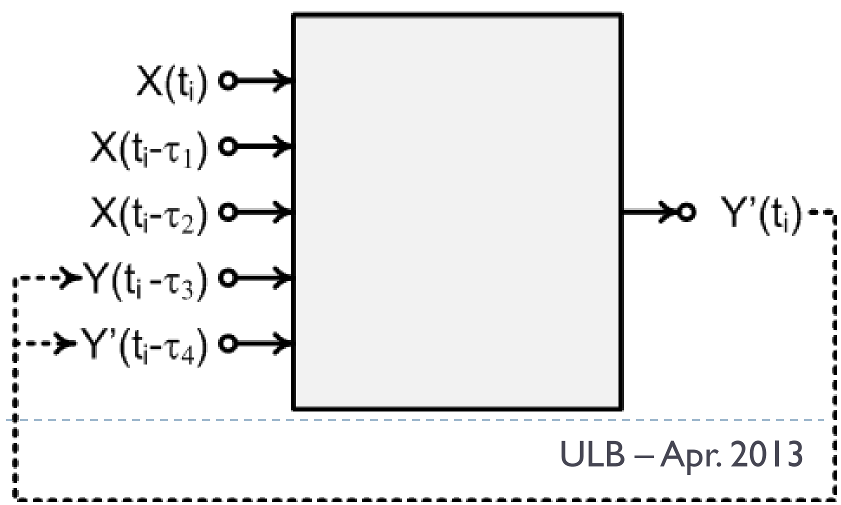
Training of Recurrent Models

- Series-parallel configuration (open-loop) :
  Use training data points as recurrent input





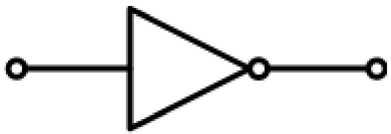
- Compensate the best you can by tuning the regression parameters on the model evaluated in closed-loop configuration

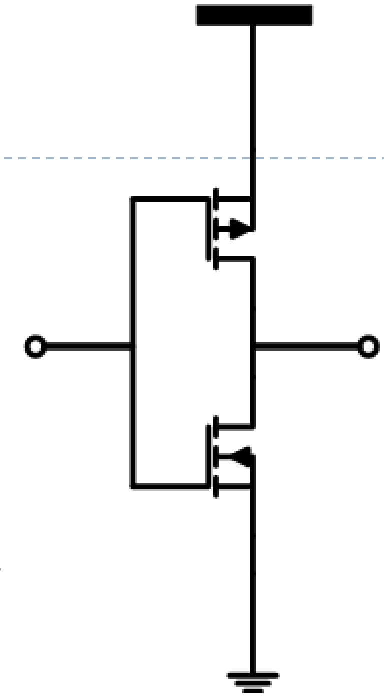


Stability of the models with feedbacks

- The stability analysis of recurrent neural networks is mostly based on simplification of the problem, e.g.
  - Lyapunov's indirect and direct methods in (Cao, 2006)
  - By employing a new Lyapunov–Krasovskii functional, a linear matrix inequality (LMI) approach is developed to establish sufficient conditions for the RNNs to be globally exponentially stable in (Liu, 2006).
- Stability analysis does not include the numerical stability problems of the circuit simulator!

