Chapter 4
Improving Design Feature Reuse in Analog Circuit Design through Topological-Symbolic Comparison and Design Concept Combination

Cristian Ferent and Alex Doboli

Abstract. This chapter presents a novel circuit synthesis flow based on concept comparison, combination, learning, and re-use. The discussion overviews a technique for systematically comparing two analog circuits. The comparison data presents the similar and distinguishing performance characteristics of two circuits with respect to DC-gain, bandwidth, common-mode rejection ratio (CMRR), noise, and sensitivity. The comparison data is important for getting insight about the common and unique benefits of a circuit, selecting fitting circuit topologies for system design, and circuit topology refinement and synthesis. The technique matches the topologies and nodal symbolic expressions of the compared circuits to find nodes with similar electric behavior. The impact on performance of the unmatched nodes is used to express the differentiating characteristics of the circuits. Experiments illustrate the comparison technique for a pair of analog circuits.

4.1 Introduction

Analog and mixed-signal (AMS) systems and circuits are important components in many modern systems integrated or interacting with the physical world, such as applications in telecommunication, environmental sensing, healthcare, and smart infrastructures. The time and cost effort to design and verify AMS systems is expected to grow due to insufficient CAD support for designing, optimizing, and validating the systems. Recent studies suggest that the current productivity gap is around 100x, and sometimes as high as 1000x, as compared to what is needed to design next-generation electronic

Cristian Ferent · Alex Doboli
State University of New York at Stony Brook, Department of Electrical and Computer Engineering
e-mail: {cferent,adoboli}@ece.sunysb.edu

M. Fakhfakh et al. (Eds.): *Analog/RF & Mixed-Signal Circuit Sys. Design*, LNEE 233, pp. 77–99.
DOI: 10.1007/978-3-642-36329-0_4

systems [14]. It has been also observed that validation and verification of new designs is more difficult because automated tools create solutions that are less similar in style to what human designers develop. Generating solutions that are more "readable" by human designers is not a main priority for many current synthesis and optimization tools. Some of the envisioned solutions to address the productivity gap challenges include developing (i) novel design methodologies and techniques based on higher levels of abstractions, (ii) design methods utilizing parameterized topologies and modules, (iii) new synthesis, evaluation/simulation, and layout design algorithms capable of better optimization results in shorter time, and (iv) novel techniques to enable superior design reuse. Design feature reuse in analog circuit design has been arguably less studied as compared to the other avenues.

CAD tools for design reuse must automatically identify the design intent, strategies, and improvements that are embedded in previously developed solutions, and then analyze how improvements can tackle a new application's performance requirements and/or constraints of the utilized fabrication processes. This insight can then be reused to produce novel design solutions. Design feature reuse is envisioned to be very effective as many industrial companies and research groups already possess large sets of previous designs, including IPs. Besides, many modern tools for transistor sizing, layout design, and circuit topology selection and synthesis use design constraints to increase the likelihood of creating feasible solutions and/or to improve the convergence of the algorithms. The insight extracted from the available designs can be also utilized as constraints for existing synthesis tools, or as templates to guide a manual design process. A central component for getting the needed insight to enhance design feature reuse is developing a procedure to systematically compare the topological and performance attributes of analog circuits to understand how similar and different circuit features introduce new design variables and trade-offs.

This chapter introduces a novel circuit synthesis methodology based on concept comparison, combination, learning, and re-use. A central component of the methodology is a technique to systematically compare two analog circuits. Using a dual topological and symbolic matching scheme, the comparison method identifies the similar and distinguishing design features of an analog circuit as compared to another circuit, and how the features impact the nodal behavior and performance attributes, like DC-gain, bandwidth, noise, CMRR, and sensitivity. A set of constraints relate the behavioral descriptions to performance attribute modification. The final step of the comparison method characterizes how topological and behavioral differences modify trade-offs in a design, availability of free (orthogonal) variables to set performance attributes, achievable performance values, and hardness to find the design parameters. The chapter also overviews some of the current AMS synthesis methods, and argues for the need of developing novel approaches that can understand and learn new design features from existing solutions and designs created during the synthesis process.

In addition to AMS synthesis, the symbolic comparison technique could also be useful in various other design activities, like incremental circuit topology design, circuit design retargeting for new fabrication processes, circuit parameter sizing, and topology selection.

The chapter has the following structure. Section 4.2 overviews related work. Section 4.3 presents the synthesis flow. Section 4.4 details the proposed method for circuit comparison. Section 4.5 offers experimental results. Conclusions end the chapter.

4.2 Related Work

An overview of analog synthesis tools based on their architecture selection approach is presented in [18]. Tools are analyzed with respect to four criteria: (i) the abstraction level at which design decisions are made, (ii) the flexibility to tackle various kinds of circuits, (iii) the coverage of the overall design space defined by the possible topologies, and (iv) the capability to optimize the parameters of a circuit topology. Based on the four criteria, the authors suggest that current synthesis tools can be categorized into four main classes: (i) tools that select a topology based on designer experience or input from knowledge-based expert system, (ii) methods that decide the topology in parallel with parameter sizing based on circuits and sub-circuits stored in a library, (iii) tools that produce a topology through top-down synthesis starting from high-level descriptions, and (iv) bottom-up techniques that create architectures through systematic rules or stochastic evolution to connect devices into structures. This section reviews some of the existing synthesis tools in every category and then explains the need for novel approaches in circuit synthesis.

Synthesis tools that mimic knowledge-based expert systems utilize a static library of design rules that can tackle a certain family of circuits. IDAC [5] is an interactive design tool for a range of circuits, like OTAs, OpAmps, voltage and current amplifiers, comparators, and oversampling ADCs. The design methodology utilizes three types of knowledge, specific to the schematic, general circuit theory, and related to the circuit family. For instance, for OTA design, the first step performs worst-case distortion analysis due to variations of temperature and bias currents to derive the acceptable, nominal gain-bandwidth product, gain, slew rate, noise and phase margins. The second step sizes the devices based on a pre-specified design plan. Finally, the third step evaluates the correctness of the design, such as the resulting phase margin. Another expert-system for analog circuit design, BLADES [9], implements a given set of inference steps in which specification requirements and design knowledge is used to reason out the circuit topology and device dimensions. The topology selection algorithm in FASY [26] uses fuzzy rules based on specification requirements. OPASYN [15] utilizes a decision tree to

select circuit topologies. A decision tree distinguishes the common topologies from special circuits based on requirements like area, open-loop gain, power supply rejection ratio, and fully differential structure.

Template-based analog circuit synthesis can be viewed as an extension of knowledge-based design methods. They use a set of invariant, circuit-specific constraints (e.g., design templates) to find performance optimized AMS topologies, thus perform simultaneously topology selection and parameter sizing. Design templates usually describe implicitly an entire family of solutions, out of which the synthesis method identifies the best solution for the current specification. OASYS [11] utilizes a hierarchical set of templates, in which the top level corresponds to a certain type of circuits, like successive approximation ADC, the next level defines the structure of the circuits out of building blocks, i.e. comparator, sample-and-hold, D/A converter, the third level defines the structure of each building block including OpAmps and their RC networks, and the fourth level presents possible structures of OpAmps. A different kind of templates has been used to synthesize $\Delta\Sigma$ ADCs [24] and reconfigurable $\Delta\Sigma$ ADCs [28]. In this work, a template is a set of mixed integer nonlinear programming (MINLP) equations that express the alternative structures and parameters of the converters as well as the impact of circuit nonidealities on performance, e.g., finite gain, bandwidth, and noise. The MINLP equations of a template are solved to find the ADC structure, including its order, feedback and feedforward loops, and the coefficients of the structure.

