Fast Time-Domain Simulation Through Combined Symbolic Analysis and PWL Modeling

Hui Zhang and Alex Doboli

Electrical and Computer Engineering Department
Stony Brook University (State University of New York),
Stony Brook, NY 11794
Email: {huizhang, adoboli}@ece.sunysb.edu

BMAS'04 - October 22, 2004

Motivation

- **Design of analog IP cores is the bottleneck in SoC design (costly, tedious, long design times)**
  - Automated synthesis methods are needed beyond circuit level
- **Fast simulation is essential for any exploration-based synthesis methods**
- **For analog synthesis, existing circuit simulation tools (Hspice, Spectre) are too slow to be used for design space exploration**
- **In the digital world, customized simulators (compiled-code simulators) can be produced for a design (much faster).**
  - Why can this not happen in the analog/mixed-signal world?

Objective

To develop a unified modeling and simulation methodology for analog and mixed-signal (AMS) circuit and system synthesis

- Separation of critical from non-critical blocks
- Flexibility in achieving different simulation accuracy
- Simulation across domains
- Handling of nonlinearities
- Compiler inspired methods

Contributions

- A new approach to fast time-domain simulation of nonlinear AMS systems
- 100x speedup in our earlier experiments
- First simulation methodology for CT AMS circuit synthesis purpose
- First compiled-code simulation for CT AMS systems with nonlinear parameters

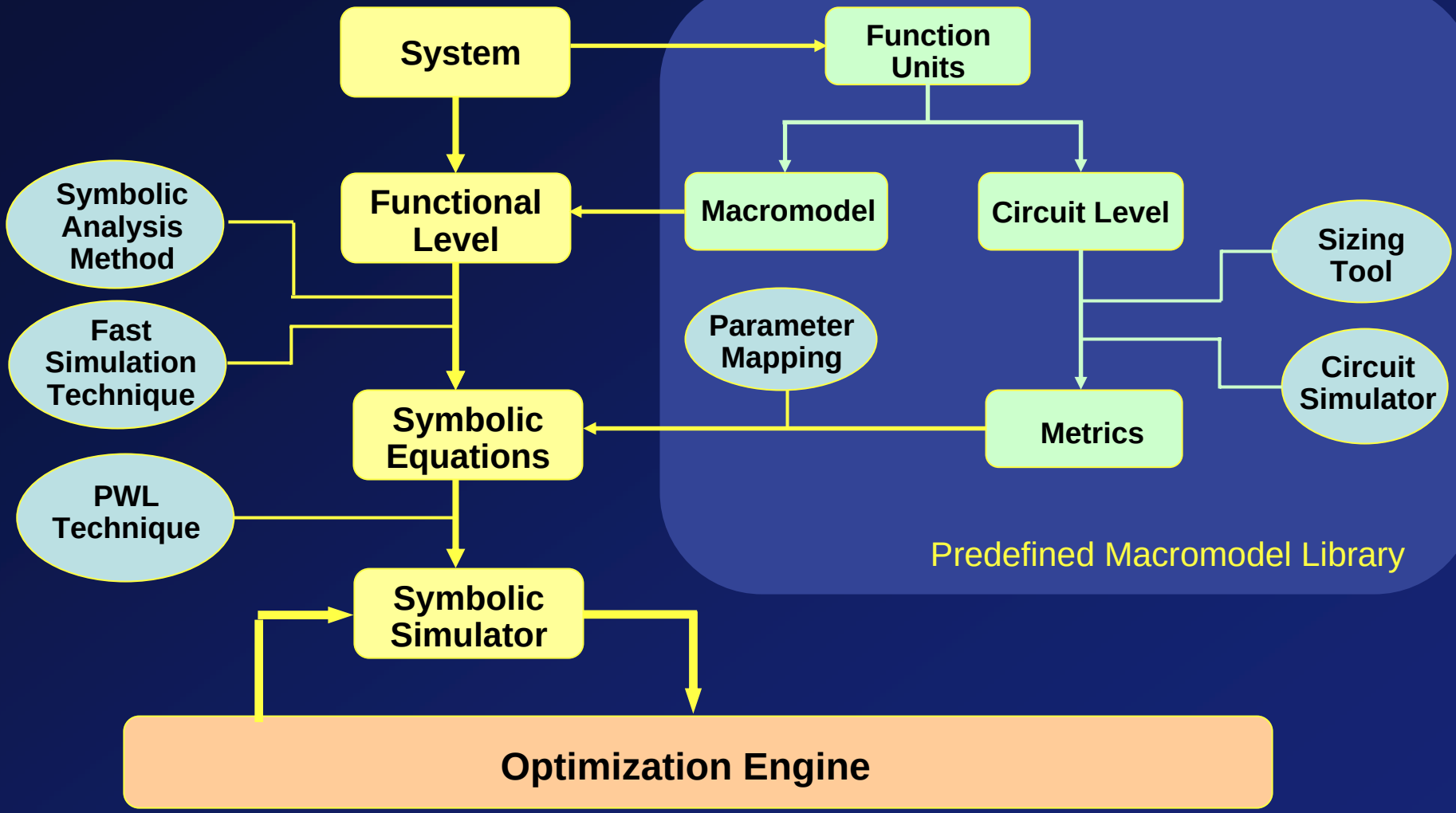
In this Paper

- **CT Σ-Δ ADC time-domain simulation as a case study.**
- **At the system-level, the method uses symbolic description of the ADC netlist**
- **Functional blocks are replaced with macromodels, which include circuit non-idealities and nonlinear behavior.**
- **Non-linear parameters are expressed using PWL models, which are created automatically through model extraction from trained neural networks (NN) .**
- **Bottom-up behavioral modeling procedure, which successively creates structural models. The composition rules between different levels were defined.**
- **Compiled-code simulation method for analog and mixed-signal systems based on code generation from symbolic descriptions**
- **Simulator code can be optimized to avoid convergence problems**

Related Work

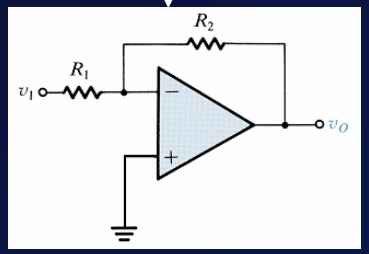
- **Behavioral modeling based simulation:**
  - G. Gielen, *et al*, "An analytical Integration Method for the Simulation of Continuous-Time Σ-Δ Modulators", *IEEE Trans. CADICS*, Vol. 23, No. 3, March 2004
- **Symbolic analysis based simulation:**
  - A. Doboli, R. Vemuri, " A Regularity-Based Hierarchical Symbolic Analysis Method for Large-Scale Analog Networks", *IEEE Trans. Circuits and Systems-II*, Vol. 48, No. 11, Nov. 2001
- **Numerical simulators:**
  - E. Acar, F. Dartu, L. Pileggi, "TETA: Transistor-Level Waveform Evaluation for Timing Analysis", *IEEE Trans. CADICS*, Vol. 21, No. 5, May 2002
  - Z. Li and R. Shi, "SILCA: Fast-yet-accurate time-domain simulation of VLSI circuits with strong parasitic coupling effects", *ICCAD'03*, Nov. 2003

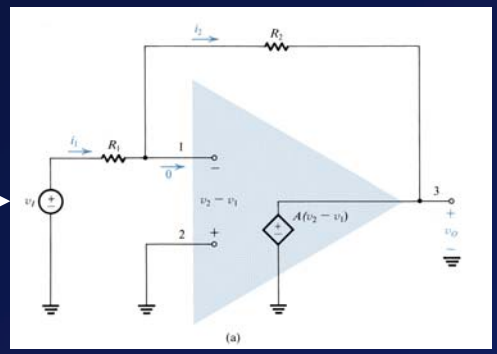
Synthesis Methodology (Rhapsody)