CMOS Inverter Test-case

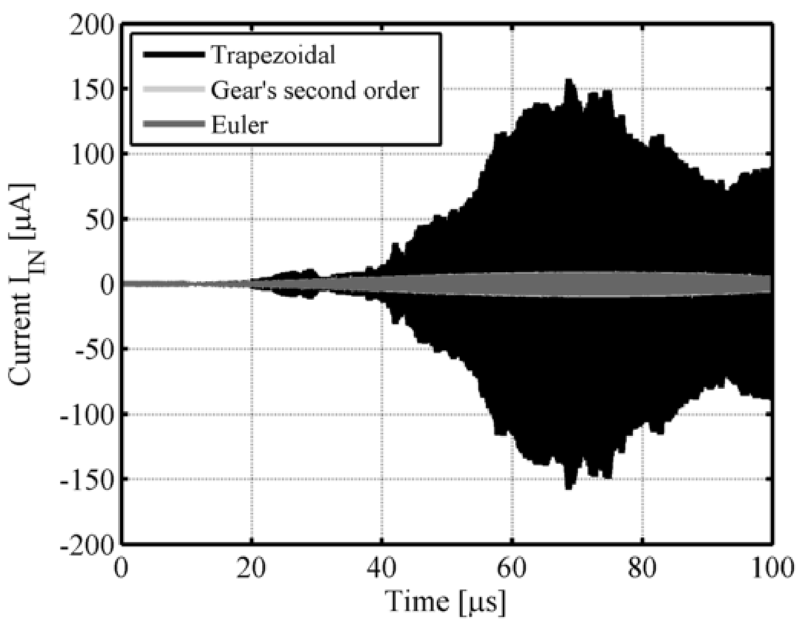


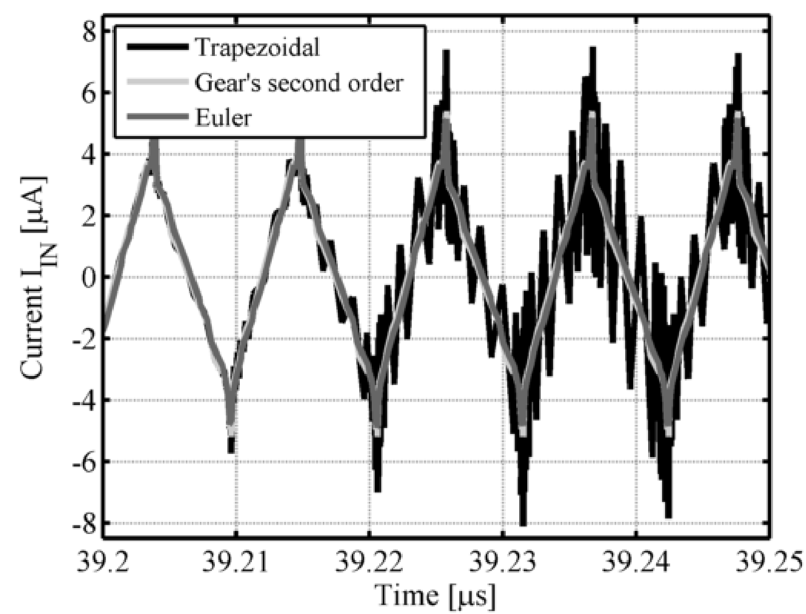


- Basic building block of all logic circuits
- Consists mainly of two transistors
- Switching speed of the output depends on the load (usually capacitive)
- Driving strengths of the transistors depend on output voltage (drain voltage)
- Need to have output feedback in order to capture dynamic switching behaviour

CMOS inverter: Model does not work !

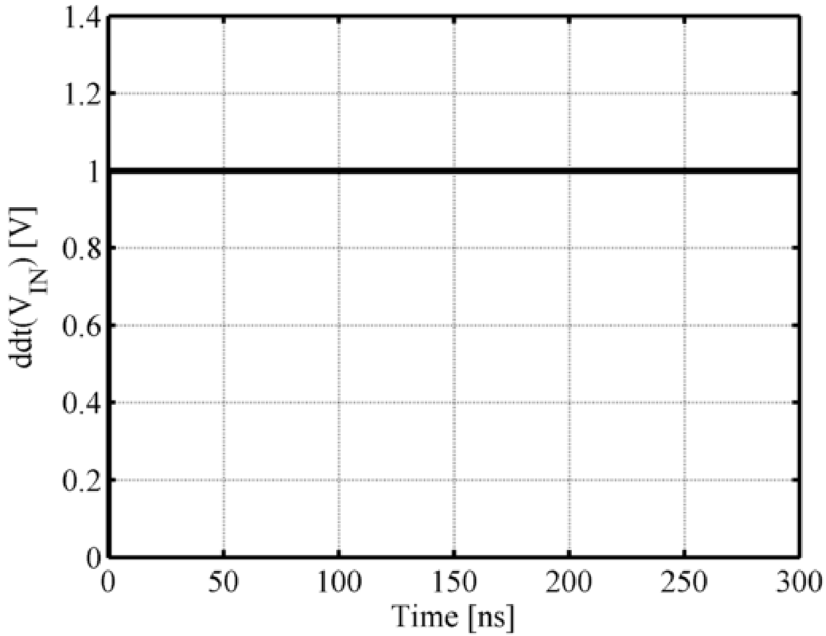
- Results in function of selected integration method
  - Model works fine in MATLAB / PYTHON, not in SPECTRE…
  - Applying a chirp signal on the gate
  - Looking at the gate charging current :

