A Behavioral Signal Path Modeling Methodology for Qualitative Insight in and Efficient Sizing of CMOS Opamps

Francky Leyn, Walter Daems*, Georges Gielen*, Willy Sansen
Katholieke Universiteit Leuven, Belgium

Abstract

This paper describes a new modeling methodology that allows to derive systematically behavioral signal path models of operational amplifiers. Combined with symbolic simulation, these models provide high qualitative insight in the small-signal functioning of a circuit. The behavioral signal path model provides compact interpretable expressions for the poles and zeros that constitute the signal path. These expressions show which design parameters have dominant influence on the position of a pole/zero and thus enable a designer to control a manual interactive sizing process. The methodology consists of the application of a sequence of abstractions, so that one gradually progresses from a full device to a full behavior circuit representation. During this translation, qualitative insight and design requirements are obtained. The methodology is implemented in an open tool called EF2ef*. The behavioral signal path model is also used for optimization based sizing in order to achieve pole placement in an efficient way. For optimization based sizing, a new strategy for hierarchical penalty function composition is proposed, which allows sequential pruning of the design space. Combined with an operating point driven DC formulation and local minimax optimization, a fast sizing method is obtained which can be used for interactive design space exploration. Experimental results of both modeling and sizing are shown.*

I. Introduction

Modeling an analog circuit is an important step in analog circuit design. Modeling provides insight in the operation of the circuit. This insight can be used to size or synthesize a new circuit and is a prerequisite in order to design successfully, be it manually, be it with an optimizer.

Modeling a circuit in many cases is time consuming. Therefore, reuse of the circuit knowledge is interesting. In order to be reusable, the knowledge must be formalized. Formalization of knowledge in a certain language not only documents the derived model, but also allows one to use the model in a sizing environment. After formalization the model can be interchanged between different designers. This is useful for redesigns and technology migrations.

Different small-signal modeling methodologies exist. The methodologies can be distinguished depending on their accuracy, generation of qualitative insight and time to deduce the model. Traditional techniques like modified nodal analysis (*MNA*) [1] model a circuit exactly but generate no physical insight. Symbolic simulation [2] gives an approximate transfer function and provides additional qualitative insight. In this paper a *behavioral signal path modeling* methodology for analog integrated circuits is presented with as strongest point the generation of qualitative insight.

We are looking for an automated sizing method which is fast enough to allow *interactive design space exploration*. This way the designer is able to explore the different trade-offs present in the design space. In order to achieve this, we need a computationally efficient sizing method. This is achieved using an operating point driven DC formulation, a behavioral signal path model which is also used for pole placement, *sequential design space pruning* and local minimax optimization.

In section 2 the merits of behavioral signal path modeling are discussed. In section 3 the modeling methodology is described. In section 4 the modeling tool EF2ef is presented. In section 5 the use of the behavioral signal path models for optimization based sizing is shown. In section 6 some examples are shown and conclusions are given in section 7.

II. Behavioral signal path modeling

Modified nodal analysis and symbolic simulation both look at the total transfer function of the complete circuit. The different effects that constitute the small-signal behavior are combined together in the total transfer function. From a certain circuit size on, this makes it hard to isolate the different contributions individually. In order to gain real qualitative insight [3] into a circuit and also to perform an efficient sizing (e.g. through pole placement), the different effects should be available separately. One has to work *locally* instead of globally, much like expert designers do. Therefore, knowledge of the location of poles and zeros that constitute the transfer function is the key to good design.

A behavioral signal path model, as depicted in Fig. 1, shows the signal flow with its different poles and zeros. This is a powerful representation for multiple reasons. It shows all the different conversions from current to voltage and vice versa. It shows

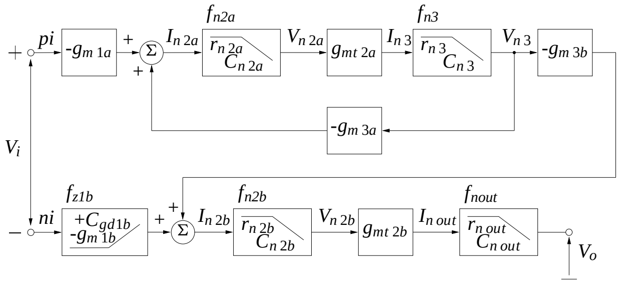


Fig. 1. Behavioral signal path model of a current buffer OTA.

* Walter Daems and Georges Gielen are respectively research assistant and research associate of the Belgian Fund of Scientific Research

the different encountered poles/zeros that cause a de-/increase in transfer along the signal path. A behavioral signal path model provides qualitative insight: regardless of the actual numerical values, the effect on the total transfer function of a shift of a pole or zero can be predicted. The different poles are a function of the small-signal parameters of the different devices and are modeled with compact symbolic equations, which makes them interpretable. The equations can be used for pole placement in both a manual interactive and optimization based sizing scheme. Pole placement is used in order to achieve first order behavior for frequencies up to the gain-bandwidth of the circuit.

In a manual interactive sizing scheme with SPICE or a constraint based sizing scheme with DONALD [4], one has to determine which design parameters must be changed in order to fulfill certain design requirements or specifications. Because the explicit equations of the poles and zeros of the behavioral signal path model are compact, they are fully comprehensible. This enables the designer to control the sizing process. Because he knows which parameters have dominant influence on the position of a pole, he can push the design in a certain direction by placing a pole or zero. The dominant impact of the different design parameters on the various performance parameters is known.

In the case of optimization based sizing, these models allow pole placement by adding penalty terms to the goal function if the poles and zeros occur before the *GBW* of the amplifier. They can be used by equation compilers like DONALD [4] or ASTRX [5]. Since the expressions are compact and explicit, they are more efficient than numerical techniques like *AWE* [6] which typically has a CPU cost of one DC analysis [7] or the QZ-algorithm [8], which is only applicable for circuits of moderate size [9].

Behavioral signal path models allow the derivation of design requirements. If a circuit contains feedback loops (like in Fig. 1), stability requirements must be fulfilled in order to obtain a stable circuit. Unlike other modeling methodologies, behavioral signal path modeling allows one to derive expressions for these design requirements.