Simulation Methodology

Vo(t)





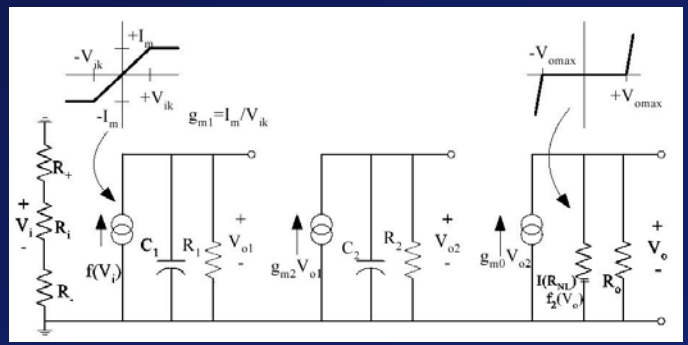
$$i_I = \frac{v_I + v_O / A}{R_1}$$

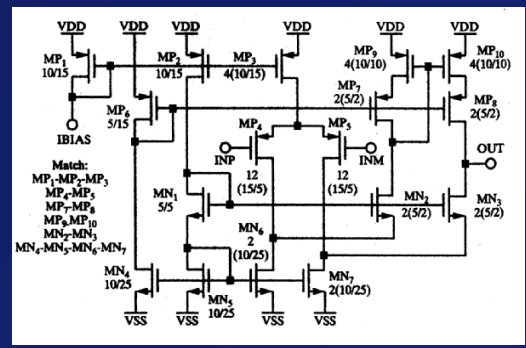
$$V_o = -\frac{V_o}{A} - \left(\frac{v_I + v_O / A}{R_1} \right) \cdot R_2$$

$$G = \frac{V_O}{V_I} = \frac{-R_2 / R_1}{1 + (1 + R_2 / R_1) / A}$$

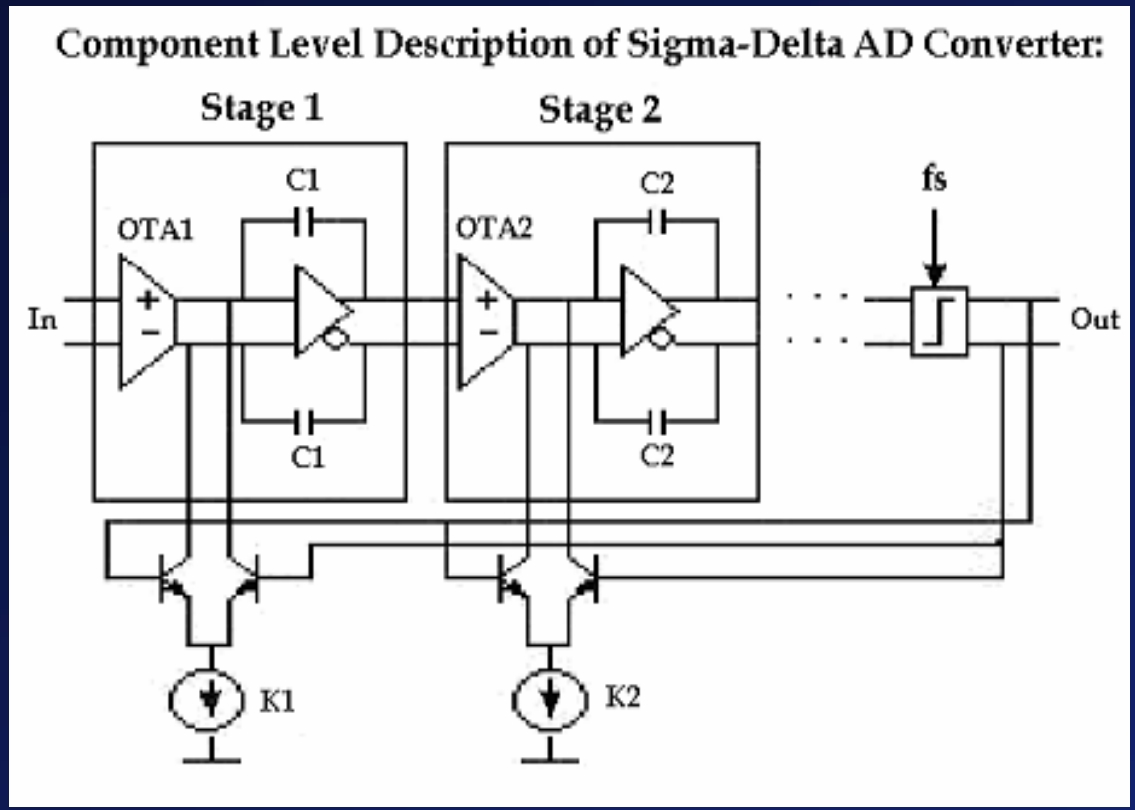
0.?? s

Vo(t)