Top-down synthesis flows transform high-level descriptions of circuits or systems into implementations optimized for the specification requirements. The transformation process includes steps for selecting the topology and sizing its parameters in addition to placement and routing the design. Various high-level descriptions have been explored, including state-space model [1] and signal-flow graphs [7, 13]. Using VHDL-AMS as a specification language for AMS synthesis is discussed in [8]. The transformation steps are realized based on predefined rules [1], exploration [8], or constraint transformation [3].

More recently, stochastic evolution has been proposed to synthesize AMS circuits and systems. The method in [22] utilizes genetic programming to evolve CMOS OpAmps utilizing basic devices and building blocks. Current flow analysis is employed to verify basic electrical requirements of an evolved topology, such as having current flow through the component lists of a circuit, operating all transistors in their correct regions, and avoiding floating and isolated devices. The approach in [17] performs variation-aware structural synthesis of analog circuits to produce decision trees that indicate the topologies that implement better a set of requirements. Decision trees partition the performance space such that every internal node describes a performance constraints, like power consumption, area, gain, dynamic range, etc., and every leaf node is a circuit topology. A path through the decision tree describes a circuit (leaf) and the set of requirements that are met by the circuit (the reunion of the leaf node constraints). The synthesis approach

utilizes hierarchical specified analog building blocks representing one and two stage OpAmps, various inputs (e.g., single-ended, differential, stacked, folded cascode, etc.) and loads (i.e. cascode, noncascode, resistor), different current mirror circuits, and outputs. Structural homotopy avoids early elimination of attempts to create new topologies by attaching a maximum age for the individuals of each layer of the evolutionary synthesis algorithm. The technique proposed in [4] creates topologies bottom-up starting from devices and gradually creating, based on rules, blocks of more complex functionality. Blocks are stored for re-use in a library together with information about their neighbors, which are blocks placed adjacently.

In addition to the criteria enumerated at the start of the section, we consider that circuit synthesis tools can be also distinguished based on their capability to "understand" and "learn" from the features of previous solutions, such as to use the explored design space to extract new knowledge on effective design decisions (e.g., to use some structures together and to size parameters according to specific constraints), to identify design space regions that have not been explored yet, and to produce design strategies, i.e. parameter optimization plans, of superior quality, shorter design closure, and higher transparency with respect to the selected decisions. Higher transparency helps designers understand easier the purpose of the selected design decisions, thus aids in design validation. Traditional reuse-based methodologies rely exclusively on static libraries of circuits, design rules and equations, circuit models, and layout templates [2]. This chapter argues that understanding and learning new design features produced dynamically during synthesis requires incorporating into synthesis flows the capability to automatically identify, characterize, and reuse design features created dynamically during synthesis. [25] indicates that reusing automatically mined constraints on design parameters significantly improves convergence of synthesis and reduces synthesis time.

Automatically understanding and learning the useful features in circuits requires to find the similarities and differences between the electrical behavior and performance of the circuits. Simulating and/or modeling individually each circuit gives some insight but understanding in detail the links between topological changes and performance differences requires complicated manual analysis. Previous work suggests that two circuits can be compared using their performance space descriptions [6, 16, 23]. Performance space descriptions specify the performance trade-offs of a circuit, like DC gain, 3db frequency, and slew rate [23], by abstracting away all design variables. However, they give little insight into how topological differences in two circuits introduce new design variables that create novel trade-offs and new opportunities (flexibility) to improve performance. Also, there is no indication about how design-specific constraints can improve performance, such as certain pole-zero placements, or constraints on device parameters, like transconductances. This insight is important for circuit topology selection and design reuse, including circuit resizing for new requirements, incremental topology changes for novel

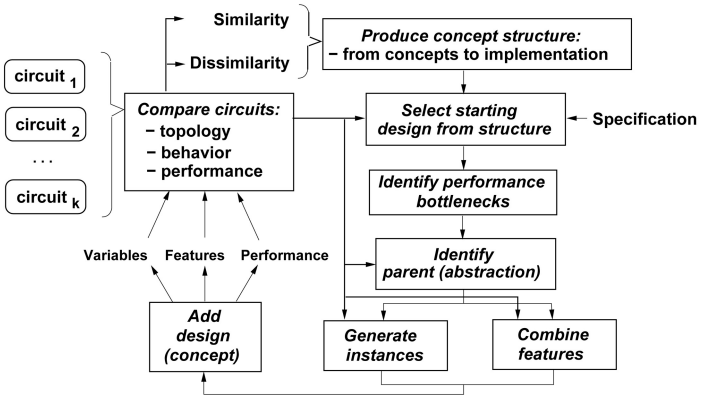

Fig. 4.1 Synthesis flow based on design concept comparison, combination, learning, and re-using

applications, and design migration to different fabrication processes. The next section presents a synthesis method based on design feature comparison and combination.

4.3 Circuit Synthesis Based on Concept Comparison and Combination

This section presents a novel circuit synthesis flow based on design concept comparison, learning, combining, and re-using. The main premise of the synthesis flow is that solving a circuit design problem requires to identify a set of design steps, so that every step is justified by the fact that it improves performance (i.e. at least one performance attribute) or relaxes design constraints (e.g., at least one constraint). Every synthesis step attempts to address the performance bottlenecks of a circuit topology by changing the relations between the design variables of the bottlenecks. Relations are changed by (i) searching for previous designs with related bottlenecks and then combining their features with the current solution, or by (ii) exploring orthogonal ways of relating the variables of the bottlenecks through new ways of interconnecting circuit nodes. Circuit comparison is a main step of this synthesis flow as it identifies the commonalities and differences among circuit features and their importance.

The proposed synthesis flow is shown in Figure 4.1. The concept structure used in design is constructed based on a set of existing circuits, e.g., $circuit_1$ to $circuit_k$ in the figure. The circuits are compared pairwise with regard to their topology, electrical behavior, and performance. Section 4.4 details the comparison procedure. The information about the similar and dissimilar

features of the circuits is used to construct a concept structure in which leaf nodes represent circuit designs, and intermediate nodes express the common design features of the children. Hence, the children of an arbitrary node n describe a sampling of the conceptual space represented by node n. Similar to [10], a classification method based on entropy can be used to create the concept structure that maximizes the amount of common design features of the intermediate nodes and minimizes the number of common features across different intermediate nodes. This concept structure optimizes the partitioning of the different circuit features depending on their impact on performance. Note that the lower level concepts of the structure represent more complex and detailed solutions while the higher level concepts correspond to more abstract (conceptual) designs.

The synthesis flow starts by selecting a concept of the design structure, such that its performance attributes are closest to the problem description and its bottlenecks do not conflict with the description. This design is likely to be efficiently refined and modified to accommodate the requirements of the specification. Next, the iterations of the synthesis flow attempt to minimize the miss-matching between the design performance and the specification requirements by conducting the following steps. First, it analyzes the nature of the performance bottlenecks of the current solution and then finds bottom-up in the design structure the first parent node that does not have the bottlenecks. The parent node is found by comparing the current node with the nodes placed bottom-up in the structure. Then, a child without the features that cause the bottleneck is used to further synthesize the solution. Alternatively, the method attempts to remove the current bottlenecks by adding (combining) features that are present in other designs (of the concept structure) that do not have the bottlenecks. The new features are structural connections between the circuit nodes that create the current bottlenecks. This changes the relations among the nodal currents and voltages. The produced performance modifications are evaluated by comparing the modified and original circuits. The third option is to exhaustively create new ways of relating the nodal variables of the bottlenecks, so the bottleneck is changed and a better matching to the problem description is possible. The performance improvement of the explored structures is characterized by comparing the original and the modified circuits. Any new solutions created during the steps of concept combination or new feature generation are added to the concept structure.