III. Incremental modeling methodology

Analog circuit modeling evolves around three orthogonal axes. Modeling thus involves three independent choices:
- circuit representation (device - behavior)
- solution method (*MNA*/symbolic simulation with possible hierarchical extensions)
- formalization language $\leftrightarrow$ target engine (SPICE, VHDL-AMS, DONALD, ASTRX)

Incremental modeling is a methodology that allows one to gradually move on the circuit representation axis from a full device to a full behavior representation. The basic step in incremental modeling is an *abstraction*. An abstraction is a transformation of the circuit, mostly topological, which allows one to simplify the circuit. During such a step, one makes abstraction of certain circuit functionality. An abstraction chain is depicted in Fig. 2. One starts with the reference topology T_R composed out of its devices. If one understands the function of certain parts of it, one makes an abstraction or transformation τ for those parts. The circuit representation becomes mixed device-

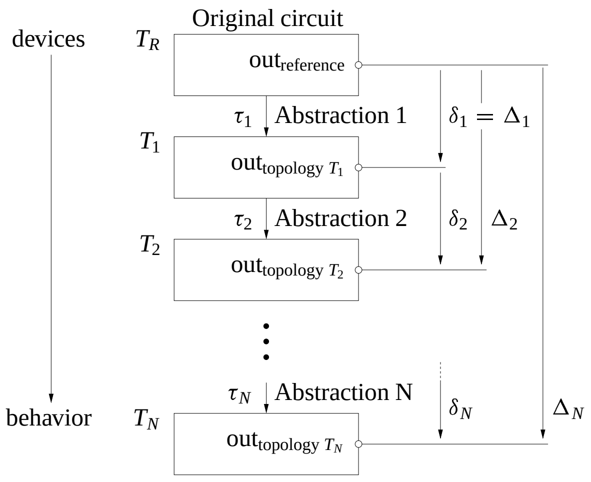


Fig. 2. **Incremental modeling as a chain of abstractions.**

behavior. While making subsequent abstractions τ_i, one gradually progresses from devices to behavior. If one fully understands the circuit, one is thus able to replace the complete circuit by a behavioral equivalent T_N:

$$T_N = (\tau_N \circ \ldots \circ \tau_2 \circ \tau_1) T_R = \left(\bigcirc_{j=1}^{N} \tau_j \right) T_R$$

We are interested in a small-signal variable S like the magnitude or phase of the transfer along the signal path. This variable $S(T, Q, f)$ is function of the topology T, the operating point Q and frequency f. Each abstraction provides additional qualitative insight. The price paid for this is loss of accuracy. Two error measures can be considered. The *incremental error* δ_i gives the error introduced by the considered abstraction τ_i. It is the error between topology T_i and the previous topology T_{i-1}:

$$\delta_i S(Q, f) = S(T_i, Q, f) - S(T_{i-1}, Q, f)$$

The *total error* Δ_i is the error between topology T_i and the original reference circuit T_R. It is the error introduced by all previous abstractions together; the accumulation of the incremental errors:

$$\Delta_i S(Q, f) = S(T_i, Q, f) - S(T_R, Q, f)$$

These errors are determined over a considered frequency range $[f_l, f_u]$ and for a certain (set of) operating point(s) $V_Q = \{Q_j | j\}$. Good models have a small deviation in multiple operating points. Good models are *global models* that track the operating point. For optimization based sizing one needs global models since the optimizer will propose operating points spread over the complete design space. Behavioral signal path modeling generates such models, since each significant effect in the signal path is modeled. The goal of behavioral signal path modeling is thus to gain as much qualitative understanding without losing too much accuracy.

Some examples of abstractions:
- Replace a floating admittance by its Y-parameter equivalent as depicted in Fig. 3. Although the number of components increases, the state-space representation remains unchanged. Each

component of the Y-parameter equivalent corresponds to a single entry of the *MNA*-stamp because they are connected to ground, which is the *MNA* datum node. This abstraction is powerfull because it allows one to isolate the effects between different nodes by explicit components. This way, it is always possible to obtain a signal flow diagram of a circuit. In many cases, the admittances of the Y-parameter equivalent can be lumped together with other circuit elements.

• Replace a floating transconductance by components that represent the entries of the *MNA*-stamp.

• Remove components of a Y-parameter equivalent. This is equivalent to removing *MNA* entries in the state-space representation. The Sherman-Morrison criterion allows efficient reduction of the state-space matrix [10].

• Replace parts of a signal path by a behavioral pole/zero equivalent. For poles this is mostly a topology where a RC-tank is isolated between a node and ground and where a transconductance pumps current onto that node. For zeros this is mostly the lumping of a transconductance g_m and a transcapacitance c_m. The two forms are depicted in Fig. 4. The expressions for the corner frequency of these building blocks are compact and highlight the dominant impact of certain design parameters.

• Replace a coupling capacitor C by its Y-parameter equivalent. The floating admittance $Y = sC$ is replaced by two grounded capacitances C and two grounded transcapacitances $c_m = C$. The transcapacitances mostly occur in parallel with a transconductance which allows one to replace them by a behavioral zero stage.

• Replace a generic one-stage amplifier as depicted in Fig. 5 by a behavioral second-order system and a positive zero. By replacing the coupling capacitor C_c by its Y-parameter equivalent and replacing the RC-tanks and transconductance-transcapacitance pair by their behavioral equivalent, the behavioral signal path model depicted in Fig. 5 is obtained. In the case of Miller compensation, coupling between in- and output is required. A transfer as depicted in Fig. 5, where the inverse of the feedback $|1/H|$ intersects the forward transfer $|G|$, resulting in pole splitting, is required. The design requirement $1/(2\pi f_{d\,ol} C_c) < |A_{R0}|$ and explicit expressions for the closed loop poles $f_{d\,cl}$ and $f_{nd\,cl}$ are