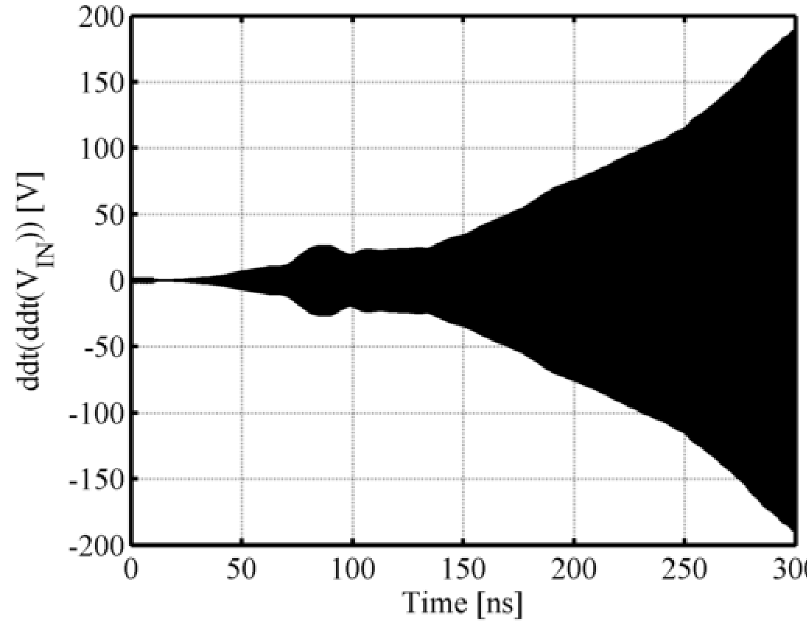




Derivatives are generally problematic

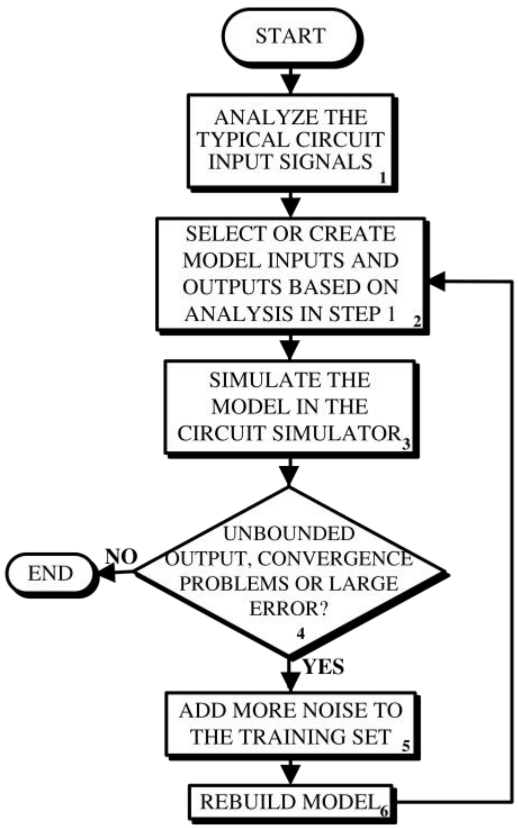
- Derivatives calculated by SPECTRE for a linear ramp :





Algorithmic Solution to the Problem

- Add white noise to the data-set…



Algorithm 1: Method for checking and improving stability of black-box models (CISB)

Step 1: The typical circuit inputs are analyzed in order to create sample cases of normal operation of the circuit.
Step 2: Model inputs and outputs are selected or created based on the analysis done in Step 1.
Step 3: Simulate the model in the circuit simulator.
Step 4: If the output is unbounded, convergence problems occur or the model output significantly deviates from the expected behaviour then go to Step 5, otherwise go to Step 2.
Step 5: Add more noise to the training data set.
Step 6: Rebuild the model and go to Step 2.