A case study example for designing differential MOS transconductor cells based on the concept structure is detailed in [31]. The problem requirements are to improve linearity and input range. The simplified concept structure is shown in Figure 4.2. The top level concept, common to all circuits, guarantees that the transconductor functionality is implemented, $I_o = f(V_i)$, where I_o and V_i are the differential output current and input voltage, respectively. Child concepts add details to the structure. One alternative is to use the same source voltage for both input MOS devices ($V_{S1} = V_{S2}$) and a direct

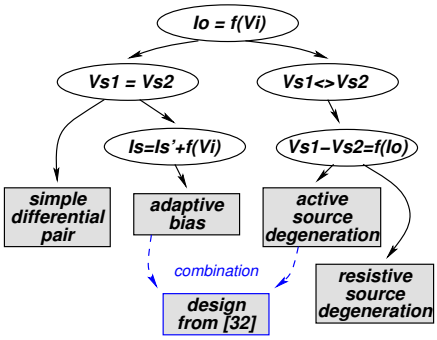

Fig. 4.2 Simplified concept structure for transconductor designs

implementation is the simple differential pair transconductor. If linearity and/or input range of this solution are insufficient for the given specification, another child concept of the parent node can be selected for synthesis. This alternative concept maintains the same source voltage, but adds an input dependence to the tail biasing current, $I_S = I'_S + f(V_i)$. This improves linearity and a specific implementation is the adaptive bias transconductor. Another option is to explore the concept utilizing different source voltages for the input transistors ($V_{S1} <> V_{S2}$). One possible child of this concept correlates the difference in source voltages to I_o and can be specifically implemented through either resistive or active source degeneration topologies. If none of the designs represented in the concept structure offer adequate performance, concept combination can be attempted to relax constraints. For example, combining the features of the adaptive bias and source degeneration transconductors can produce a highly linear design [32].

The concept structure of the flow is the main data structure used to learn, combine, and re-use design features. It is produced based on comparing structural, behavioral, and performance attributes of circuits, as presented in the next section.

4.4 Systematic Comparison of Analog Circuits

Comparing the nodal behavior and performance of two analog circuits comprises of two activities: (i) relating the electrical behavior of the circuit nodes and sub-circuits in the two circuits, and (ii) understanding how commonalities and differences in the nodal behavior impact performance. The two activities require a dual topological-symbolic matching to find nodes connected in similar structures and with similar electrical behavior. Topological matching alone is insufficient as it only identifies the circuit nodes with similar connectivity (to other devices) and the node groups (clusters) with same structure, but does not perform any analysis of the electrical behavior. It is known that similar topological structures can produce different signal flows depending on

biasing, output sensing, and connectivity with other structures. The literature presents several topological matching approaches, including clan-based matching [12] and string-based matching [21].

Two circuit nodes (in different circuits) have similar electrical behavior, if there are conditions under which the transfer function (TFs) between the nodes and inputs can be matched, such that the two TFs represent the same mathematical expressions. Examples of such conditions include requirements that certain device parameters are equal (matching), some device values are much larger (smaller) than others, certain device parameters can be neglected, and so on. Characterizing the performance differences of two circuits must capture how topological and behavioral changes modify performance attributes with respect to the following key aspects of a design: (i) new trade-offs, (ii) availability of free (orthogonal) variables to control specific performance attributes, (iii) achievable performance values, and (iv) hardness to find the parameter values that set a desired performance value.

The circuit comparison procedure performs the following four steps to quantify the above key aspects. The first step, *topological matching*, relates the structural features of the two circuits, e.g., it identifies maximal sets of nodes with similar poles and connectivity to other nodes. The second step, *symbolic matching*, matches the electrical behavior of the circuit nodes using the topologically matched nodes as a reference. It computes transfer functions H_{comm} and H_{diff} expressing the similarities and differences in the electrical behavior of the circuit nodes. *Constraint generation*, the third step, creates constraints defining how functions H_{comm} and H_{diff} impact performance, such as the resulting ΔDC gain, Δbandwidth, Δnoise, ΔCMRR, and Δsensitivity. Finally, the *performance characterization* step describes the capability of a design to meet the generated constraints, and thus achieve certain performance and trade-off values. The four steps are detailed next.

4.4.1 Topological Matching

We define that two nodes are topologically matched if they have the same number and kind of devices connected to the nodes. Two clusters of connected nodes (pertaining to two circuits) are topologically matched if all nodes in the clusters are also topologically matched.

Topological matching finds the maximal node clusters with similar structure. Given two circuits C_1 and C_2, the topological matching algorithm starts by identifying for every node i in circuit C_1 the list of nodes in C_2 that are candidates to be topologically matched with node i. Then, it exhaustively considers for every node in C_1 every matching candidate in C_2 while maximizing the size of the matched clusters that are found using the candidate.

Figure 4.3(a) shows the superimposed schematics of a compensated two-stage amplifier and a class AB two-stage amplifier [19]. The two topologies

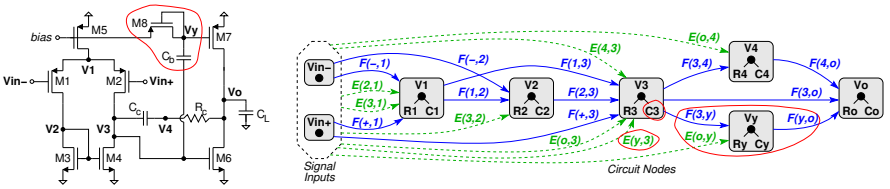

Fig. 4.3 Schematic and macromodel for simple OpAmp and class AB two-stage OpAmp [19]

have all nodes matched with the exception of nodes V_3, and V_y. The matched clusters of the two circuits include all nodes but these two nodes.

4.4.2 Symbolic Matching

Given two circuits, symbolic matching finds the nodes with similar electrical behavior, including any constraints on the device parameters that make the behavior similar. The proposed method identifies the similarities and differences in the behavior of two nodes by computing the following TFs: (i) H_{comm} defines the common symbolic parts of the TFs of the compared nodal voltage, and (ii) H_{diff} expresses the distinguishing symbolic terms of the two TFs. Figure 4.4(a) illustrates the TFs of the two structures shown as signal-flow graphs. The topologically matched nodes are highlighted.

The electrical behavior is described in the proposed method through macromodels, which are built using the technique in [29]. The uncoupled models are structural and represent all circuit nodes. Based on the nodal formulation, frequency domain behavior is expressed at the circuit nodes using parameters of connected devices. The voltage at each node V_i is characterized by a pole expression (R_i and C_i) and the couplings from other nodal voltages in the circuit in the form of voltage-controlled current sources (VCCS). For MOS transistors, a decoupled model is used, obtained from an error free transformation of the hybrid-π representation. Considered device parameters include terminal transconductances (g_{mg}, g_{md}, and g_{ms}) and capacitances (C_{gd}, C_{gs}, C_{gb}, C_{db}, and C_{sb}). For example, the drain terminal voltage V_D in the decoupled MOS model is determined by pole components $R_D = 1/g_{md}$ in parallel with $C_D = C_{gd} + C_{db}$ and the parallel coupling (VCCS) from the gate $(sC_{gd} - g_{mg})V_G$ and source $g_{ms}V_S$ terminals. Similar forms are expressed for the gate and source voltages in the decoupled MOS model [29].