Generic Structure of CT Single-Loop Σ-Δ Modulators



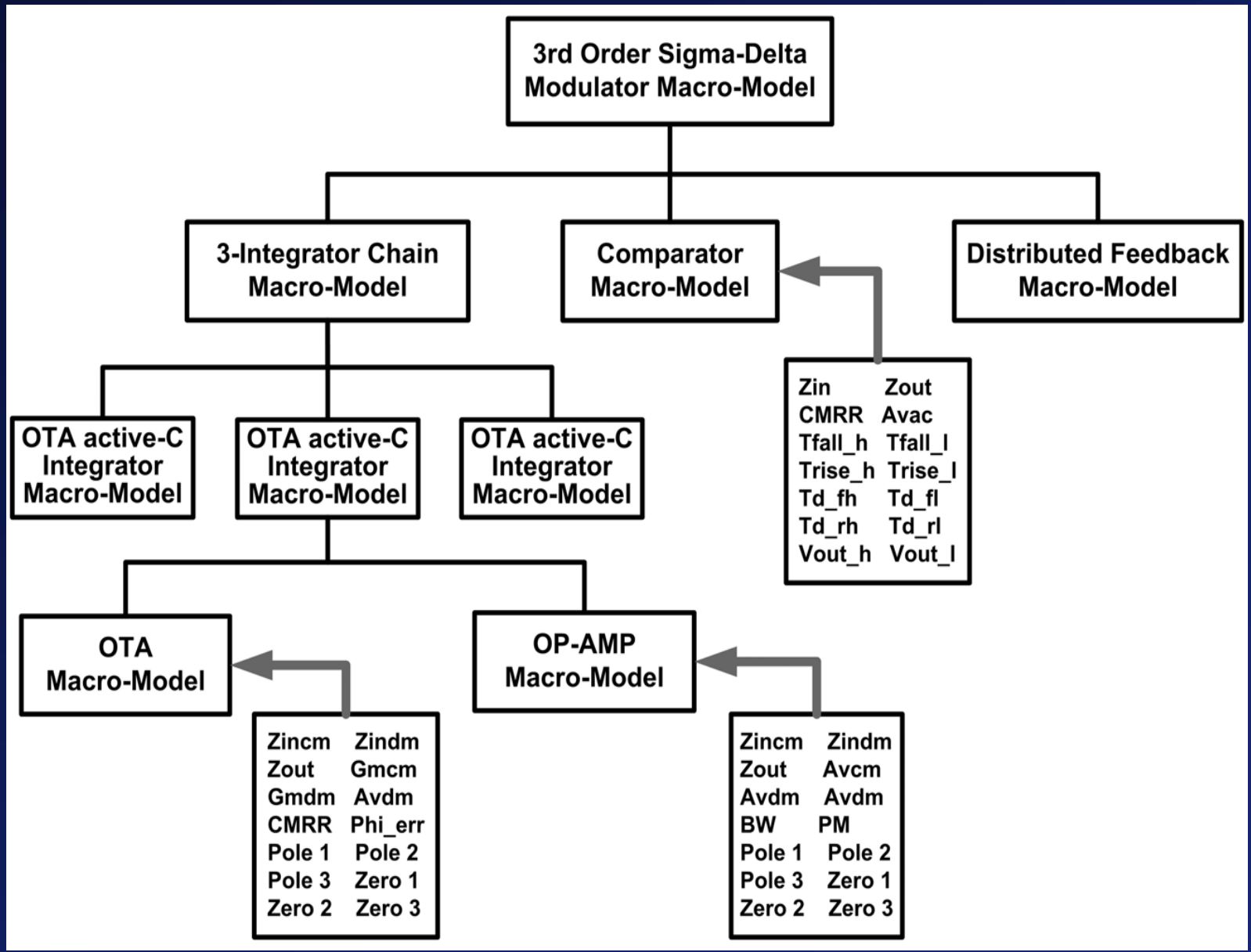
Basic Blocks

- OTA
- OPAMP-C
- Comparator
- DAC

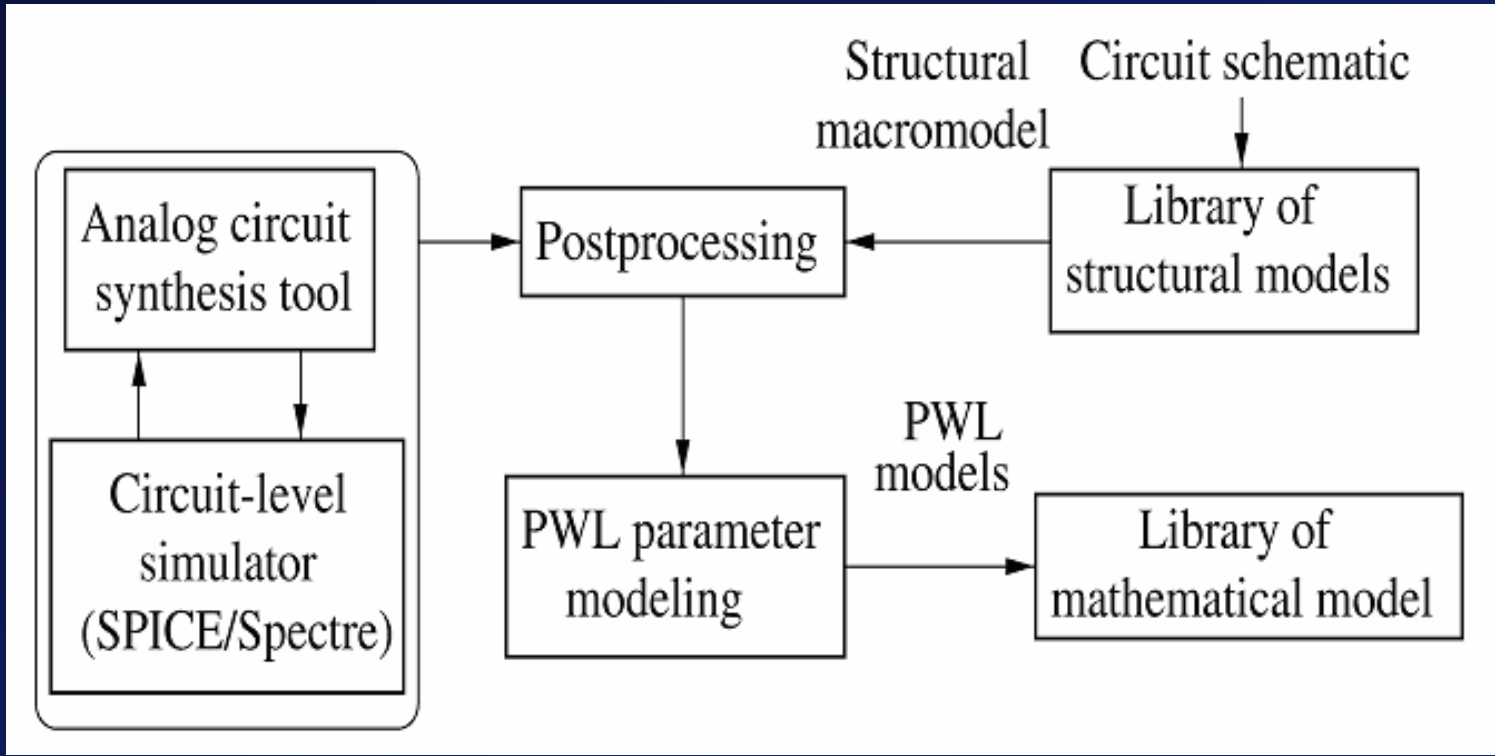
Composed Blocks

- One-stage
- Three-Stage

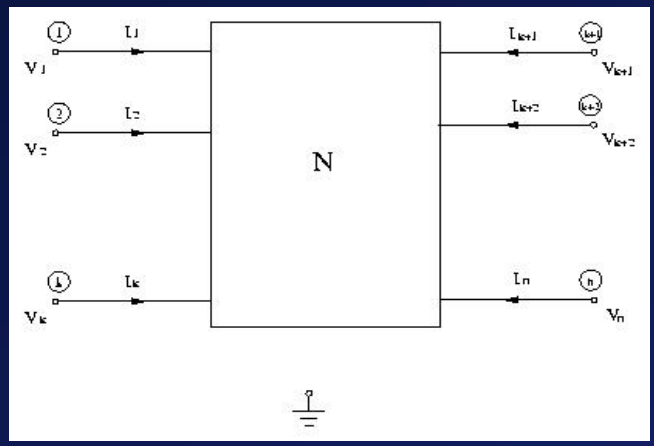
Pattern Structure of 3rd-order Single-loop Σ-Δ Modulator



Circuit Modeling Method



Block Description



$$x(t+h) = A\,x(t) + B\,u(t)$$

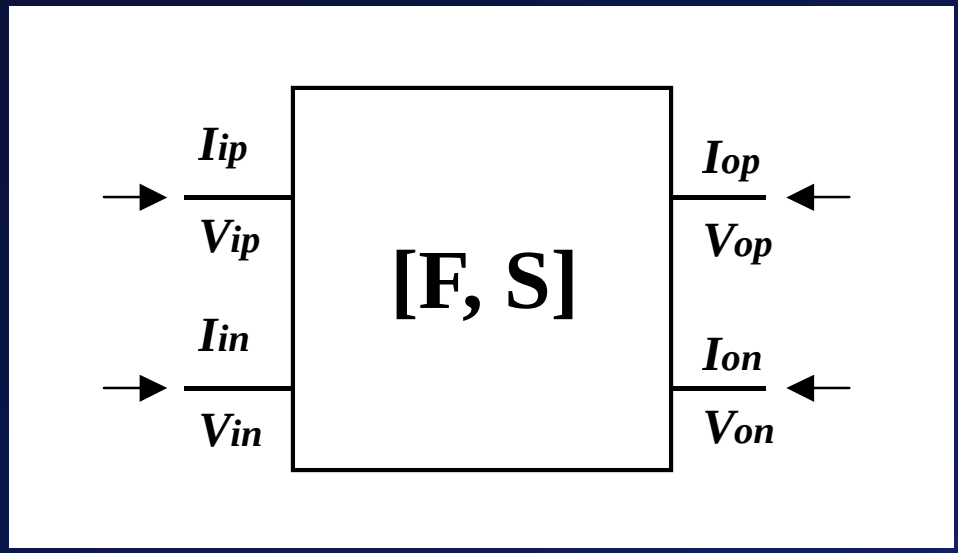
$$S(t) = D\,x(t)$$

$$y(t) = F\,u(t) + S(t)$$

x **:** state variables ***u*** **:** knowns ***y*** **:** unkowns

F **:** function Matrix ***S*** **:** state vector

Block Description - Example



F - Function Sub-matrix

Constant after set

S - State Vector

Update each time step

$$\begin{pmatrix} I_{ip} \\ I_{in} \\ I_{op} \\ I_{on} \end{pmatrix} = \begin{pmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \\ f_{31} & f_{32} & f_{33} & f_{34} \\ f_{41} & f_{42} & f_{43} & f_{44} \end{pmatrix} \begin{pmatrix} V_{ip} \\ V_{in} \\ V_{op} \\ V_{on} \end{pmatrix} + \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix}$$