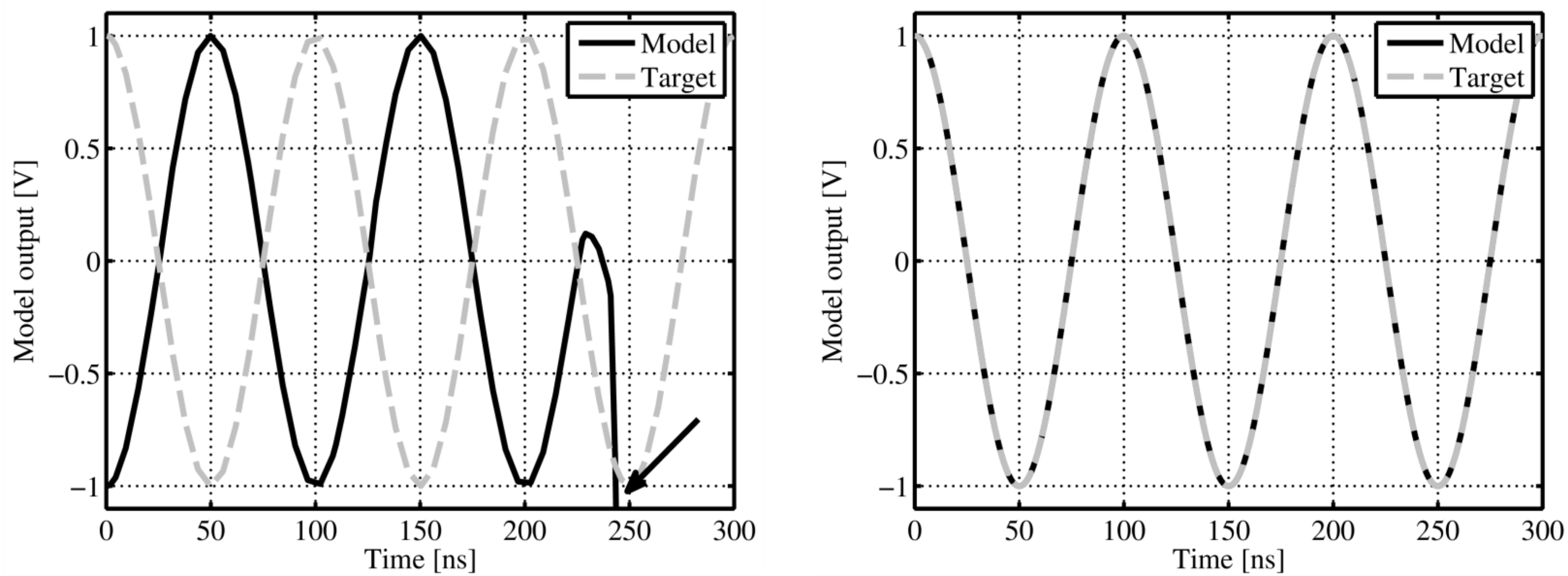
Algorithmic Solution to the Problem

- Adding noise in each iteration. We are trying to find minumum noise level to supress the circuit numerical problems.
- Number of SPICE evaluations should be as small as possible because they are computationally expensive.
- Minimum number of iterations according to the binary search algorithm
- A binary search halves the number of items to check with each iteration, so locating optimal noise level takes logarithmic time.
- Average number of SPICE evaluations is log2(N)-1 where N is a number of possible levels that can be tested, e.g. for SNR from 0 to 100 dB with resolution of 0.5, average number of evaluations is only 6.6!
- If no upper (or lower) limit is defined, average number of evaluations is 2*log2(k)+1 where k is the (unknown) minimum noise level.