Figure 4.3(b) presents the superimposed macromodels of the two amplifiers. The encircled nodes and edges correspond to the unmatched nodes. All circuit nodes V_i are represented in the model. Coupling between nodes can be direct (expressions $F(i, j)$), or equivalent, after removing feedback connections (expressions $E(k, j)$). The voltage at any node V_j is expressed as follows:

$$V_j = \frac{R_j}{1 + sR_jC_j} \times \left(\sum_i F(i,j)V_i + \sum_k E(k,j) \right);\ i,k \neq j. \tag{4.1}$$

Terms R_j and C_j are the resistive and capacitive components of the pole at node j, and depend on the transconductances and capacitances of the devices connected at node j. For example, for V_1 in Figure 4.3, $R_1 = 1/(g_{ms1}+g_{ms2}+g_{md5})$ and $C_1 = C_{gs1} + C_{sb1} + C_{gs2} + C_{sb2} + C_{gd5} + C_{db5}$.

Direct coupling has the general form:

$$F(i,j) = sC_m \pm G_m. \tag{4.2}$$

C_m and G_m are the junction capacitances and terminal transconductances of the MOS devices connected to nodes i and j. For example, at node V_1 in Figure 4.3, $F(-,1) = sC_{gs1} + g_{mg1}$.

The equivalent coupling results by decoupling and eliminating all feedback relations from the initial coupled model and replacing them with equivalent edges:

$$E(k,j) = (sC_m \pm G_m) \times V_{k\ eq}. \tag{4.3}$$

k denotes the circuit node where the (eliminated) feedback originated and j is the influenced node. C_m and G_m are the junction capacitances and terminal transconductances of the devices connected to nodes k and j. $V_{k\ eq}$ is the equivalent voltage of node k after removing the feedback to node j.

The decoupling sequence is performed following the input-output signal path to order the nodes of the coupled model and identify which $V_{k\ eq}$ needs to be solved first. Each equivalent voltage is then expressed in the general s-polynomial form, in terms of the circuit inputs. The symbolic expression is explored from lower to higher order with controllable accuracy. The algorithm exploits the pattern of each coefficient of s^k in the expression, $a_k = \sum(\pm \prod^k C_m \prod^{N-k} G_m)$, where N is the total number of nodes. An optimization method identifies the combination of model parameters for each coefficient a_k such that the cumulative error between the original coupled and current uncoupled models is minimized across the frequency range and different operating points [29].

The symbolic matching step analyzes separately each set of topologically matched clusters, so that the effect of the unmatched nodes is kept locally to the cluster. It computes for every topologically matched pair of nodes in the clusters the TFs H_{comm} of the common part, and $H_{1,diff}$ and $H_{2,diff}$ of the unmatched nodes in each of the two circuits. TFs H_{comm} and H_{diff} are using the following expression:

$$H = \sum_p \left(\frac{\prod_t F_t(s)}{\prod_j \frac{(1+sR_jC_j)}{R_j}} \right). \tag{4.4}$$

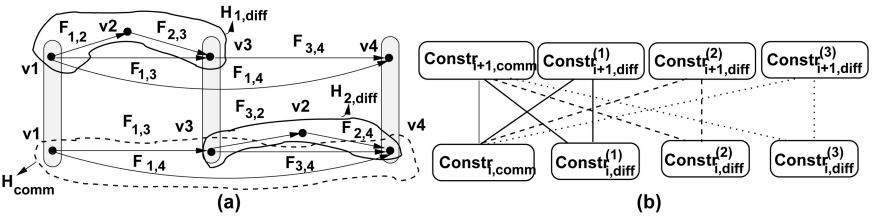

Fig. 4.4 (a) Symbolic matching and (b) constraint generation

p are the signal paths of a structure (including matched and unmatched nodes), j represents all nodes of a path with pole components R_j and C_j, and F_t are the edge labels in the signal flow graphs of the macromodel.

Example: The following TFs exist for the two structures in Figure 4.4(a). TF $H_{comm} = \frac{F_{1,3}F_{3,4}}{P_1P_3P_4} + \frac{F_{1,4}}{P_1P_4}$, where P_i are the symbolic expressions of the pole in node i. $F_{i,j}$ is the symbolic expression defining the connection between nodes i and j. TF $H_{1,diff} = \frac{F_{1,2}F_{2,3}F_{3,4}}{P_1P_2P_3P_4}$ is the difference between the top and bottom structure, and TF $H_{2,diff} = \frac{F_{1,3}F_{3,2}F_{2,4}}{P_1P_2P_3P_4}$ distinguishes the bottom from the top structure.

4.4.3 Constraint Generation

The impact on performance attribute $\mathcal{P}_i$ (e.g., DC gain, bandwidth, noise, CMRR, and sensitivity) due to topological and TF differences are estimated using H_{comm} and H_{diff} for every matched cluster of the circuits. Each TF generates a set of constraints that must be met to satisfy performance $\mathcal{P}_i$. Figure 4.4(b) shows the constraint sets corresponding to the TFs in Figure 4.4(a).

1. DC gain. Series connection of TFs H_{comm} and H_{diff} produces a DC gain of $DCgain_{H_{comm}} \times DCgain_{H_{diff}}$. Parallel connection creates a DC gain of $DCgain_{H_{comm}} + DCgain_{H_{diff}}$.

2. Bandwidth. The relation between the TFs H_{comm} and H_{diff} and bandwidth can be estimated based on Loop-Gain-Poles product (LP) [27]:

$$\omega_{max} \approx (|1 - DC\ gain| \prod_{i=1}^{n} p_i)^{\frac{1}{n}}. \tag{4.5}$$

ω_{max} is the estimated maximum bandwidth, and n is the number of dominant poles p_i. The set of considered dominant poles (usually $n \leq 2$) is selected from the total number of poles, given by the number of circuit nodes in the matched cluster. Then the bandwidth corresponding to TF H_{comm} has the following upper bound:

$$\omega_{max}^{H_{comm}} \approx (|1 - DC\ gain_{H_{comm}}| \prod_{k=1}^{m} p_k)^{\frac{1}{m}}, \tag{4.6}$$

where p_k are the m dominant poles of the common TF.

The difference TF, H_{diff}, modifies the bandwidth expression ω_{max} depending on how H_{comm} and H_{diff} are connected with each other, e.g., series or parallel.

For series connection, the bandwidth corresponding to TF $H_{comm} \times H_{diff}$ is equal to the following value:

$$\omega_{max}^{H_{comm}H_{diff}} \approx (|1 - DC\ gain_{H_{comm}} DC\ gain_{H_{diff}}| \prod_{k=1}^{m+n} p_i)^{\frac{1}{m+n}}, \tag{4.7}$$

where TF H_{diff} has n dominant poles that are not among the m dominant poles of TF H_{comm}. The change in bandwidth due to TF H_{diff} is equal to the expression:

$$\frac{\omega_{max}^{H_{comm}H_{diff}}}{\omega_{max}^{H_{comm}}} \approx \left[\frac{(DC\ gain_{H_{diff}} \prod_n p_i)^m}{(DC\ gain_{H_{comm}} \prod_m p_i)^n}\right]^{\frac{1}{m(m+n)}}. \tag{4.8}$$

For $m = n$, a sufficient condition for increasing the resulting bandwidth is if the DC gain of H_{diff} is higher than that of H_{comm} and the distance of the dominant poles to the origin is higher for TF H_{diff} than for TF H_{comm}.