Unknowns

Function sub-matrix

Knowns

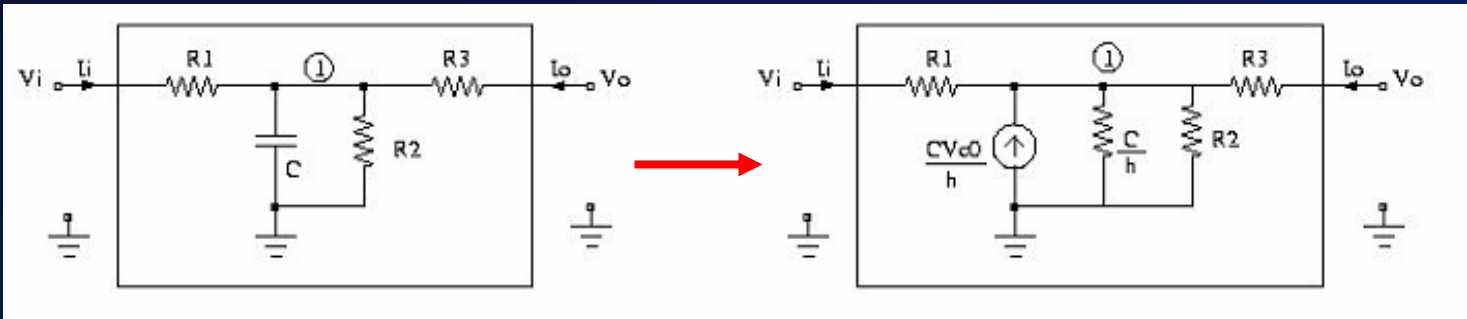
State vector

Basic/Terminal block analysis

Steps:

1. Replace all the energy storage elements with their time-domain companion models
2. Replace all the nonlinear elements with their PWL models
3. Symbolic Analysis to get expressions

Basic/Terminal block analysis

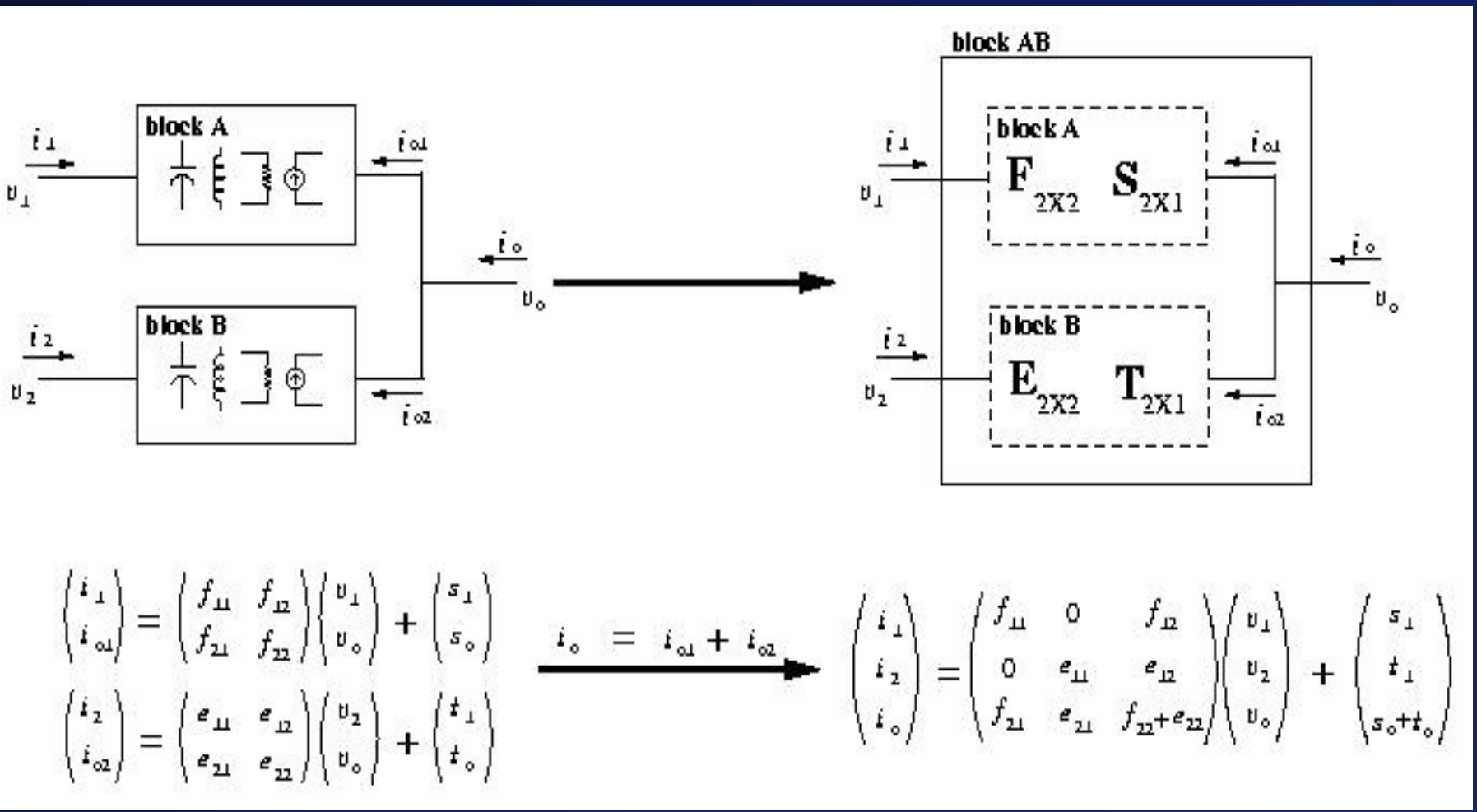


$$\begin{pmatrix} G_1 & -G_1 & 0 \\ -G_1 & G_1+G_2+G_3+\dfrac{h}{C} & -G_3 \\ 0 & -G_3 & G_3 \end{pmatrix} \begin{pmatrix} v_i \\ v_1 \\ v_o \end{pmatrix} = \begin{pmatrix} i_i \\ \dfrac{Cv_{c0}}{h} \\ i_o \end{pmatrix}$$

$$v_{c0}^{+} = \begin{pmatrix} \dfrac{C}{CG_2+h} & \dfrac{C}{CG_2+h} \end{pmatrix} \begin{pmatrix} i_i \\ i_o \end{pmatrix} + \frac{C^2}{C(CG_2+h)} v_{c0}$$

$$\begin{pmatrix} v_i \\ v_o \end{pmatrix} = \begin{pmatrix} \dfrac{1}{G_1}+\dfrac{C}{CG_2+h} & \dfrac{C}{CG_2+h} \\ \dfrac{C}{CG_2+h} & \dfrac{CG_2+CG_3+h}{G_3(CG_2+h)} \end{pmatrix} \begin{pmatrix} i_i \\ i_o \end{pmatrix} + \begin{pmatrix} \dfrac{C^2}{h(CG_2+h)} \\ \dfrac{C^2}{h(CG_2+h)} \end{pmatrix} v_{c0}$$