Verilog-A Model (CMOS inverter)

```
// Example of a simple model of inverter built by TASVR algorithm.
// The number of SV is limited to 10 SVs (that is the reason why the behavioural descripion is small).
// INITIALISATION
`include "discipline.h
`include "constants.h

// In this simple example, inputs are input voltage (in1E) and delayed input voltage (in2E).
// The output is the output voltage.
// The delayed version can in2E can be generated  inside of Verilog A code by absdelay command.

module Vout_inverter_tran( in1E, in2E, outE, gnd );

electrical in1E, in2E, outE, gnd;

real o1; real i1; real i2;

analog begin

    // We first must scale inputs. In this case we scale to [-1,1] range.
    i1=(((1)-(-1))*(V(in1E, gnd)-(                0)))/((            3.3)-(              0)) + (-1);
    i2=(((1)-(-1))*(V(in2E, gnd)-(-16140891.7306778)))/((15596982.6714803)-(-16140891.7306778)) + (-1);

    // module behavioral description. It is generated by TASVR algorithm and limited to 10SV.
    o1 = (2/(1+exp(-2*((2/(1+exp(-2*(i1*(-1.98355288e+00) + i2*(4.87907886e-01) + (1.24497687e+00))))-1)*(1.48062767e+00) + (2/
(1+exp(-2*(i1*(5.91734491e+00) + i2*(5.96737824e+00) + (2.95055755e-01))))-1)*(-4.02777537e-01) + (2/(1+exp(-2*(i1*
(2.79975258e+00) + i2*(-7.10623655e+00) + (2.78072909e+00))))-1)*(-9.60205344e-02) + (2/(1+exp(-2*(i1*(1.40332998e+01) + i2*
(2.71512546e+00) + (8.63017720e-01))))-1)*(-6.99170641e-01) + (2/(1+exp(-2*(i1*(4.02501497e+00) + i2*(-4.26195915e-
01) + (2.64414994e+00))))-1)*(-1.33922945e+00) + (-1.81352214e-03))))-1);

    // At last, we set the output voltage. Notice rescaling before applying voltage to the output of the model!
    V(outE,gnd) <+ ((o1-(-000000000000000000000001))*((3.2890231145676498414331945e+00)-(-2.6617793710389699157747856e-03)))/
((000000000000000000000001)-(-000000000000000000000001))+(-2.6617793710389699157747856e-03);

endmodule
```