For parallel connection, the bandwidth change due to $H_{comm} + H_{diff}$ can be estimated when assuming that each TF is expressed as $H_i = \frac{\prod_j z_j}{\prod_i p_i}$, where z_j are zeros and p_i are poles. The two bandwidths can then be related as in the next expression:

$$\frac{\omega_{max}^{H_{comm}+H_{diff}}}{\omega_{max}^{H_{comm}}} \approx \left[\frac{[(1 + \frac{DCgain_{H_{diff}}}{DCgain_{H_{comm}}}) \prod_n p_i)]^m}{(DC\ gain_{H_{comm}} \prod_m p_i)^n}\right]^{\frac{1}{m(m+n)}}. \tag{4.9}$$

Expressions (4.8) and (4.9) are used repeatedly for generalized products and sums of TFs.

Similar constraints are formulated for CMRR and noise.

4.4.4 Performance Characterization

Let $\mathcal{R}_i(H, p_1, p_2, ..., p_k)$ be the constraint introduced on performance attribute i through TF H with parameters p_i. The difficulty in meeting constraint $\mathcal{R}_i$ can be approximated by the value $Rate(i, H) = \max |\frac{\sum_{i,j} \frac{\partial^2 \mathcal{R}_i}{\partial p_i \partial p_j}}{\sum_i \frac{\partial \mathcal{R}_i}{\partial p_i}}|$, an estimation of the minimum rate of convergence when optimizing constraint

```
PROCEDURE (Set of constraints) IS
(1) FOR every equation ej DO
(2)   FOR every variable xi DO
(3)    compute sensitivity of ej with respect to variable xi;
(4) FOR every equation ej DO
(5)   find sets FVj of free variables and TVj of trade-off variables;
(6)   bind variables in sets TVj and FVj;
(7) FOR every equation ej DO
(8)   Fl = Fl + cardinality of set FVj;
(9)   FOR every variable xi in FVj DO
(10)     Diffj = Diffj * Ratei;
(11)  Diff = Diff + Diffj;
END PROCEDURE;
```

Fig. 4.5 Procedure to estimate the flexibility and convergence of a design

$\mathcal{R}_i(H, p_1, p_2, ..., p_k)$ over the domains of parameters p_i. If circuit C_i is defined by transfer functions H_{comm} and $H_{i,diff}$ then the difficulty in optimizing a set of performance attributes m is approximated as $\sum_m Rate(m, H_{comm}) + \sum_m Rate(m, H_{i,diff})$.

The flexibility in meeting a set of constraints depends on the maximum number of free variables that intervene in the constraints. The difficulty in meeting the constraints can be estimated depending on the value *Rate*. Figure 4.5 presents the algorithm for estimating the flexibility and difficulty of optimizing a design. First, the algorithm computes for every equation e_j the sets FV_j and TV_j of free and trade-off variables. A variable x_i is free, if it has the same kind of sensitivity for all equations, e.g., only positive or only negative. A variable x_i is a trade-off variable if it has both positive and negative sensitivities. Some trade-off variables are removed from sets TV_j by binding free variables in sets FV_j, such that they cancel out the effect of the trade-off variables. The number of remaining free variables describes the available flexibility in deciding the performance values described through constraints. The product of the variable rates indicates the difficulty of a gradient-based optimization algorithm to find the solution.

4.5 Experiments

This section presents the symbolic comparison of two circuits to understand the design trade-offs and performance changes introduced by the dissimilar topological features of the circuits.

Figure 4.6 shows two low-voltage amplifier circuits, named AMP_1 and AMP_2. The first design is a two-stage class-AB topology [19]. The second circuit is a three-stage amplifier with positive feedback compensation [20].

First, topological matching found the similar and distinct nodes of the two circuits with respect to their structure and electrical behavior. Figure 4.7 illustrates the nodes and couplings of the two amplifiers. Nodes V_{in+}, V_{in-}

(signal inputs), V_1, and V_o (output) have identical symbolic pole expressions in both designs. Similarly, the couplings between nodes, Fc_i ($i = \overline{1,9}$), are the same in both circuits. Nodes V_3 and V_7 are only partially-matched due to small differences in their symbolic pole expressions. Enforcing that nodes V_3 have comparable pole components in both circuits results in the following two constraints: $g_{mg3}|_{AMP_1} \equiv g_{ms8}|_{AMP_2}$ and $(C_{gs3} + C_{gb3})|_{AMP_1} \equiv (C_{gs8} + C_{sb8})|_{AMP_2}$, when device M_1 parameters are matched between the two circuits, and transistors M_3 and M_4 are functionally matched in AMP_1 (current mirror). The constraints imply that device parameters of M_3 in circuit AMP_1 are paired with those of device M_8 in circuit AMP_2.

Similar matching constraints exist for the pole at node V_7: $(g_{mdR} + g_{mgR})|_{AMP_1} \equiv (g_{md12} + g_{md13})|_{AMP_2}$ and $(C_{dbR} + C_{gsR} + C_{gbR})|_{AMP_1} \equiv (C_{gd12} + C_{db12} + C_{gd13} + C_{db13})|_{AMP_2}$, when devices M_7 and capacitances $C_b \equiv C_{m2}$ are matched between the two designs. The two constraints link the parameters of device M_R in AMP_1 to the combined parameters of two devices, M_{12} and M_{13}, in AMP_2. The second circuit has a greater flexibility in meeting the matching of the above conditions.

In Figure 4.7, the input and output blocks of circuits AMP_1 and AMP_2 are matched and express similar electrical behavior. The distinguishing substructures are at nodes V_2, V_4, V_5, and V_6 with couplings between the nodes given by symbolic expressions Fd_j, $j = \overline{1,5}$. For example, the additional nodes V_4 and V_6 in AMP_2 impose different model graph edges: Fd_1 and Fd_2 for the input block, and Fd_4 and Fd_5 for the output block. The extra block in circuit AMP_2 has no equivalent in AMP_1.

Finally, the set of symbolic transfer functions for each circuit block are generated: (i) H_{comm} defines the common symbolic parts of the blocks in both circuits, and (ii) H_{diff} expressing the distinguishing symbolic terms for the blocks in each circuit.

The transfer functions of the input blocks of circuits AMP_1 and AMP_2, respectively, to node V_2 are as follows:

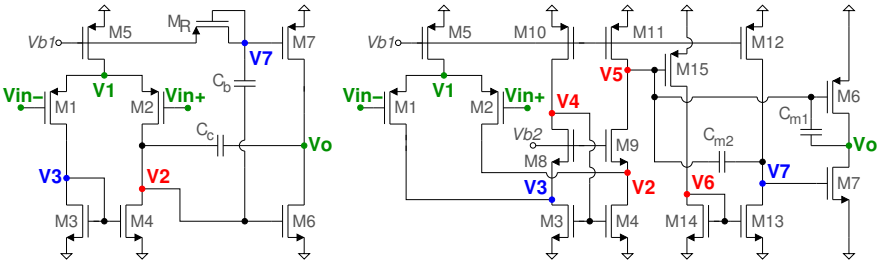

Fig. 4.6 Two low-voltage differential amplifiers: (a) class-AB 2-stage AMP_1 [19] and (b) positive feedback compensation 3-stage AMP_2 [20]