Composed/Middle block analysis - example



$$\begin{pmatrix} i_1 \\ i_{o1} \end{pmatrix} = \begin{pmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{pmatrix} \begin{pmatrix} v_1 \\ v_o \end{pmatrix} + \begin{pmatrix} s_1 \\ s_o \end{pmatrix}$$

$$\begin{pmatrix} i_2 \\ i_{o2} \end{pmatrix} = \begin{pmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{pmatrix} \begin{pmatrix} v_2 \\ v_o \end{pmatrix} + \begin{pmatrix} t_1 \\ t_o \end{pmatrix}$$

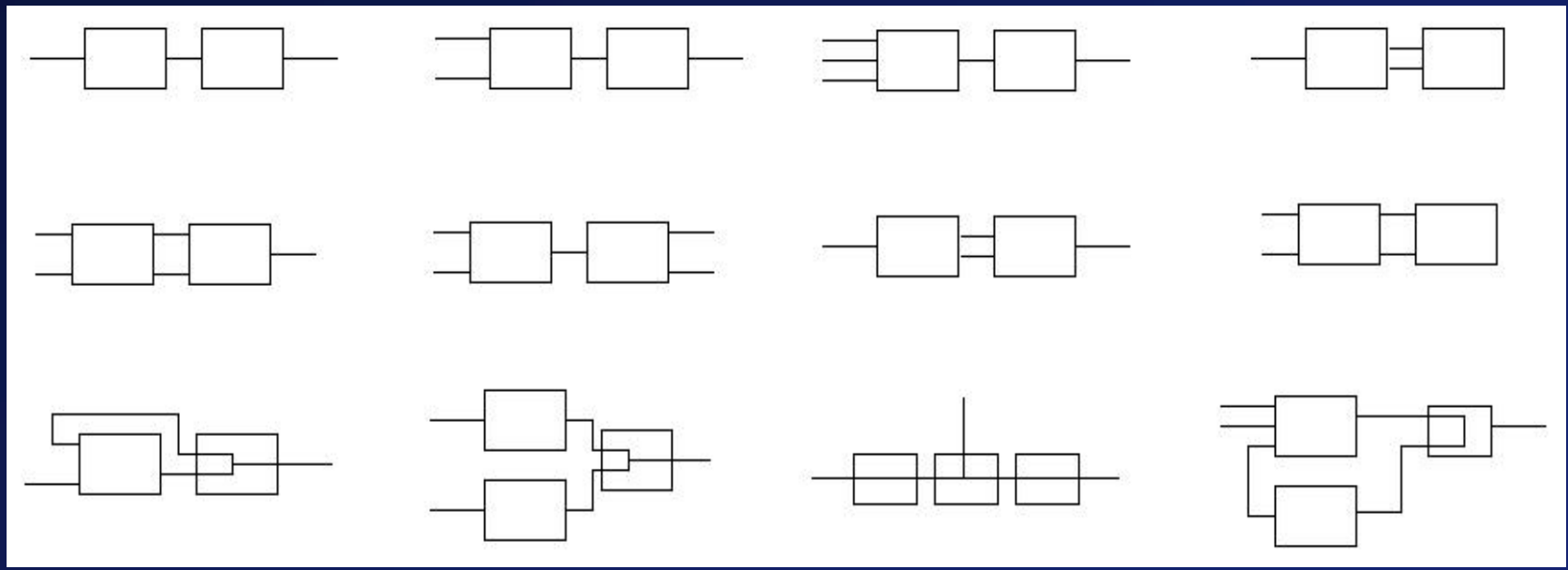
$$i_o = i_{o1} + i_{o2}$$

$$\begin{pmatrix} i_1 \\ i_2 \\ i_o \end{pmatrix} = \begin{pmatrix} f_{11} & 0 & f_{12} \\ 0 & e_{11} & e_{12} \\ f_{21} & e_{21} & f_{22}+e_{22} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_o \end{pmatrix} + \begin{pmatrix} s_1 \\ t_1 \\ s_o+t_o \end{pmatrix}$$

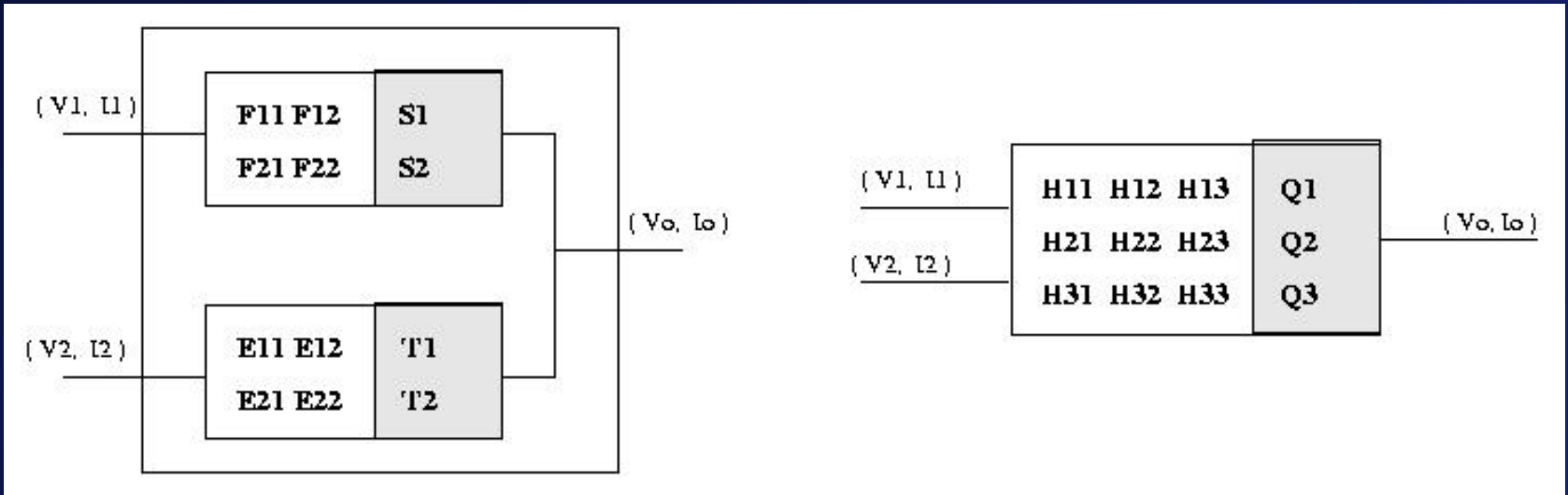
Composed/Middle block analysis - GIT

Generic Interconnection Template (GIT)

– "A regularity-based hierarchical symbolic analysis method for large-scale analog networks," *IEEE Trans. On Circuits and Systems – II*, vol. 48, no. 11, pp. 1054-1068, Nov. 2001