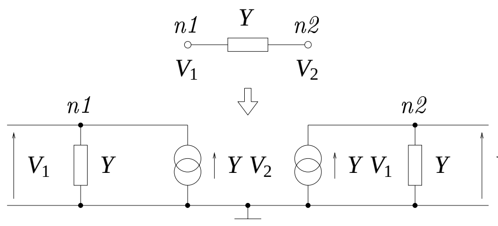
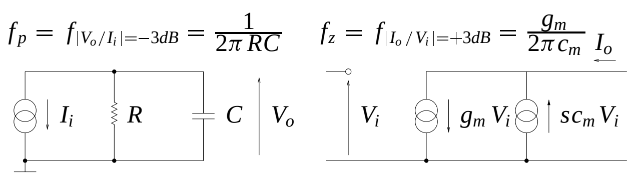
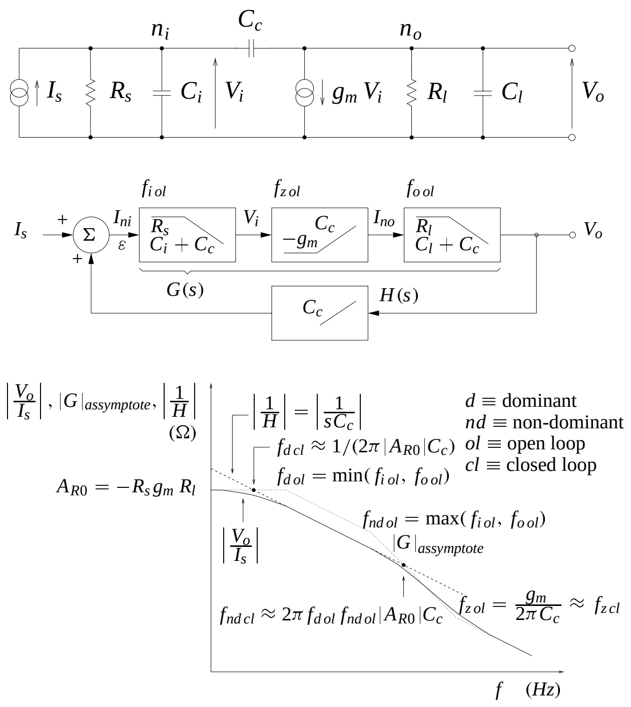


Fig. 3. Y**-parameter equivalent of a floating admittance.**



$f_p = f_{|V_o/I_i|=-3dB} = \frac{1}{2\pi RC}$
$f_z = f_{|I_o/V_i|=+3dB} = \frac{g_m}{2\pi c_m}$
I_i
R
C
V_o
I_o
V_i
$g_m V_i$
$s c_m V_i$

Fig. 4. **Topologies that allow easy determination of behavioral poles/zeros.**



n_i
C_c
n_o
I_s
R_s
C_i
V_i
$g_m V_i$
R_l
C_l
V_o
$f_{i\,ol}$
$f_{z\,ol}$
$f_{o\,ol}$
I_s
Σ
I_{ni}
ε
$\frac{R_s}{C_i + C_c}$
V_i
$\frac{C_c}{-g_m}$
I_{no}
$\frac{R_l}{C_l + C_c}$
V_o
$G(s)$
$H(s)$
C_c
$\left|\frac{V_o}{I_s}\right|, |G|_{assymptote}, \left|\frac{1}{H}\right|$ (Ω)
$\left|\frac{1}{H}\right| = \left|\frac{1}{sC_c}\right|$
$f_{d\,cl} \approx 1/(2\pi|A_{R0}|C_c)$
$f_{d\,ol} = \min(f_{i\,ol}, f_{o\,ol})$
$A_{R0} = -R_s g_m R_l$
$f_{nd\,ol} = \max(f_{i\,ol}, f_{o\,ol})$
$|G|_{assymptote}$
$\left|\frac{V_o}{I_s}\right|$
$f_{nd\,cl} \approx 2\pi f_{d\,ol} f_{nd\,ol} |A_{R0}| C_c$
$f_{z\,ol} = \frac{g_m}{2\pi C_c} \approx f_{z\,cl}$
$d \equiv$ dominant
$nd \equiv$ non-dominant
$ol \equiv$ open loop
$cl \equiv$ closed loop
f (Hz)

Fig. 5. **Schematic of a generic one-stage amplifier, its behavioral signal path equivalent and the transfer function in the case of Miller-effect.**

obtained.

• Substitute an input transistor with a transconductance amplifier with an associated positive zero.

• Replace dynamic cascode loads with a behavioral equivalent output impedance. This abstraction introduces hierarchy into the model. The equivalent output impedance can be determined exactly or approximately. For a MOS cascode stage consisting out of a bias transistor M_b and a cascode transistor M_c, an approximate but accurate equivalent impedance is given by $r_{eq} = r^*_{ob} + r_{oc}(1 + g_{mc} r^*_{ob}) + r_{Dc}$ with $r^*_{ob} = (r_{Sb} \parallel r_{BSb} + r_{ob}) \parallel r_{BDc} + r_{Db} + r_{Sc}$ in parallel with $c_{eq} = C_{GDc} + C_{BDc}$.

• Short a branch between two nodes and delete the elements between these nodes. This abstraction is mostly applied for series resistors like r_S, r_D, r_B, r_C and r_E. This abstraction reduces the number of nodes. The size of the *MNA*-matrix is reduced, which results in shorter execution times.

• Remove certain elements with negligible influence on the transfer function. This is mostly done for parasitics like C_μ and r_μ. Parasitics like C_{GB}, r_{BD} and r_{BS} mostly can be lumped with other components, which makes their removal unnecessary.

• Reconnect small-signal components.

• Ground biasing nodes and remove biasing transistors. This abstraction assumes that the biasing nets are perfectly decoupled.

• Ground the tail node of a differential pair. This is in fact a sequence of three abstractions. In the first abstraction, one replaces the current source by its equivalent output impedance. In the second abstraction, this impedance is assumed to be infinite. In the third abstraction, the branch impedance seen from the tail node is assumed to be equal for both branches. In that case the tail node is virtually grounded for differential signals.

• Replace a current mirror by a behavioral signal path equivalent.

In the context of symbolic simulation, the different abstractions can be seen as *simplification before generation* [11] steps: the topology is simplified before the coefficients of the closed form transfer function $TF(s) = \sum_i a_i s^i / \sum_j b_j s^j$ are generated. The obtained expressions are more compact, which improves the interpretability. By combining both behavioral signal path modeling and symbolic simulation, high qualitative insight is obtained.