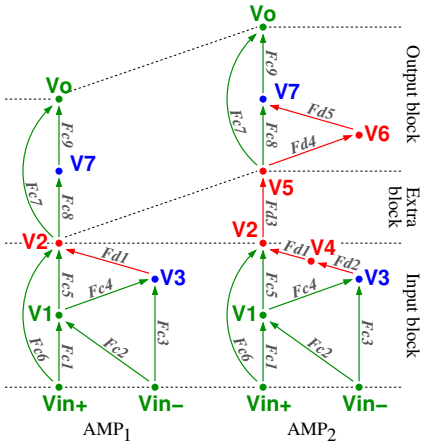

Fig. 4.7 Model graphs and sub-blocks for circuits AMP_1 and AMP_2

$$AMP_1 : H_2 = H_{comm_1} \times H_{diff_1} + H_{comm_2} \times H_{diff_2}, \tag{4.10}$$

$$AMP_2 : H_2 = H_{comm_1} \times H_{diff_3} + H_{comm_2} \times H_{diff_4}, \tag{4.11}$$

where H_{comm_1} and H_{comm_2} are the common signal paths in both designs and H_{diff_i} $(i = \overline{1,4})$ define the differences of the two input blocks:

$$H_{diff_1} = \frac{R_2}{1 + sR_2C_2}, H_{diff_2} = \frac{R_2Fd_1}{1 + sR_2C_2}, \tag{4.12}$$

$$H_{diff_3} = \frac{R_2}{1 + sR_2C_2}, H_{diff_4} = \frac{R_2R_4Fd_1Fd_2}{(1 + sR_2C_2)(1 + sR_4C_4)}. \tag{4.13}$$

Expressions (4.10)-(4.13) indicate that the input blocks of circuits AMP_1 and AMP_2 differ because of the poles at nodes V_2 and V_4, and the coupling between nodes $V_3 \rightarrow V_2$ and $V_3 \rightarrow V_4 \rightarrow V_2$.

The extra block of design AMP_2 has no equivalent in circuit AMP_1 (see Figure 4.7). Its transfer function to circuit node V_5 is defined only by unmatched components:

$$H_5 = H_{diff_5} = \frac{R_5Fd_3}{1 + sR_5C_5}. \tag{4.14}$$

The matching illustrated in Figure 4.7 shows that the output block of AMP_1 is composed of only matched nodes. Only AMP_2 exhibits differences in this circuit block with a transfer function to node V_o defined as follows:

$$H_o = H_{comm_3} + H_{comm_4} \times H_{diff_6}, \tag{4.15}$$

where $H_{comm_{3,4}}$ represent the common structures (also present in AMP_1). H_{diff_6} is defined by the unmatched pole and edges related to node V_6, $H_{diff_6} = \frac{R_6 F d_4 F d_5}{1+sR_6C_6}$.

For constraint extraction, the terms of the common (H_{comm}) and different (H_{diff}) transfer functions are instantiated with their respective macromodel symbolic expressions based on device parameters. Considering the corresponding matched parameters constant, constraints are expressed such that unmatched parameters of H_{diff} can be used to improve performance.

For example, in the output block of AMP_2, the node coupling terms of H_{diff_6} from equation (4.15) are expressed as $Fd_4 = sC_{gd15} - g_{mg15}$ and $Fd_5 = sC_{gd13} - g_{mg13}$. The pole components for node V_6 are $R_6 = \frac{1}{g_{md15}+g_{md14}+g_{mg14}}$ and $C_6 = C_{gd15} + C_{db15} + C_{db14} + C_{gs14} + C_{gb14} + C_{gs13} + C_{gd13} + C_{gb13}$.

The DC-gain of each circuit sub-structure is characterized by transfer functions $H_{comm_i}(0)$ and $H_{diff_i}(0)$. For analysis, considering $H_{comm_i}(0)$ constant, the constraints on $H_{diff_i}(0)$'s design variables are identified, such that the overall gain is improved. For the output block of AMP_2, the constraint on different parameters is given by the following expression:

$$K_1 + K_2 \frac{g_{mg15}}{g_{md15} + g_{md14} + g_{mg14}} \nearrow . \tag{4.16}$$

Constants $K_1 = \frac{g_{mg6}}{g_{md6}+g_{md7}}$ and $K_2 = \frac{g_{mg7}g_{mg13}}{(g_{md6}+g_{md7})(g_{md12}+g_{md13})}$ correspond to the parameters of common transfers $H_{comm_{3,4}}$ from equation (4.15). Note that we also consider constant parameters for M_{13} in the expression of Fd_5 as it contains device parameters topologically matched for the pole at V_7.

The gain-poles product (GPP) is used to estimate bandwidth [27] in each circuit block by selecting at most two dominant poles. Any remaining poles of the block are considered non-dominant and their constraints are expressed. For the output block of AMP_2, the only valid dominant pole set is P_o and P_7, introducing two constraints on distinguishing parameters, $\frac{1}{g_{md15}+g_{md14}+g_{mg14}} \searrow$ and $K_3 + C_{gd15} + C_{db15} + C_{db14} + C_{gs14} + C_{gb14} \searrow$. Then the gain-pole product increases when:

$$K_4 \frac{g_{mg15}}{g_{md15} + g_{md14} + g_{mg14}} \nearrow . \tag{4.17}$$

The constants for this case are expressed as $K_3 = C_{gs13} + C_{gd13} + C_{gb13}$ and $K_4 = \frac{g_{mg7}g_{mg13}}{g_{md12}+g_{md13}}$.

Noise is modeled for each circuit block of the two circuits using the method in [29]. Considering the common path noise fixed, constraints are extracted such that the noise contribution of the differences is diminished. The relevant constraints for the output block of AMP_2 are $\frac{g_{mg15}}{g_{md15}+g_{md14}+g_{mg14}} \searrow$ and $K_3 + C_{gd15} + C_{db15} + C_{db14} + C_{gs14} + C_{gb14} \nearrow$; or $g_{m15} \approx g_{m14}$.

With respect to CMRR, both circuits include only matched design parameters. The relation between the parameters of devices M_5, M_1, and M_2 controls this performance. The differences impact DC-gain and CM-gain in

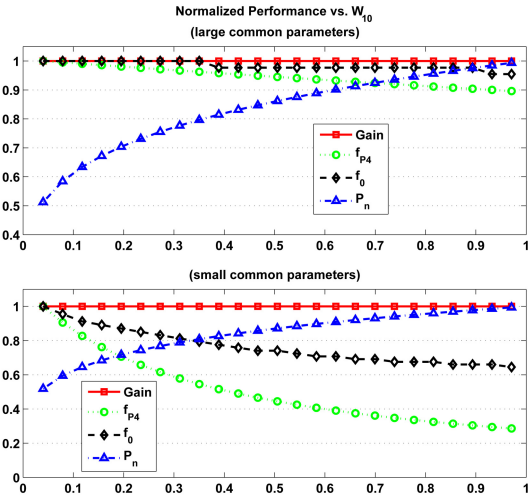

Fig. 4.8 Input block performance trade-offs for comparing circuits AMP_2 and AMP_1 with respect to W_{10}

the same way. This implies that similar CM performance can be expected in AMP_1 and AMP_2 when common signal path variables are precisely matched.

Constraints are extracted in the same manner for all circuit blocks [30]. Table 4.1 summarizes the required trends (except equality relations) for the distinguishing design variables of circuit AMP_2 across the entire amplifier, including input, extra, and output blocks. The trends were determined using the extracted constraints. For example, increasing the output block's DC-gain in AMP_2 by satisfying constraint (4.16) can only be performed by deteriorating noise performance. Parameters of device M_{10} have two variations for dominant poles and bandwidth, with respect to each feasible set. Design variables g_{mg15} and g_{md15} are bound by conflicting trade-offs. However, both parameters are determined by device M_{15}, as is the case of $C_{gd15} + C_{db15}$, and the common dependence of both transconductance and capacitance on W_{15} correlates these parameters.