Composed/Middle block analysis

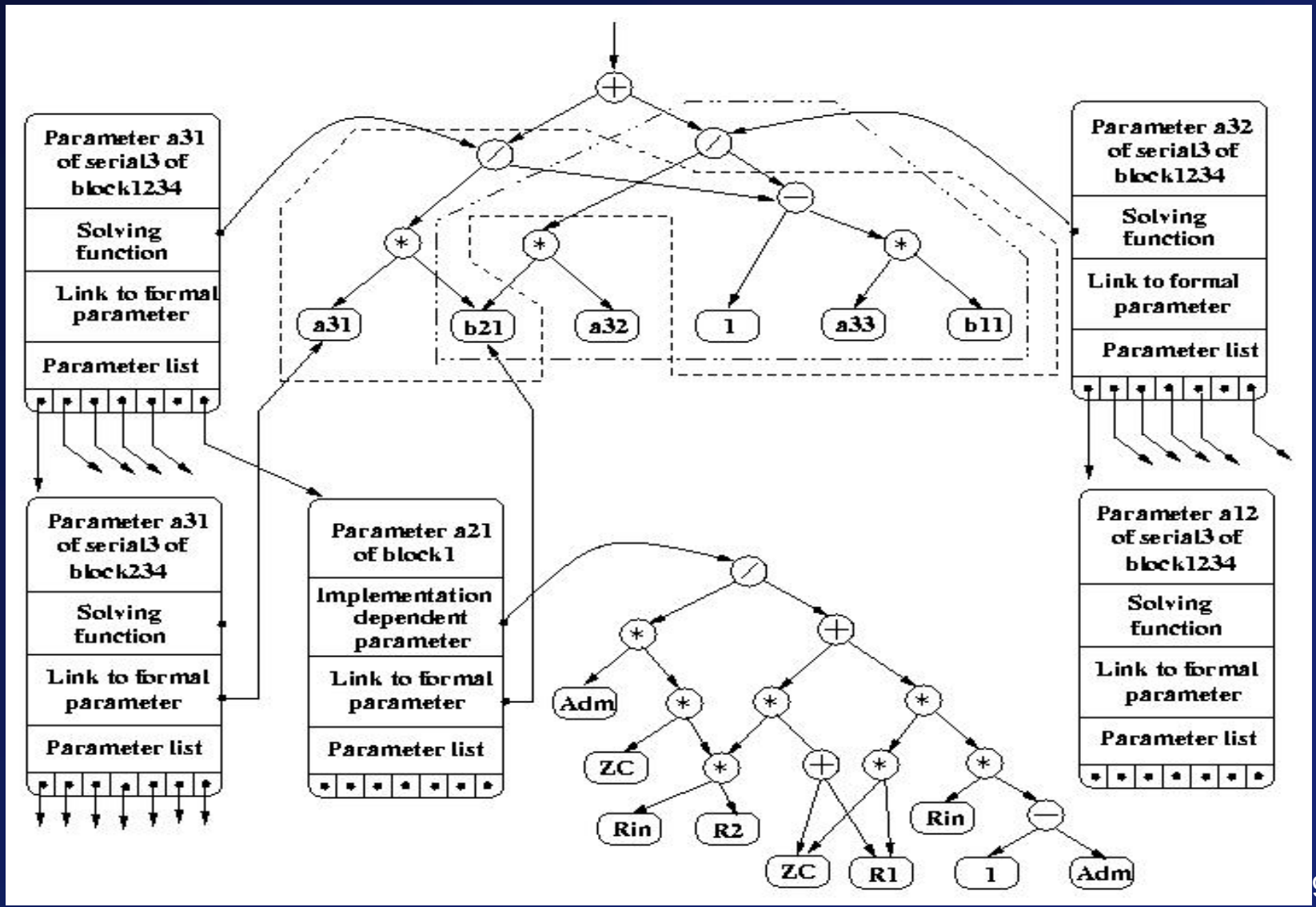


H11 = F11	H12 = 0	H13 = F12	Q1 = S2
H21 = 0	H22 = E11	H23 = E12	Q2 = T2
H31 = F21	H32 = E21	H33 = F22 + E22	Q3 = S1 + T1

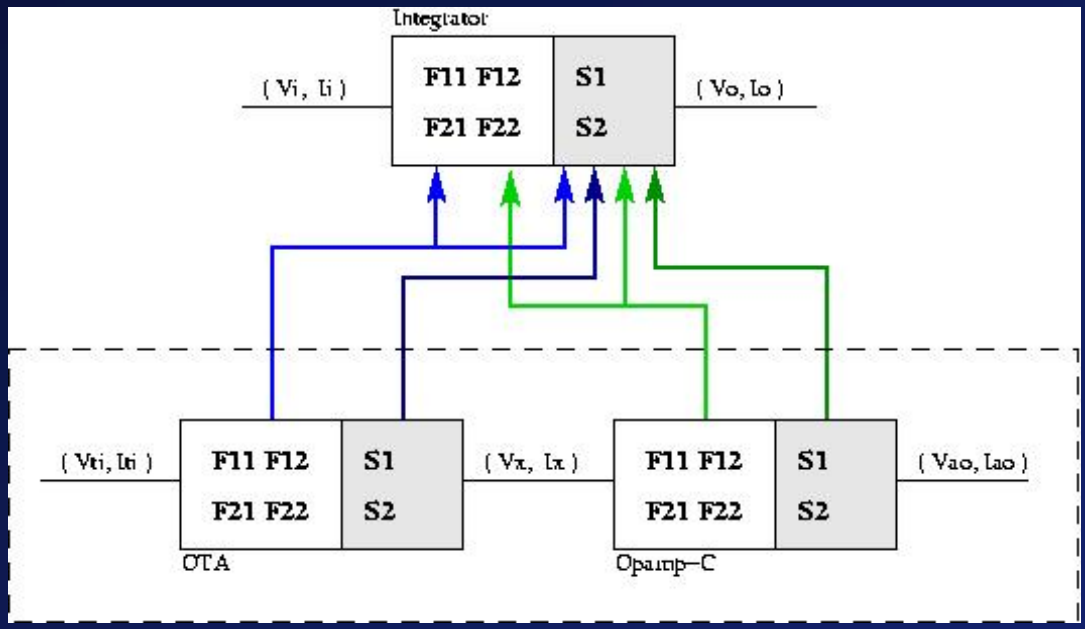
Composed/Middle block analysis - APT

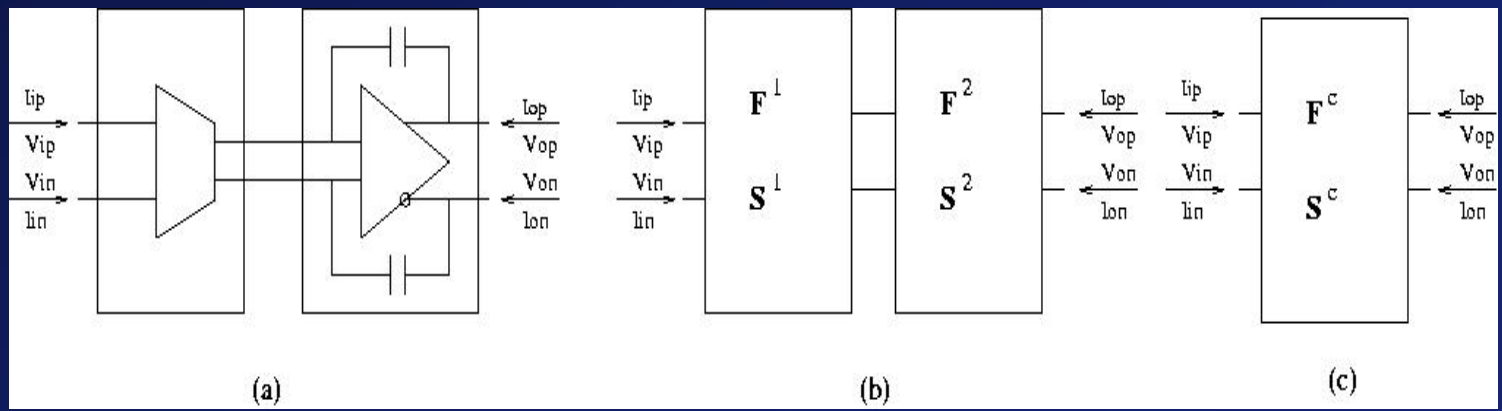
Analog Performance Tree (APT)

– "A regularity-based hierarchical symbolic analysis method for large-scale analog networks," Trans. On CAS-II Nov. 2001