The introduction of an abstraction is mostly driven by recognition of well known building blocks or specific interconnection patterns. This allows one to build an expert system based on the logic programming paradigm that suggests the designer which abstractions could be used for which parts of the circuit. The expert system contains rules for *building block recognition* [12]. These rules also allow to determine the nodes of the signal path. This information can then be used to associate an abstraction with a certain building block. A PROLOG example that detects the abstraction for a cascode bias stage is given below. The cascode bias abstraction results in the removal of the current source transistors and the grounding of the tail node of the differential pair.

```
diff_pair(X,Y,N) :- source(X,N), source(Y,N),
  gate(X,G1),  gate(X,G2),  G1 \== G2,
  drain(X,D1), drain(X,D2), D1 \== D2.
cascode_bias(X,Y,N) :- drain(X,Nc), source(Y,Nc)
  gate(X,Gx), bias(Gx), gate(Y,Gy), bias(Gy),
  source(X,Sx), power(Sx),
  drain(Y,Dy), diff_pair(K,L,Dy).
abstraction(cascode_bias, [X,Y,N]) :-
  cascode_bias(X,Y,N).
```

Incremental modeling can be automated with algorithm 1. As input we take a set of operating points $V_Q = \{Q_j | j\}$ in order to check the generality of the model. In step 1, the functionality of the circuit components is determined. In step 2 and 3, ideal biasing is assumed. In step 6 the series resistors are removed. Although these floating elements could be replaced by their Y-parameter equivalent, this gives rise to highly interconnected signal flow diagrams, which are rarely interpretable. This step is justified because the error introduced by removing these elements is low. In the cases where these elements have significant influence on the transfer (e.g. output stages), they mostly can be incorporated into the equivalent output impedance of that stage (step 4). In step 7, each entry of the *MNA* stamp of a floating circuit element is replaced by an equivalent circuit element which always has one grounded node. In step 8, lumping is performed, resulting in a reduction of the number of circuit elements. In step 9, the parasitics are removed one after another, with as criterion the introduced total error if they were removed.

If a behavioral signal path model is not required, the algorithm can be used as a model reduction technique (simplification before generation in the context of symbolic simulation). In step 2 and 3 the bias circuitry is then replaced by its small-signal equivalent. Step 6 can be postponed and transfered to step 9.

Incremental modeling allows one to transform small-signal modeling into control theory modeling. This gives the modeling methodology a strong foundation. Incremental modeling is a suitable methodology to go from full device to full behavior.

Algorithm 1: Automated incremental modeling

```
      input : maximal total error $\Delta_{max}S$ for frequency range $[f_l, f_u]$
              set of operating points $V_Q = \{Q_j | j\}$
              reference topology $T_R$ with transistor instantiations
1     building block recognition and signal path detection
2     ground tail nodes and remove current source transistors
3     ground biasing nets and remove biasing transistors
4     replace cascode loads with an equivalent output impedance
5     small-signal expansion of transistor models
6     removal of series resistors
7     replace floating elements by their MNA circuit equivalent
8     lump resistors, capacitors and transconductances
9     do forever
        begin
10        Determine for all remaining non-lumped parasitics $P_i$
            $\Delta_i^* S = \max_{Q \in V_Q} \max_{f \in [f_l, f_u]} |\Delta_i S(Q, f)|$
11        Determine $\Delta^* S = \min_i \Delta_i^* S$
12        if $\Delta^* S \leq \Delta_{max} S$
            then remove all parasitics $P_i$ for which $\Delta_i^* S = \Delta^* S$
            else exit
        end
```

For textbook designs, going from full device to full behavior in one step still may be feasible. However, from a certain circuit complexity on, this step is too big and must be split up. Incremental modeling is a systematic modeling method in the sense that it guides the efforts to go from full device to full behavior. One knows the error introduced by each abstraction, which allows one to detect erroneous abstractions, introduced by reasoning faults.

IV. Open modeling tool EF2ef

The modeling methodology described above is implemented in an open tool called EF2ef (Electronic Format t(w)o Electronic Format). It has a *pipelined architecture with a generic backbone*. Its software architecture is depicted in Fig. 6. Starting point is a netlist in a certain hardware description language. With a markup processor dedicated for that hardware language, the netlist information is converted to the generic EF2ef internal tool format, which is a structured text format. On the internal tool format, all kind of abstractions can be performed. Each abstraction is implemented as a text filter and changes the pipeline text stream. At the end of the pipeline, the state-space representation of the circuit is solved with modified nodal analysis [1] or symbolic simulation [2]. Both are implemented as filters. The formalization of the resulting analytical model is done with a back end dedicated for the desired formalization language.

The software architecture of EF2ef is non-monolithic. It consists of a library of filters. The user can reconfigure the

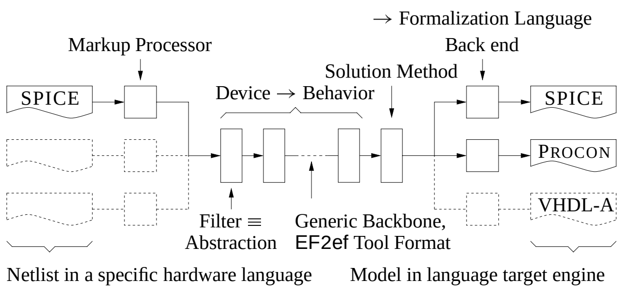


Fig. 6. Software architecture of EF2ef**.**

pipeline to his own needs if necessary. This way, the flexibility required for analog circuit modeling is obtained. Although we feel the EF2ef library is quite complete, the situation may occur where an user wants to add a new abstraction. Addition of new filters is simplified because the programming language to implement a filter can be choosen freely and because the used data structure is text which is easy accesible. The system is thus extendible, which makes it an open system.

V. Optimization based sizing

OTA's and opamps must have first order behavior up to the gain-bandwidth. In order to obtain this, the different nondominant poles and positive zeros in the signal path must occur at frequencies beyond the gain-bandwidth. Imposing these design requirements is called *pole placement*.