Table 4.1 Desired variable trends with respect to performance in circuit AMP_2

Variables	Gain	CM	Noise	Pole	GPP
g_{md10}	↘	-	↗	↗, ↘	↘,-
$C_{gd10} + C_{db10}$	-	-	↗	↘, ↗	-, ↘
g_{mg15}	↗	-	↘	-	↗
g_{md15}	↘	-	↗	↗	↗
$C_{gd15} + C_{db15}$	-	-	↗	↘	-
$C_{gs15} + C_{gd15} + C_{gb15}$	-	-	↗	↘	-
$g_{md14} + g_{mg14}$	↘	-	↗	↗	↘
$C_{db14} + C_{gs14} + C_{gb14}$	-	-	↗	↘	-

The procedure further investigates these parameter correlations and performance trade-offs by evaluating the model of each individual circuit block. We vary the widths of devices identified by parameters in Table 4.1 across a predefined range and characterize the normalized performance trends. Common parameters and branch bias currents through matched devices are considered constant. This allows us to estimate the performance sensitivities to distinguishing circuit parameters and provides insight about the most appropriate relative sizing strategy that can improve performance.

For the input block of circuit AMP_2, design parameters related to device M_{10}, i.e. g_{md10} and $C_{gd10} + C_{db10}$, influence the resistive (R_4) and capacitive (C_4) components of the pole at node V_4. Figure 4.8 shows the correlations between W_{10} and gain, pole frequency (f_{P4}), unity-gain frequency (f_0), and total block output noise (P_n). The normalized performances are presented for two scenarios: when the common path parameters are (i) large or (ii) small with respect to those of M_{10}. For either case, P_4 is the first dominant pole. We observe that parameters of device M_{10} have virtually no impact on the circuit block's gain. This is due to the constant branch current imposed by the matched devices. The total noise exhibits the same sensitivity to M_{10} parameters regardless of the constant parameter values and is minimized for relatively small transistor widths. The variation of 50% across the investigated range suggests that when the common parameters are matched between the two designs, M_{10}'s width should be kept low. This would be required to attempt comparable noise performance in both AMP_1 and AMP_2. For pole and unity-gain frequency, W_{10} is again best kept low especially in the case of small common parameters. There is a deterioration in first dominant pole frequency (equivalent to the -3dB point) of up to $\approx$ 70% from the maximum. Impact on unity-gain frequency is reduced, but can still decrease with up to $\approx$ 35% as the size increases. The opposing trends between total noise and pole frequencies suggests that g_{m10} is a dominant parameter for noise. $C_{gd10}+C_{db10}$ best controls bandwidth performance and has limited impact on total noise. In terms of sensitivity, all impacted performances provide a more pronounced variation within the first third of the width range. As W_{10} is further increased, the impact on performance is reduced.

Considering the output block of circuit AMP_2, Figure 4.9 depicts the normalized performance plots based on the width of device M_{15} when M_{14} is kept constant. The analyzed parameters are g_{md15} and $C_{gd15} + C_{db15}$ controlling the non-dominant pole at node V_6 (P_6) and $sC_{gd15} - g_{mg15}$ defining the coupling between nodes V_5 and V_6. Gain follows the same increasing trend for both common parameters cases, showing that the impact of W_{15} on this performance is dominant. Furthermore, the increase in gain across the entire analyzed range suggests that g_{mg15} is the dominant parameter and compensates for the smaller increase in g_{md15}. Unity-gain frequency also follows similar trends in either case. While f_0 is dominated by the common path attributes, it can still be increased by up to 20% for the maximum analyzed

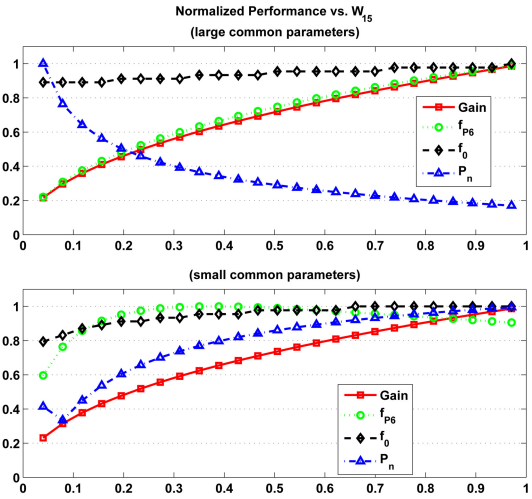

Fig. 4.9 Output block performance trade-offs for comparing circuits AMP_2 and AMP_1 with respect to W_{15} when W_{14} is constant

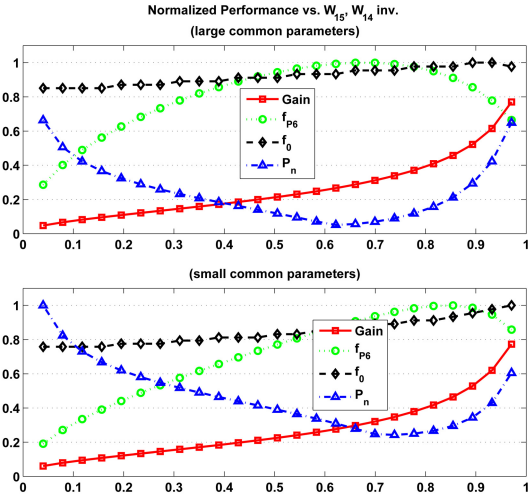

Fig. 4.10 Output block performance trade-offs for comparing circuits AMP_2 and AMP_1 with respect to W_{15} and W_{14} with opposing variation

value of W_{15}. Factoring in the total noise for the output block of AMP_2, the scenario of large common parameters becomes favorable. This allows for increasing gain and f_0, while reducing noise. An optimal point is reached when W_{15} is maximum. For small common parameters, this advantage is lost after the first tenth of the analyzed range, when gain is $\approx 70\%$ less than the maximum. In addition, beyond this point, noise exhibits a more pronounced variation and deteriorates faster than gain increases.

An interesting situation occurs when W_{14} is also varied, but in opposition to W_{15}, illustrated in Figure 4.10. The added parameters of device M_{14} are $g_{md14} + g_{mg14}$ and $C_{db14} + C_{gs14} + C_{gb14}$ influencing the pole at node V_6 (P_6).

As in the previous scenario, a limited variation in unity gain is observed, which increases as W_{15} increases and W_{14} decreases. Similarly, gain increases across the range. However, the impact is now more pronounced in the last third of the interval, when W_{15} is significantly larger than W_{14}. For noise performance, it is now possible to minimize the impact for both large and small common path parameters. However, this minimum is no longer achieved when the gain and f_0 are maximized. This suggests that in this scenario gain is sacrificed in the output block of AMP_2 in order to reduce the noise impact of distinguishing features. Closer inspection shows that the achievable noise minimum is still better for the case of large common parameters.

Overall, the analysis of AMP_2 suggests that compared to the structures of AMP_1 this topology can exploit the distinguishing attributes through different sizing schemes and offer performance improvements.

4.6 Conclusion

This chapter presents a novel circuit synthesis flow based on concept comparison, combination, learning, and re-use. A technique for systematically producing comparison data between two analog circuits is introduced. The comparison data refers to DC-gain, bandwidth, noise, CMRR, and sensitivity. The nodes with similar electric behavior in the two circuits are found through a dual matching approach of circuit topologies and symbolic expressions. Dissimilarities are also identified in the process. Next, the method computes the constraints that relate the electrical behavior to changes of the performance attributes. Using the constraints, the final step produces the comparison data, which includes modification of design trade-offs, availability of free design variables, achievable performance values, and hardness to find optimized design parameters.