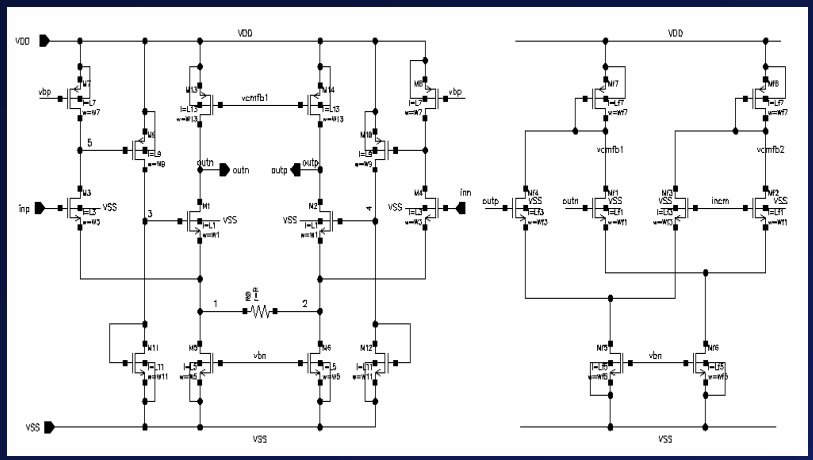


Symbolic Composition Rules (SCR)

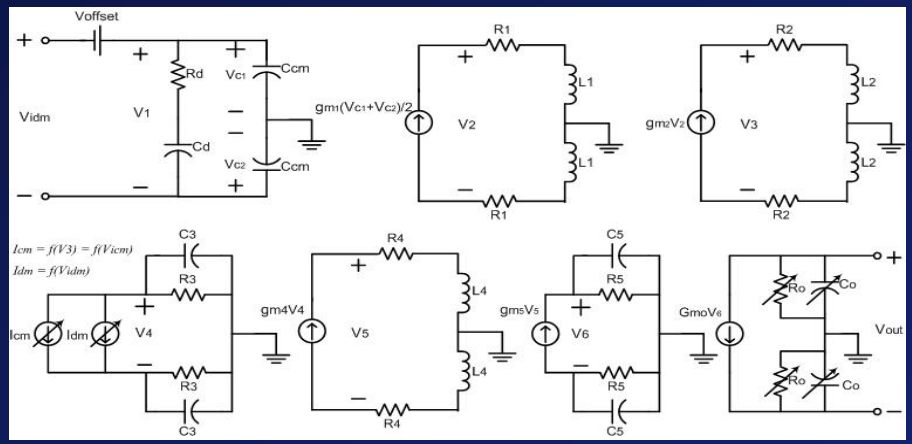




OTA Macromodel



A. Leuciuc *et al*., "A Highly Linear Low-Voltage MOS Transconductor", *ISCAS'02*



G. J. Gomez *et al*, "A nonlinear macromodel for CMOS OTAs", *ISCAS'95*

W1	L1	W2	L2	W3	L3	…	Gain	Zin	Zout	CMRR	Pole1	Pole2	…	Vos	Rd	Ccm	R1	L1	Gm	…

SIZES PERFORMANCE PARAMETERS

$$\begin{pmatrix} I_{ip} \\ I_{in} \\ I_{op} \\ I_{on} \end{pmatrix} = \begin{pmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \\ f_{31} & f_{32} & f_{33} & f_{34} \\ f_{41} & f_{42} & f_{43} & f_{44} \end{pmatrix} \begin{pmatrix} V_{ip} \\ V_{in} \\ V_{op} \\ V_{on} \end{pmatrix} + \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix}$$

Mapping Rules for OTA

- V_{os} was obtained trough SPICE/Spectre simulation.
- Ccm = 1 / (4 π |Zicm(f1)|).
- Cd= 1/(2πf1Zidm[Im]|f1) - Ccm.
- Rd = Zidm[Re](Cd + Ccm) 2 / Cd2.
- C3 depends on the position of the first pole, as given by SPICE simulation.
- L4 relates to the dominant zero, as offered by SPICE simulation.
- 1/(RoCo) is the frequency of the dominant pole.
- Ro results from SPICE/Spectre simulation directly
- R1, R2, L1 and L2 are determined by common mode zeros.
- R3 = R4 = R5 = 1k Ohm are constant.
- gm4 = gmo = 1mmhos are fixed.
- L1 = L2 = 0 are constant.
- gm was obtained from SPICE/Spectre simulation.

OTA Function Sub-Matrix Entries

F11 = Ccm/h + Cd/(h + Cd*Rd);

F12 = - (Cd/(h + Cd*Rd));

…

F31 = (8*h*Gm*Gm4*Gm5*Gmo*R3*(L4 + h*R4)*R5)/((h + C3*R3)*(h + C5*R5));

F32 = (- 8*h*Gm*Gm4*Gm5*Gmo*R3*(L4 + h*R4)*R5)/((h + C3*R3)*(h + C5*R5));

…

F42 = (8*h*Gm*Gm4*Gm5*Gmo*R3*(L4 + h*R4)*R5)/((h + C3*R3)*(h + C5*R5));

…

OTA State Vector Entries

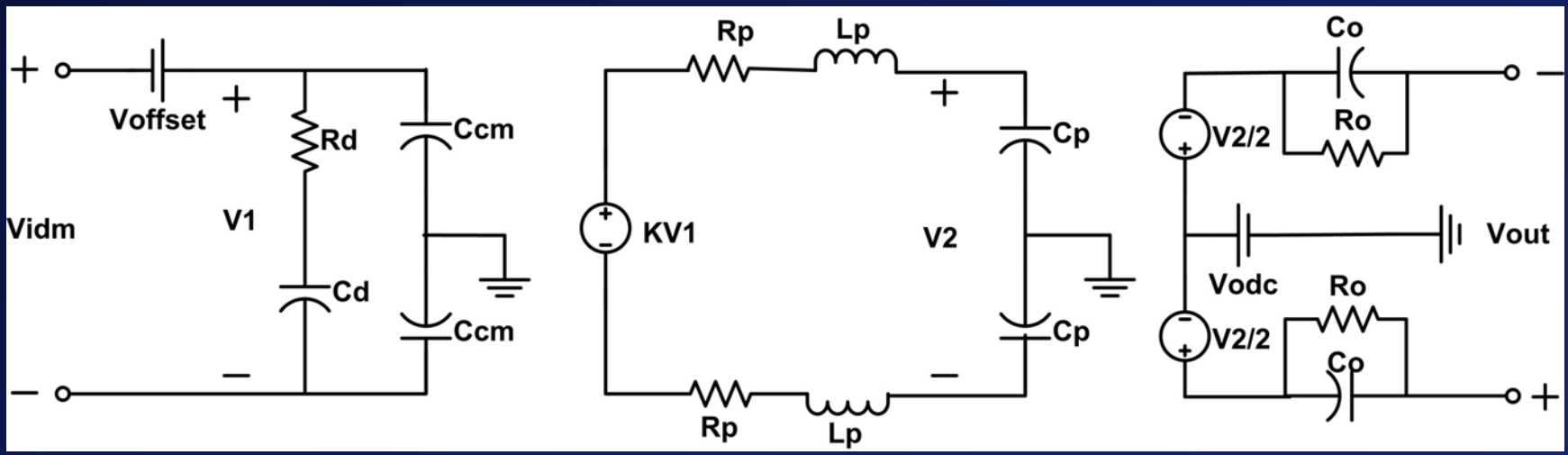
…

S3 = (2*h*C3*Gmo*R3*R5*(2*Gm5*L4*(I40 + 2*Gm4*Vc30) - C5*Vc50) - C3*C5*Co*R3*R5*Vop0 - h*Co*(C3*R3 + C5*R5)*Vop0 + 8*h*h*h*Gm*Gm4*Gm5*Gmo*R3*R4*R5*Vos + h*h*(-2*C5*Gmo*R5*Vc50 - Co*Vop0 + 4*Gm5*Gmo*R5*(I40*L4 + 2*Gm4*R3*(C3*R4*Vc30 + Gm*L4*Vos))))/(h*(h + C3*R3)*(h + C5*R5));