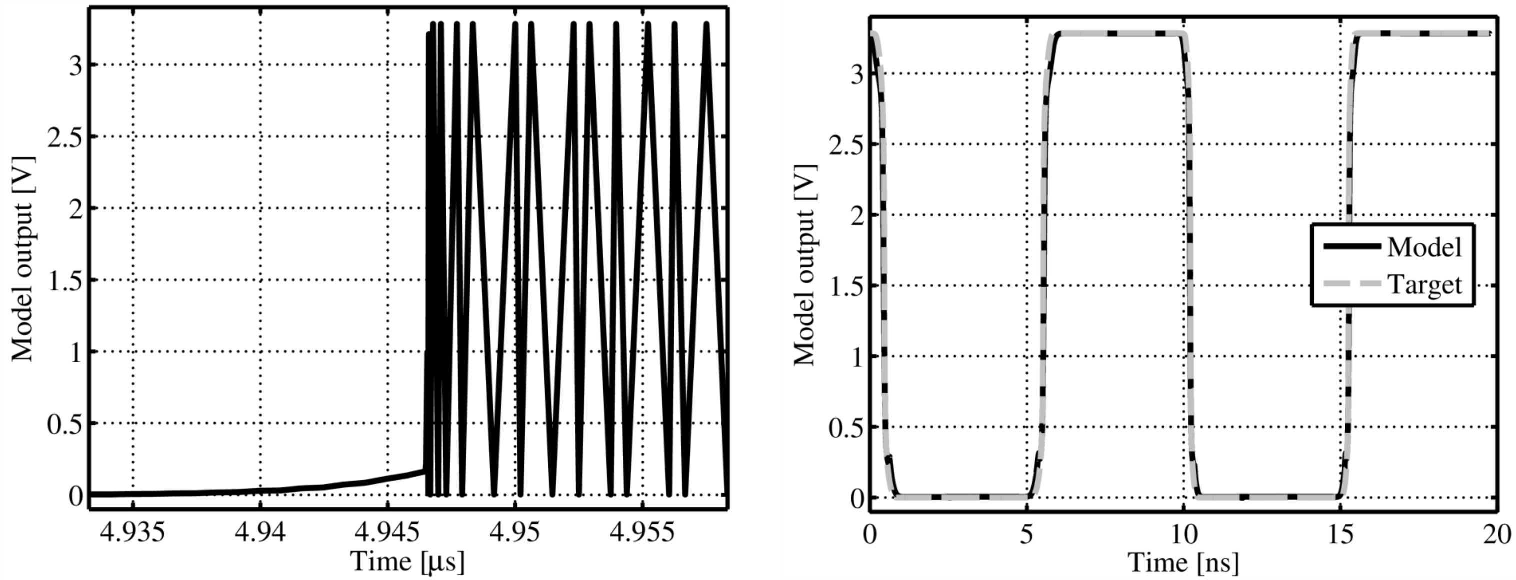
CMOS inverter model with CISB



(a) (b)

- (a) before and (b) after application of the CISB method
  - Signal to noise (SNR) ratio of the added white Gaussian noise to the feedback signal is equal to 75 dB
  - NRMSE = 2.1E-5 (with CISB)

CMOS inverter model with CISB



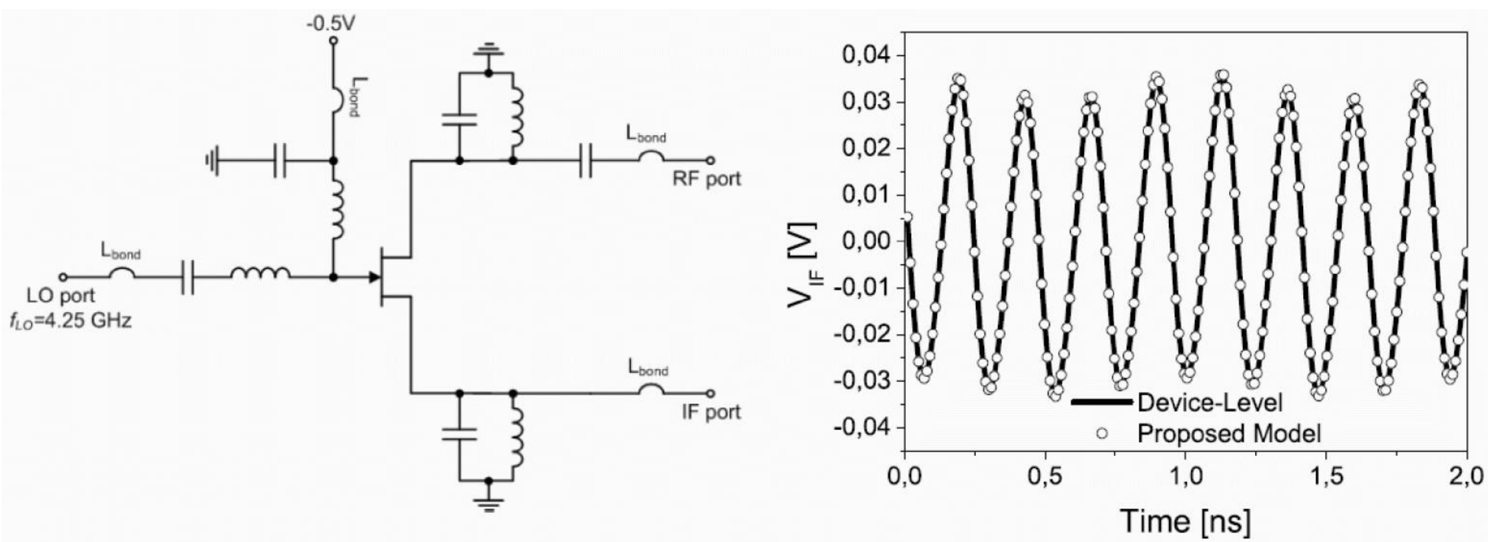
(a) (b)