Acknowledgements. This work was supported by major collaborative research National Science Foundation CreativeIT Grant no. IIS 0856038.

References

1. Antao, B., Brodersen, A.: ARCHGEN: automated synthesis of analog systems. IEEE Transactions on VLSI Systems 3(2), 231–244 (1995)
2. Castro-Lopez, R., Fernandez, F., Guerra-Vinuesa, O., Rodriguez-Vazquez, A.: Reuse-Based Methodologies and Tools in the Design of Analog and Mixed-Signal Integrated Circuits. Kluwer (2006)
3. Chang, H., Charbon, E., et al.: A Top-Down Constraint Driven Design Methodology for Analog Integrated Circuits. Kluwer (1996)

4. Das, A., Vemuri, R.: Topology Synthesis of Analog Circuits based on Adaptively Generated Building Blocks. In: Proceedings of Design Automation Conference, pp. 44–49 (2008)
5. Degrauwe, M., Nys, O., Dijkstra, E., Rijmenants, J., Bitz, S., Goffart, B., Vittoz, E., Cserveny, S., Meixenberger, C., Van Der Stappen, G., Oguey, H.: IDAC: An Interactive Design Tool for Analog CMOS Circuits. IEEE Journal of Solid-State Circuits SC 22(6), 1106–1116 (1987)
6. De Smedt, B., Gielen, G.: WATSON: Design Space Boundary Exploration and Model Generation for Analog and RF IC Design. IEEE Transactions on CAD of Integrated Circuits and Systems 22(2), 213–224 (2003)
7. Doboli, A., Vemuri, R.: Exploration-Based High-Level Synthesis of Linear Analog Systems Operating at Low/Medium Frequencies. IEEE Transactions on CAD of Integrated Circuits and Systems 22(11), 1556–1568 (2003)
8. Doboli, A., Dhanwada, N., Nunez, A., Vemuri, R.: A Library-Based Approach to Synthesis of Analog Systems from VHDL-AMS Specifications. ACM Transactions on Design Automation 9(2), 238–271 (2004)
9. El-Turky, F., Perry, E.: BLADES: An Artificial Intelligence Approach to Analog Circuit Design. IEEE Transactions on CAD of Integrated Circuits and Systems 8(6), 680–692 (1989)
10. Ferent, C., Doboli, A.: A Symbolic Technique for Automated Characterization of the Uniqueness and Similarity of Analog Circuit Design Features. In: Proceedings of Design, Automation and Test in Europe Conference (2011)
11. Harjani, R., Rutenbar, R., Carley, L.: OASYS: a framework for analog circuit synthesis. IEEE Transactions on CAD of Integrated Circuits and Systems 8(12), 1247–1266 (1989)
12. Hassoun, S., McCreary, C.: Regularity Extraction via Clan-based Structural Circuit Decomposition. In: Proceedings of ICCAD, pp. 414–419 (1999)
13. Horta, N., Franca, J.: Algorithm-driven synthesis of data conversion architectures. IEEE Transactions on CAD of Integrated Circuits and Systems 16(10), 1116–1135 (1997)
14. Hum, R.: Where are the dragons? Presentation at Frontiers in Analog Circuit Synthesis and Verification Workshop (2011)
15. Koh, H., Sequin, C., Gray, P.: OPASYN: a compiler for CMOS operational amplifiers. IEEE Transactions on CAD of Integrated Circuits and Systems 9(2), 113–125 (1990)
16. Li, X., Wang, J., Pileggi, L., Chen, T.-S., Chiang, W.: Performance-Centering Optimization for System-Level Analog Design Exploration. In: Proceedings Design Automation Conference, pp. 422–429 (2005)
17. McConaghy, T., Palmers, P., Steyaert, M., Gielen, G.: Variation-Aware Structural Synthesis of Analog Circuits via Hierarchical Building Blocks and Structural Homotopy. IEEE Transactions on CAD of Integrated Circuits and Systems 28(9), 1281–1294 (2009)
18. Martens, E., Gielen, G.: Classification of Analog Synthesis Tools based on their Architecture Selection Mechanisms. Integration, the VLSI Journal 41, 238–252 (2008)
19. Ramirez-Angulo, J., Carvajal, R., Galan, J., Lopez-Martin, A.: A Free But Efficient Low-Voltage Class-AB Two-Stage Operational Amplifier. IEEE Transactions on Circuits & Systems - II 53, 568–571 (2006)

20. Ramos, J., Steyaert, M.: Positive Feedback Frequency Compensation for Low-Voltage Low-Power Three-Stage Amplifier. IEEE Transactions on Circuits & Systems - II 51, 1967–1974 (2004)
21. Rao, D., Kurdahi, F.: On Clustering for Maximal Regularity Extraction. IEEE Transactions on CAD of Integrated Circuits and Systems 12(8), 1198–1208 (1993)
22. Sripramong, T., Toumazou, C.: The Invention of CMOS Amplifiers Using Genetic Programming and Current-Flow Analysis. IEEE Transactions on CAD of Integrated Circuits and Systems 21(11), 1237–1252 (2002)
23. Stehr, G., Graeb, H., Antreich, K.: Analog Performance Space Exploration by Normal-Boundary Intersection and by Fourier-Motzkin Elimination. IEEE Transactions on CAD of Integrated Circuits and Systems 26(10), 1733–1748 (2007)
24. Tang, H., Doboli, A.: High-Level Synthesis of Delta-Sigma Modulators Optimized for Complexity, Sensitivity and Power Consumption. IEEE Transactions on CAD of Integrated Circuits and Systems 25(3), 597–607 (2006)
25. Tang, H., Zhang, H., Doboli, A.: Refinement based Synthesis of Continuous-Time Analog Filters Through Successive Domain Pruning, Plateau Search and Adaptive Sampling. IEEE Transactions on CAD of Integrated Circuits and Systems 25(8), 1421–1440 (2006)
26. Torralba, A., Chavez, J., Franquelo, L.: FASY: a fuzzy-logic based tool for analog synthesis. IEEE Transactions on CAD of Integrated Circuits and Systems 15(7), 705–715 (1996)
27. Verhoeven, C., et al.: Structured Electronic Design - Negative-Feedback Amplifiers. Kluwer (2006)
28. Wei, Y., Tang, H., Doboli, A.: Systematic Methodology for Designing Reconfigurable Delta Sigma Modulator Topologies for Multimode Communication Systems. IEEE Transactions on CAD of Integrated Circuits and Systems 26(3), 480–496 (2007)
29. Wei, Y., Doboli, A.: Structural Macromodeling of Analog Circuits Through Model Decoupling and Transformation. IEEE Transactions on CAD of Integrated Circuits and Systems 27, 712–725 (2008)
30. Ferent, C., Doboli, A.: Systematic Comparison of Two Low-Voltage Amplifiers using Topology Matching and Performance Constraints. IEEE International New Circuits And Systems Conference, NEWCAS (2012)
31. Ferent, C., Doboli, A.: A Prototype Framework for Conceptual Design of Novel Analog Circuits. In: International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design, SMACD (2012)
32. Kuo, K.-C., Leuciuc, A.: A Linear MOS Transconductor using Source Degeneration and Adaptive Biasing. IEEE Transactions on Circuits & Systems - II 48, 937–943 (2001)