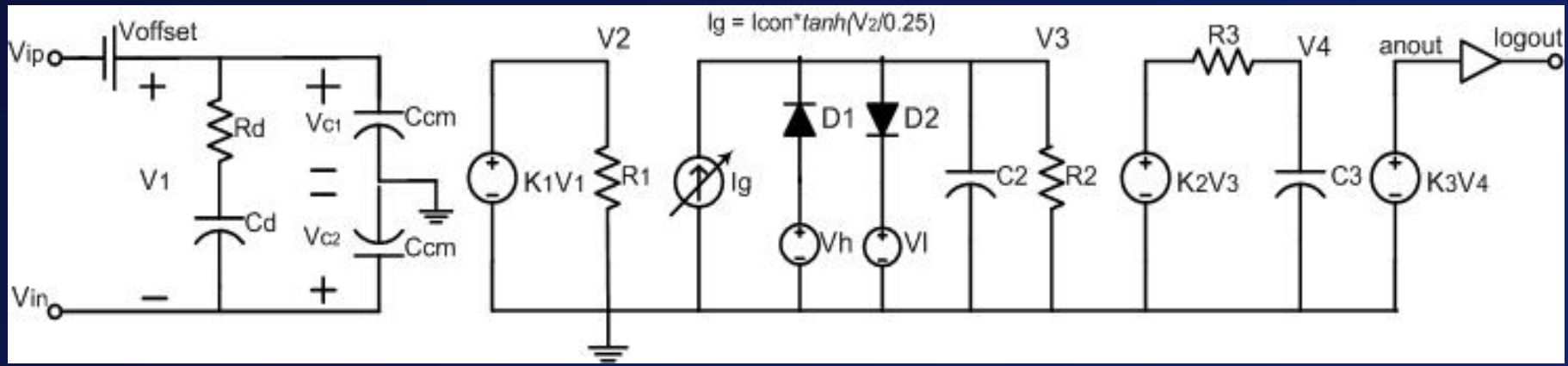
S4 = - ((2*h*C3*Gmo*R3*R5*(2*Gm5*L4*(I40 + 2*Gm4*Vc30) - C5*Vc50) + C3*C5*Co*R3*R5*Von0 + h*Co*(C3*R3 + C5*R5)*Von0 + 8*h*h*h*Gm*Gm4*Gm5*Gmo*R3*R4*R5*Vos + h*h*(-2*C5*Gmo*R5*Vc50 + Co*Von0 + 4*Gm5*Gmo*R5*(I40*L4 + 2*Gm4*R3*(C3*R4*Vc30 + Gm*L4*Vos))))/(h*(h + C3*R3)*(h + C5*R5));

…

Macromodel – OpAmp



Macromodel – Comparator



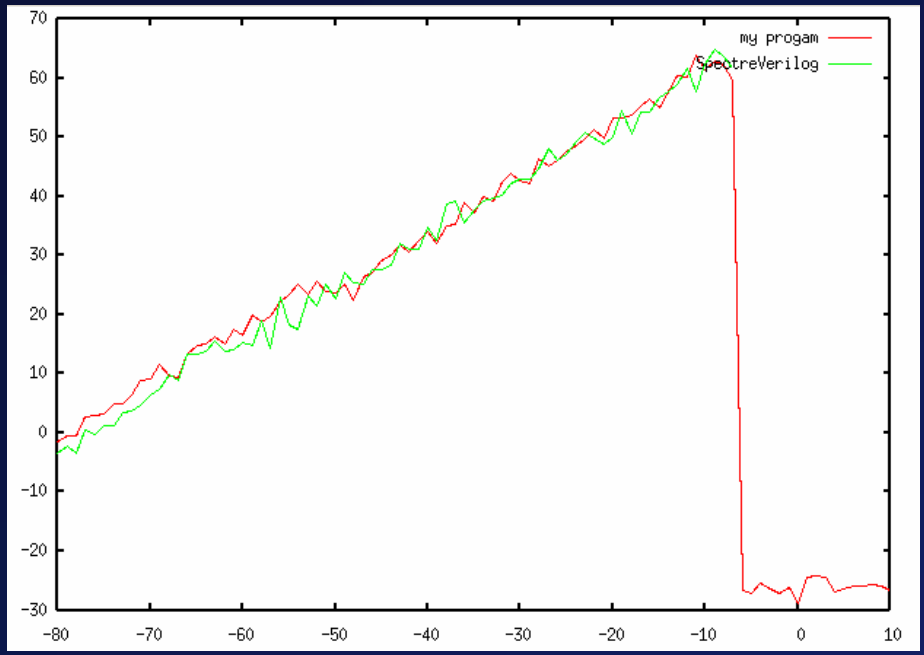
Convergence and Truncation Error Detection

```
for (int i = 0; i < 4; i++)
   unknown[i] = ( F[i][0] * Vip + F[i] [1] * Vin ) +
                ( F[i][2] * If1 + F[i] [3] * If2 ) + S[i];
```

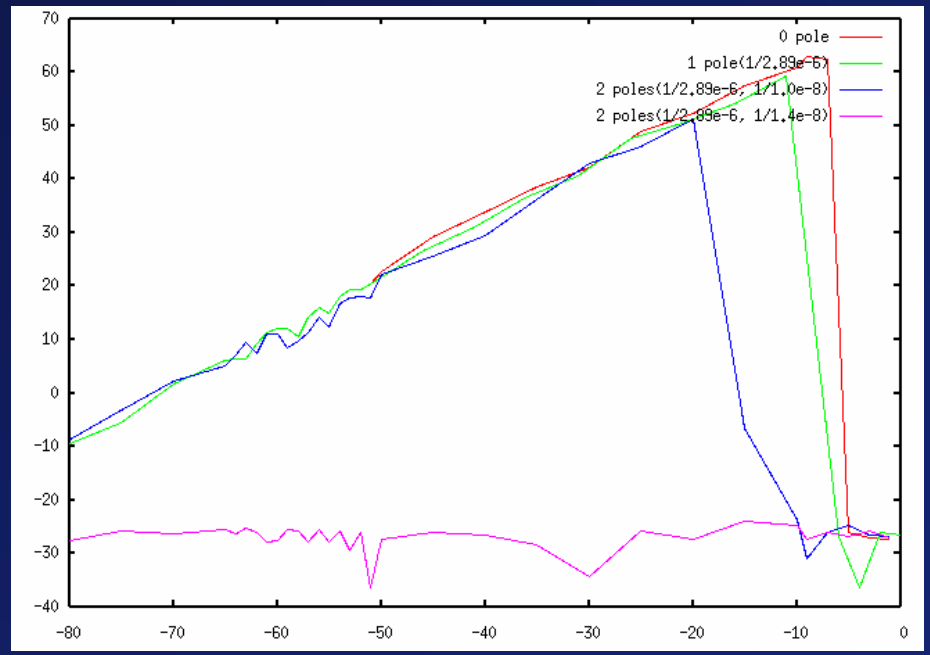
How to detect truncation errors?

How to detect convergence problem?

Simulation Results



Our simulation vs Spectre



Simulation results with OTA macromodel in different abstraction levels

Simulation Results

ΣΔ ADC order	Spectre (s)	Symbolic (s)	Speed-up
1	507.1	3.5	144.88
2	533.9	5.88	90.79
3	852.3	8.24	103.43
4	1284.9	10.69	120.19
5	1752.0	12.91	135.70

Comment: Because of the extreme values of some parameters, we had severe convergency problems in Cadence Mixed-Signal Simulation Environment (Spectre + Verilog XL).

Conclusions

- Fast time-domain simulation method for CT AMS circuit synthesis purpose.
- At the system-level, the method uses symbolic description of the circuit netlist.
- Functional blocks are replaced with their macromodels, which include circuit non-idealities and nonlinear behavior.
- Method is significantly faster than numerical simulation (two orders of magnitude).
- Accuracy vs. speed is tunable.
- The first compiled-code simulator for AMS circuits.
- Simulator code could be optimized to maximize speed and to avoid convergence problem.