- (a) before and (b) after application of the CISB method
  - (a) zoomed in at the point where the simulator diverges
  - Signal to noise (SNR) ratio of the added white Gaussian noise to the feedback signal is equal to 50 dB.
  - NRMSE = 0.048

Modelling of Resistive Mixer

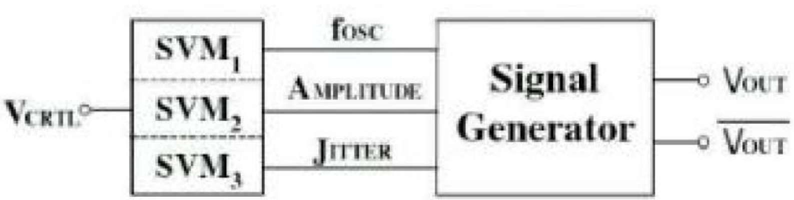
- Training: - frequency f_{RF} is varied from 5.1 GHz to 5.3 GHz (50 MHz step) - power of V_{RF} is varied from -30 dBm to -40 dBm (step 2 dBm) - IF port is loaded with 45, 50 and 55 Ohm resistor

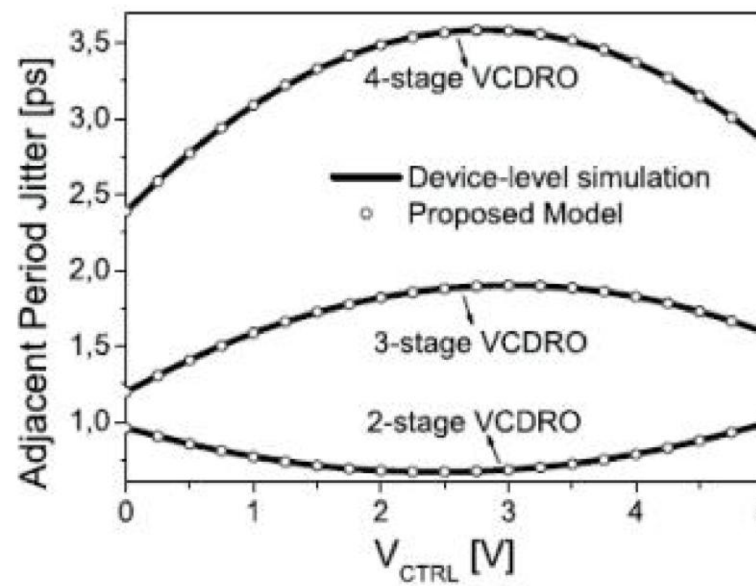
- Testing: - frequency f_{RF} is 5.225 GHz - power of V_{RF} is set to -35 dBm - IF port loaded with 52.5 Ohm- Not used in training set!



Modelling of VCO

- GaAs 0.5 um SCFL Voltage Controlled Differential Oscillator
- Using signal generator as the output stage
  - Only 10 to 20 input-output pairs needed!





Conclusions (modelling tips)

- Mix generic stimuli (e.g. chirp waveform) with typical stimuli for the target electronic circuit to obtain training data set.
- Scale the data before applying machine learning techniques.
- Use delays to capture dynamic behaviour rather than derivatives, as it will result in much more stable model code
- Use feature/training data point selection methods.
- Don't use ANNs, they are prone to overfitting.
- If possible, simplify the modelling problem!
  Example: VCO signal generator.
- Always test in the circuit simulator.
  Adding white noise can improve stability.

References

- Barford, Lee A. et al. (Feb. 2005). " Method and apparatus for extraction of nonlinear black-box behavioral models from embeddings of the time-domain measurements". Patent US 6,850,871 (US).
- Cao, Yi, Runtao Ding, and Q. J. Zhang (2006). " State-space dynamic neural network technique for high-speed IC applications: modeling and stability analysis". In: IEEE Transactions on Microwave Theory and Techniques 54.6, pp. 2398–2409.
- Ceperic, V. and A. Baric (2004a). " Modeling of analog circuits by using support vector regression machines". In: Proceedings of the 11th IEEE International Conference on Electronics, Circuits and Systems, ICECS 2004, pp. 391–394.
- — (May 2004b). "Modelling of SCFL circuits". In: Proceedings of the 27th International Convention MIPRO 2004: conferences Microelectronics, Electronics and Electronics Technologies (MEET) and Hypermedia and Grid System (HGS), pp. 72–77.