A quantity expressing the first order behavior is the phase margin. Out of the phase margin, one can determine if nondominant poles occur before the unity gain frequency f_u. The phase margin PM could thus be used for pole placement. However, a pole or zero only has an influence on the phase in a region of one decade before and after its corner frequency (corresponding to one decade I_{DS} and/or V_{GS}-V_T). This implies that if a pole or zero falls outside the region between one decade before and one decade after the unity gain frequency f_u, its influence on the phase margin PM is negligible. This makes that a local optimizer, guided by the decrease in goal function, has insufficient notion about what direction to move, because the phase sensitivity is too low. Only a computationally expensive global optimization algorithm is able to find the "hole" in the design space where the nondominant poles and zeros are placed correctly.

So, how can we achieve pole placement using an efficient local optimizer? One can determine the poles and zeros of the total transfer function. This can be done in several ways. One can use polynomial interpolation [8] in order to determine the coefficients of the closed form transfer function $TF(s) = \sum_i a_i s^i / \sum_j b_j s^j$. Out of the numerator and denominator polynomials, the zeros respectively poles can be determined using an iterative root solving algorithm. Another method is to apply the iterative QZ-algorithm [8] on the system matrix equation, generating the system poles and zeros in a direct way. Both methods are iterative, CPU intensive, provide weak guarantee concerning convergence and are only applicable for circuits of moderate size [9]. Another method is asymptotic waveform evaluation *AWE* [6]. This method delivers a reduced complexity model with AC circuit response below a specifiable limit. However, the algorithm requires the iterative solving of a set of simultaneous nonlinear equations, or a more direct iterative root solving [6], which both introduce weak convergence guarantee and moreover typically have a CPU cost of one DC analysis [7]. Since all the previous mentioned techniques are numerical of nature, they provide no qualitative insight in the small-signal functioning of a circuit.

The poles and zeros in the behavioral signal path model are local and determine the global poles and zeros of the total transfer function. If the local nondominant poles and zeros occur beyond the gain-bandwith GBW of the circuit, the global poles and zeros of the total transfer function will also do so. Given the algorithm for automated incremental modeling, one is able to derive *explicit, compact, iteration free* equations for the poles and zeros that constitute the transfer function. These numerical efficient expressions allow pole placement in a very efficient way. They allow to generate penalty functions which are able to guide a local optimizer, even if the local poles and zeros occur decades beyond or below the unity gain frequency f_u. If a global optimizer is used, increased optimization speed can be expected.

By using an *operating point driven DC formulation* [13], one obtains a robust design plan, that is always DC consistent and has minimal dimension. Possible convergence problems as in the case of simulation based sizing are lacking since one only has to compute a set of one-dimensional root solving problems. The DC problem is solved in the design plan itself, instead of transfering it to the optimizer as is done in a relaxed DC formulation [14]. This way, the optimization problem has minimal dimensionality and the obtained optimum is guaranteed to be DC consistent. The solvability space Ω_S of the design plan is identical to the theoretical solvability space $\{W_E, L_E, I_{DS}, V_{DS}, V_{GS} - V_T \in \mathbb{R}_0^+\}$. The starting point is obtained directly by specifying the operating point.

The solvability space Ω_S can be subdivided into other spaces as depicted in Fig. 7. The manufacturability space Ω_M is the set of circuits that can be produced with a given technology. The operationality space Ω_O contains all circuits that have proper DC biasing. The transistors of a current source or current mirror must be biased in saturation regime in order to operate properly. Transistors used as resistors for common mode feedback must be biased in the linear region. The functionality space Ω_F contains all designs that provide the functionality they are intended for. For an OTA this is first order behavior up to the gain-bandwith. In order to achieve this, design requirements must be fulfilled. The applicability space Ω_A contains designs that fulfill the given specifications. If all specifications are fulfilled, there is still room left to make a trade-off between different performance parameters. These designs are typically situated at the edge of the different subspaces of the manufacturability space. They contain minimal length devices and transistors biased near the edge of the saturation/linear operation region. The design requirements are just fulfilled and some specifications are just met.

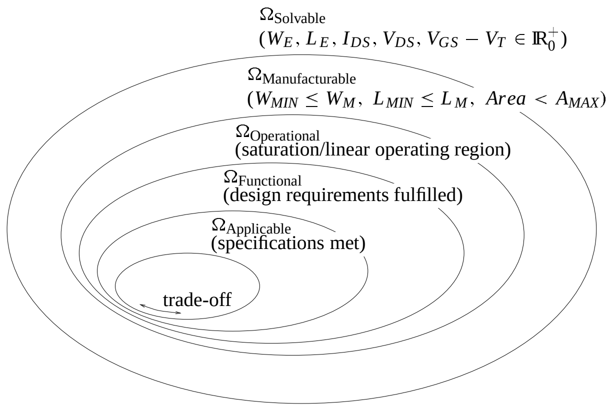

Ω_{Solvable}
$(W_E, L_E, I_{DS}, V_{DS}, V_{GS} - V_T \in \mathbb{R}_0^+)$
$\Omega_{\text{Manufacturable}}$
$(W_{MIN} \le W_M,\ L_{MIN} \le L_M,\ Area < A_{MAX})$
$\Omega_{\text{Operational}}$
(saturation/linear operating region)
$\Omega_{\text{Functional}}$
(design requirements fulfilled)
$\Omega_{\text{Applicable}}$
(specifications met)
trade-off

Fig. 7. Spaces encountered in optimization based sizing of analog circuits.

With each of these spaces, one can associate inequality functions. These on their turn can be used to generate penalty functions. The subdivision of the manufacturability is now used in a *hierarchical penalty function composition strategy*. The trade-off function gets the reference weight W_R. The penalties of the enclosing applicability space Ω_A get weights of $10W_R$. The weights of the other enclosing spaces are 10^2W_R for Ω_F, 10^3W_R for Ω_O and 10^4W_R for Ω_M. This way, the different design problems are tackled only when it makes sense to do so. The same divide and conquer strategy is used by designers when designing manually. First, one ensures that the circuit has a proper operating point. It makes no sense to fulfill the design requirements if a circuit is badly biased. Once the design requirements are fulfilled, one tries to achieve the specifications. During an optimization one thus sequentially prunes the design space.

The trade-off function $\zeta(X)$ which is function of the independent optimization variables X, allows one to interchange different performance parameters $p_i(X)$ against one another using steering factors β_i:

$$\zeta(X) = W_R \sum_i \beta_i p_i(X)$$

An inequality function has negative or zero value if the corresponding condition is fulfilled and positive if it isn't. The inequality $f_i < ub$ is converted into the inequality function $h_i = W(f_i - ub)$ where W represents the weighting factor. The inequality $lb < f_i$ corresponds to the inequality function $h_i = W(lb - f_i)$. The trade-off function $\zeta(X)$ is now combined with the inequality functions h_j:

$$\varphi(X) = \max_j \theta_j(X) \ with \ \begin{cases} \theta_0 = \zeta(X) & \text{trade-off in } \Omega_{Applicable} \\ \theta_j = \zeta(X) + W_j h_j(X), & j \neq 0 \end{cases}$$

The optimal design solution is found at the minimum of this function. An optimal design therefore fulfills the following *minimax* condition:

$$\varphi^* = \min_X \varphi(X) = \min_X \max_j \theta_j(X) = \varphi(X^*)$$

where X^* represents the optimal vector of independent design variables which result in an optimal design in Ω_A.

In each point X only one function $\theta_j(X)$ determines the maximum function. This has the advantage that only one function determines the convexity and/or presence of local optima. The more functions are summed, the more correlation is introduced and thus the more risk that convexity is absent and the higher the chance for local optima. For the minimax optimization, the algorithm of Dem'yanov and Malozemov [15] is used.

Scaling of the independent input variables, performance variables and inequality functions is essential in order to obtain good conditioned optimization problems. The purpose of *scaling* is to linearize strongly nonlinear functions to a maximum extent, in order to ease and accelerate the optimization process. Almost all encountered equations in analog circuit design have logarithmic behavior. For the device equations, $\log I_{DS}$ depends in an asymptotically linear way on $\log V_{GS}$ and $\log V_{DS}$. The same is valid for the small-signal parameters. The expressions for poles also have a logarithmic nature (bode/pole-zero plot). Logarithmic transformations are therefore mostly the appropriate scaling method.

VI. Examples

The modeling methodology described above has been tested on three circuits: a pMOS Miller OTA, a current buffer OTA [16] and a class-AB amplifier [17]. The current buffer OTA is depicted in Fig. 8. The first abstraction is the grounding of the biasing nets *cmrpp*, *cmrp* and *cmrn*. Next the tail node *n1* is grounded and the bias transistors M6 and M7 are removed. Then the cascode loads M4a-M5a and M4b-M5b are replaced by their equivalent impedance $r_{eq} = r_{o4} + r_{o5}(1 + g_{m5}r_{o4}) \parallel c_{eq} = C_{GD} + C_{BD}$. The capacitors C_{GD1a}, C_{GD1b}, C_{GD3a} and C_{GD3b} are replaced by their Y-parameter equivalent. The transconductance $g_{mt\,2a} = g_{m\,2a} + g_{mb\,2a}$ is replaced by a transconductance between *n3* and ground and an equivalent resistor $r = 1/g_{mt\,2a}$ is placed between *n2a* and ground. The same procedure is followed for $g_{mt\,2b}$. Next, $r_{o\,2a}$ is removed and replaced by a resistor between *n3* and ground with value $r_{o\,1a3a} + r_{o\,2a}(1 + g_{mt\,2a}r_{o\,1a3a})$ with $r_{o\,1a3a} = r_{o\,1a} \parallel r_{o\,3a}$. The same is done for $r_{o\,2b}$. Next, the transcapacitance $c_{m\,n2b\rightarrow n3}$ pumping current on node *n3* and controlled by $V_{n\,2b}$ is removed. After a lumping step the circuit is now reduced to a form where RC-tanks are isolated between a node and ground, which allows one to replace them by a behavioral equivalent. The transconductance-transcapacitance pairs are also replaced by their behavioral equivalent. Now we can make abstractions at the behavioral level. The zeros of the positive signal path become active decades later than the cut-off frequency of the positive signal path, regardless of the operating point Q. This allows one to remove them. Finally, the behavioral signal path model depicted in Fig. 1 is obtained. During the modeling, the number of circuit elements is reduced from 75 to 17.

The loop $I_{n\,2a}$-$V_{n\,2a}$-$I_{n\,3}$-$V_{n\,3}$-$I_{n\,2a}$ (Fig. 1) forms a second order system with damping ratio $\zeta = (f_{n2a} + f_{n3})/(2f_n)$ and natural frequency $f_n = \sqrt{f_{n2a} f_{n3}(1 + r_{n\,2a}g_{mt\,2a}r_{n\,3}g_{n3a})}$. In order to avoid ringing on node *n3*, the design requirement $3GBW_{2a} = 3r_{n\,2a}g_{mt\,2a}r_{n\,3}g_{m\,3a}f_{n3} < f_{n2a}$ must be fulfilled.

The computationally efficient behavioral signal path model of Fig. 1 was derived with EF2ef. Together with an efficient operating point driven DC formulation [13] and some manually derived equations for offset, slew rate and output range, the goal function was defined using the hierarchical penalty func-

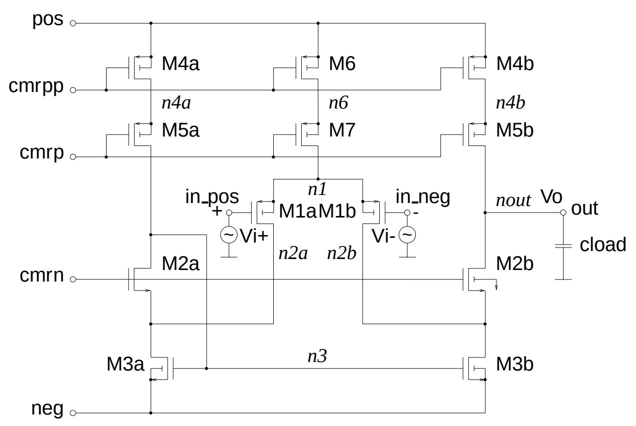


Fig. 8. **Schematic of a current buffer OTA.**

tion composition method described above. Combined with local minimax optimization, a very efficient optimization based sizing method is obtained. All these models were formalized into PROCON, the language used by the contraint programming tool DONALD [18]. The DONALD tool was used to generate a minimal cost computational path [18], which was then dumped to C. This code was then loaded by the local minimax optimization engine MINIMAN [13][15]. The optimum is obtained typically after 100 iterations.