References

- Feo, O. De and M. Storace (2005). " PWL approximation of nonlinear dynamical systems, part II: identification issues". In: Journal of Physics: Conference Series 22, pp. 30–42. Guo, Yunchuan et al. (2007).
- Guo, Yunchuan et al. (2007). " A Support Vector Regression Nonlinear Model for SiC MESFET". In: Proceeding of 2007 International Workshop on Electron Devices and Semiconductor Technology, EDST 2007, pp. 153–156.
- Guyon I., et al. (2006). "Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing)". Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- Jaeger, H. The "echo state" approach to analysing and training recurrent neural networks. Tech. rep. 148. GMD – German National Research Institute for Computer Science.
- — (2002). Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach. GMD Report. St. Augustin-Germany: Fraunhofer Institute AIS.
- Karsmakers, P. et al. (2008). " Least Squares Support Vector Machines for Modelling Electronic Devices". In: European Conference on the Use of Modern Information and Communication Technologies (ECUMICT). Ghent, Belgium, pp. 1–4.
- Litovski, V. B. et al. (1992). " MOS Transistor Modeling Using Neural Network". In: Electronics Letters 28.18, pp. 1766–1768.

References

- Litovski, V. B. et al. (2004). " ANN application to modelling of the D/A and A/D interface for mixed-mode behavioural simulation". In: Journal of Circuits, Systems and Computers 13.1, pp. 181–192.
- Liu Yurong Zidong Wang, Xiaohui Liu (2006), Global exponential stability of generalized recurrent neural networks with discrete and distributed delays, Neural Networks, Volume 19, Issue 5, June 2006, Pages 667-675, ISSN 0893-6080, 10.1016/j.neunet.2005.03.015.
- Moriyasu, Hiro (Sept. 1994). " Behavioral model parameter extractor". Patent US 5,349,539 (US).
- Patan K. and M. Patan. (2010). "Selection of Training Data for Locally Recurrent Neural Network". In: Artificial Neural Networks – ICANN 2010. Ed. by K. Diamantaras, W. Duch, and L. S. Iliadis. Vol. 6353. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 134–137.
- Rutenbar, R. A., G. G. E. Gielen, and J. Roychowdhury (2007). " Hierarchical modeling, optimization, and synthesis for system-level analog and RF designs". In: Proceedings of the IEEE 95.3, pp. 640–669.
- Russell, J., P. Norvig (2009). Artificial Intelligence: A Modern Approach. Pearson Education.
- Schetzen, M. (1980). The Volterra and Wiener Theories Nonlinear Systems. New York: Wiley.
- Sevic, John F. and Gary R. Simpson (June 2006). " Method and apparatus for model extraction". Patent application 20060116857 (US).

References

- Settles, B. (2009). “Active Learning Literature Survey”. In:Computer Sciences Technical Report 1648. University of Wisconsin–Madison.
- Taher, H., D. Schreurs, and B. Nauwelaers (2008). ” Black box modelling of the Op-Amp including switching power supply on effect”. In: AEU - International Journal of Electronics and Communications 62.7, pp. 544–548.
- Verspecht, J. (2005). ”Large-signal network analysis”. In: IEEE Microwave Magazine 6.4, pp. 82–92.
- Wang,J., P. Neskovic, and L. N. Cooper (2005). “Training data selection for support vector machines”. In: Proceedings of the First international conference on Advances in Natural Computation - Volume Part I. ICNC’05. Changsha, China: Springer-Verlag, 2005, pp. 554–564.
- Wood, John and David E. Root (2005). Fundamentals of Nonlinear Behavioral Modelling for RF and Microwave Design. Artech House, Inc.
- Zhang, Lei and Qi-Jun Zhang (2008). ”Neuro-space mapping technique for semiconductor device modeling”. In: Optimization and Engineering 9.4, pp. 393–405.