In Table I, a set of specifications is given. Also given are the obtained performances and frequencies of the poles and zeros along the signal path when the design requirement $3GBW_{2a} < f_{n2a}$ is taken into account. The technology used is MIETEC CMOS $0\mu7$. The cut-off frequency GBW_{2a}=458MHz of the positive signal path determines the second pole of the total transfer function. The transfer function gets an additional decay of 20dB up to twice that frequency (916MHz), where a negative zero occurs. From that zero on, the output signal is halved and only the negative signal path determines the output signal. At the frequency f_{n2b}=1.38GHz, the decay in the negative signal path is reflected in another total transfer function pole. The positive zero f_{z1b} =3.42GHz also occurs in the total transfer function. Since the presence of a pole or zero is felt even a decade earlier, the phase margin PM =60°at the unity gain frequency f_u =251MHz is thus determined by the GBW_{2a}, f_{n2b}, the negative zero at 2GBW_{2a} and the positive zero f_{z1b}. The pole f_{n2a} of the internal loop is pushed against the $3GBW_{2a}$ border, resulting in a critically damped feedback loop. In Fig. 9, the incremental error on $|V_o/V_i|$ in function of the frequency is shown for the different abstractions. The errors are relatively small. Morever, the error peaks occur at frequencies beyond the unity gain frequency of 251MHz.

If the stability requirement is disregarded, the optimizer returns a design for which the design requirement $3GBW_{2a} = 8.78GHz \not< f_{n2a} = 1.09GHz$ is violated. The second order system has complex conjugated poles with a natural frequency f_n=1.38GHz and a damping ratio $\zeta = 0.396$. At the frequency f_n=1.38GHz, the total transfer gets an additional decay of 40dB/dec. At $\sqrt{2} f_n$=1.96GHz, the signal is halved and two negative zeros occur. An additional pole and positive zero in the total transfer occur at respectively f_{n2b}=1.09GHz and f_{z1b} =2.93MHz. All these poles and zeros determine the phase margin PM =60°at f_u =333MHz.

spec	value	required
I_{tot}	$3mA$	$\leq 3mA$
PM	60°	$\geq 60^\circ$
A_{v0}	$90.9dB$	$\geq 60dB$
V_{off}	$2.58mV$	$\leq 5mV$
SR	$162V/\mu s$	$\geq 150V/\mu s$
OR	1.5V	$\geq \pm 1.5V$
f_u	$251MHz$	max
$Area$	$0.122mm^2$	

$3GBW_{2a}$	$\leq f_{n2a}$	$\not\leq f_{n2a}$
f_u	$251MHz$	$333MHz$
GBW	$271MHz$	$349MHz$
$3GBW$	$814MHz$	$1.05GHz$
f_{z1b}	$3.42GHz$	$2.93GHz$
$f_{n2a,2b}$	$1.38GHz$	$1.09GHz$
GBW_{2a}	$458MHz$	$1.74GHz$
$3GBW_{2a}$	$1.38GHz$	$8.78GHz$
f_{n3}	$30.0kHz$	$67.6kHz$
f_{nout}	$7.77kHz$	$9.61kHz$

TABLE I

Specifications and obtained performance if the design requirement $3GBW_{n2} \leq f_{n2a}$ is fulfilled. Frequencies for a design where the internal stability requirement is fulfilled, respectively not fulfilled.

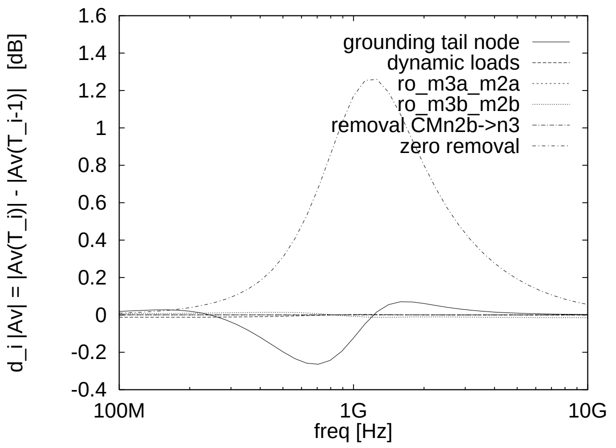


Fig. 9. Incremental error in function of the frequency for the different abstractions.

Modeling a class-AB opamp like the Hogervorst opamp [17] depicted in Fig. 11 is complicated by the presence of source and sink currents during large signal operation. The first abstraction therefore is to model transient behavior in the time domain. We assume quasi-stationary operation for all capacitors, except for the load capacitance $C_{n\,out}$. We consider the negative zero transition, where the sink current is maximal. This is modeled by adding a current source at the ouput node in the DC schematic. Due to the sink current, class-AB operation occurs, which results in the cut-off of M_{11b}. The transistors M_{8b}, M_{9b} and M_{10b} form a double cascode. The behavioral signal path model for this situation is depicted in Fig. 10. During the modeling, the number of circuit elements is reduced from 135 to 36.

VII. Conclusions

An incremental modeling methodology for the generation of behavioral signal path models has been presented. It offers the following advantages:

• High qualitative insight in a circuit is obtained that can be used for manual interactive sizing with SPICE, constraint based sizing with DONALD or optimization based sizing.
• Qualitative insight is generated locally, not at the global circuit level; the different effects are available individually.
• Qualitative understanding is generated by going from devices to behavior. By also applying symbolic simulation, additional qualitative insight is obtained.
• The poles and zeros that compose the transfer function are derived, not the poles and zeros of the total transfer function. These poles are directly available for pole placement.
• The three orthogonal axes encountered in analog circuit modeling are covered by the open modeling tool EF2ef.

For optimization based sizing, a strategy for hierarchical penalty function composition that allows sequential pruning of the design space was proposed. Together with an operating point driven DC formulation and local minimax optimization, an efficient sizing method is obtained, which allows interactive design space exploration.

Acknowledgment

Support from Robert Bosch GmbH is acknowledged.

References

[1] Chung-Wen Ho, Albert. E. Ruehli, and Pierce A. Brennan. The modified nodal approach to network analysis. *IEEE Transactions on Circuits and Systems*, 22(6):504–509, June 1975.

[2] Georges Gielen, Piet Wambacq, and Willy M. C. Sansen. Symbolic analysis methods and applications for analog circuits: a tutorial overview. *Proceedings of the IEEE*, 82(2):287–304, February 1990.

[3] Brian C. Williams and Johan de Kleer. Qualitative reasoning about physical systems: a return to roots. *Artificial Intelligence*, 51(1-3):1–9, October 1991.

[4] K. Swings and W. Sansen. Donald: a workbench for interactive design space exploration and sizing of analog circuits. In *Proceedings EDAC*, pages 475–479. IEEE, February 1991.

[5] E. Ochotta, L. Carley, and R. Rutenbar. ASTRX/OBLX: tools for rapid synthesis of high-performance analog circuits. In *Proceedings ACM/IEEE DAC*, pages 24–30. IEEE, 1994.

[6] Lawrence T. Pillage and Ronald A. Rohrer. Asymptotic waveform evaluation for timing analysis. *IEEE Transactions on Computer-Aided Design*, 9(4):352–366, April 1990.

[7] Eli Chiprout and Michel S. Nakhla. *Asymptotic waveform evaluation and moment matching for interconnect analysis*. Kluwer Academic Publishers, Massachusetts, 1994.

[8] Jiri Vlach and Kishore Singhal. *Computer methods for circuit analysis and design*. Van Nostrand Reinhold, New York, 1983.

[9] X. Huang. *Padè approximation of linear(ized) circuit responses*. PhD thesis, Carnegie Mellon University, November 1990.

[10] Ralf Sommer, Eckhard Hennig, Guido Dröge, and Ernest-Helmut Horneber. Equation-based symbolic approximation by matrix reduction with quantitative error prediction. *Alta Frequenza - Rivista di elettronica*, 5(6):317–325, November 1993.

[11] Q. Yu and C. Sechen. Approximate symbolic analysis of large analog integrated circuits. In *Proceedings IEEE/ACM ICCAD*, pages 664–671. IEEE/ACM, 1994.

[12] Leon Sterling and Ehud Shapiro. *The art of Prolog: advanced programming techniques*. The MIT Press, Massachusetts, 1994.

[13] Francky Leyn, Walter Daems, Georges Gielen, and Willy Sansen. Analog circuit sizing with constraint programming modeling and minimax optimization. In *Proceedings ISCAS*. IEEE, 1997.

[14] P. C. Maulik and L. R. Carley. Automating analog circuit design using constrained optimization techniques. In *Proceedings ICCAD*. IEEE, November 1993.

[15] V. F. Dem'yanov and V. N. Malozemov. *Introduction to minimax*. John Wiley & Sons Ltd., New York, 1974.

[16] John A. Fisher and Rudolf Koch. A highly linear CMOS buffer amplifier. *IEEE Journal of Solid-State Circuits*, 22(3):330–334, June 1987.

[17] Ron Hogervorst, John P. Tero, Ruud G. H. Eschauzier, and Johan H. Huijsing. A compact power-efficient 3V CMOS rail-to-rail input/output operational amplifier for VLSI cell libraries. *IEEE Journal of Solid-State Circuits*, 29(12):1505–1513, December 1994.

[18] Koen Swings. *Analog circuit design using constraint programming*. PhD thesis, Katholieke Universiteit Leuven, May 1995.

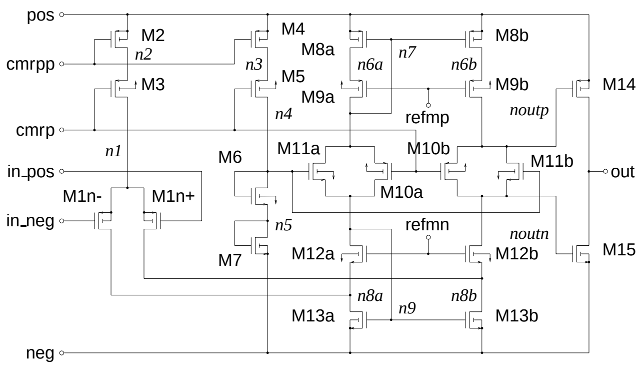


Fig. 11. **Schematic of the Hogervorst class-AB opamp.**

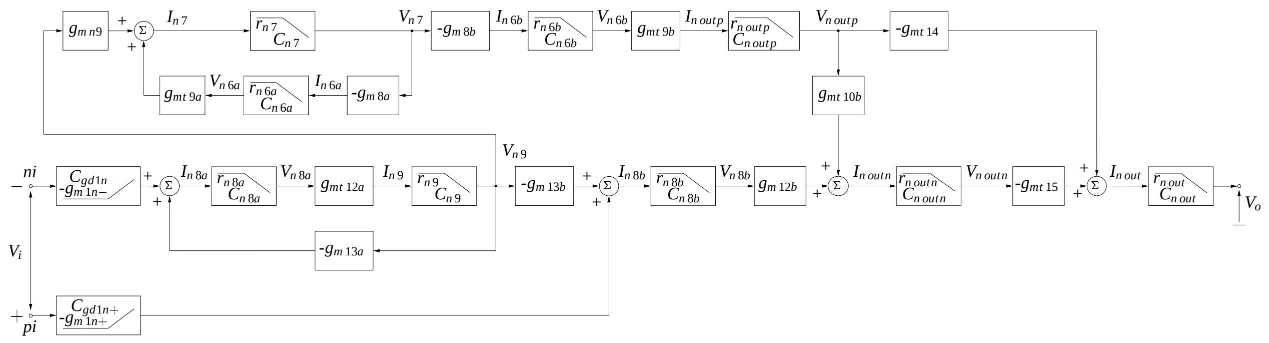


Fig. 10. **Behavioral signal path model of the Hogervorst class-AB opamp.**