3

Structured design with FOMs

WHAT is structured[1] design? An appropriate meaning of the word "structured" in the context of this chapter is "having a well-defined structure or organization". In this chapter we explore various existing structured design methods for design of analog building blocks (Section 3.1 and 3.2). Most approaches are generic in the sense that they are of use for the design of analog building blocks in general, including oscillators. The exploration of design methods and methodologies allows us to formulate a number of requirements (or better, desired features) of a structured design method for oscillators. None of the reviewed structured design approaches help the designer in an insightful manner during the important design phase of oscillator type and topology[2] selection.

In Section 3.3 a structured design approach is discussed that is based on Figures of Merit (FOMs), which aims at shortening the oscillator design time and it provides qualitative as well as quantitative insight.

FOMs assist the designer in a useful way during the important design phase of oscillator type and topology selection. This phase has a profound influence on the design time and thus on the cost of design. In terms of the analogy between an oscillator designer and a traveler introduced in Section 1.4, FOMs allow the designer to quickly assess whether a possible design route is a high-way, motor-way, country road, desert

[1] From the Latin word "struere"; "to construct". Source: www.dictionary.com.

[2] We use the word circuit topology, or short, topology, in a broad sense in this work: the topology of a circuit represents the way all devices are interconnected but also the type of the devices that are interconnected. Note that in graph network theory often the way to interconnect only is called "topology" [63].

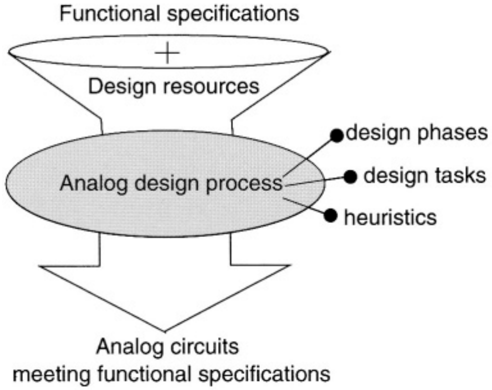

Figure 3.1 Functional specifications and design resources form the input of the analog design process.

path or dead-end street. FOMs are likely to be a very effective element of a FOM-based design methodology. However, a design methodology, a design flow defining each design step for oscillator design is outside the discussed scope of this work; it would require a much more formal approach, and is a huge subject on its own. System level, behavioral level and circuit level modeling play an important role in the formulation of useful FOMs. These hierarchies in modeling and modeling conventions are discussed in Section 3.4. Section 3.5 concludes this chapter with a summary.

3.1 Analog circuit design

Analog circuit design differs substantially from digital circuit design. In the past decades, the design of digital systems has largely been structured and automated. Complex digital systems can be described with a high level language such as the VHSIC Hardware Description Language (VHDL)[3], which can then be synthesized to gate level [64]. This synthesis process is based on well-defined explicit rules. Automatic placement and routing complete the digital design trajectory. The resulting systems can have tens or even hundreds of millions of transistors on one die. The digital designer still has to interact with the Computer Aided Design (CAD) tools in every phase of the design, but in terms of man-hours spent per device, digital design has reached a state of enlightenment compared to analog design.

The design of high-frequency oscillators is an analog design process. This process is illustrated in Figure 3.1. First, the two inputs to this process, functional specifications and design resources are described in Section 3.1.1. In Section 3.1.2, a division of the analog design process in three phases is introduced.

[3] IEEE Standard 1076-1987.

An important reason why the analog design is far less automated and less efficient in terms of man-hours per device, is that it uses a lot of heuristics and rules in contrast to the well-defined rules and boolean algebra of digital design.

In digital design, the redundancy obtained by using only 0's and 1's, makes it relatively easy to divide a large design problem into many small sub-problems. This complexity reduction is a lot tougher for analog design challenges, and one approach is to use heuristics. A number of heuristics used in analog design are investigated in Section 3.1.3.

3.1.1 Functional specifications and design resources

The functional specifications form the design question for the designer who in turn answers with a design meeting these specifications. The majority of functional specifications, denoted $Spec_x$, can be written in the form $Spec_x \in [S_{lb}, S_{ub}]$; the specification is constrained in an interval. The lower bound S_{lb} or upper bound S_{ub} can be $\pm\infty$ or can be expressed in finite tolerances. Furthermore, the interval can be open or closed or be a single value. In other words it can be an inequality constraint or a indexFunctional specifications!equality constraint equality constraint, respectively [65]. Examples are the power dissipation should be lower than 100 mW, the power supply voltage is 3 V ±10%, and the tuning range should start at 1 MHz and extend to at least 1 GHz. In integrated circuits these specifications have to be met over a specified set of environmental conditions like temperature range and with a specified amount of process variation. Most common functional specifications for oscillators are extensively described in Chapter 4.

Design resources are considered to be the items that enable the design process to reach its desired end: an analog circuit that meets the functional requirements. These resources and their range are as follows:

○ **The designer:**

  - very experienced: high salary, difficult to find, exception.
  - inexperienced: not unusual.

○ **Design time:**

  - large: not available in high-tech companies, short product life-cycle.
  - small: common practice.

○ **Technology:**

  - high performance: exotic, not well characterized, costly.
  - low performance: main stream, short turn-around time.

○ **Tools:**

- Design flow:
  - fully automated: nearly bug-free, all verification steps automatic such as back-annotation of parasitics, large initial costs, expensive to maintain.
  - a lot of manual labor: several cheap tools, designer has to perform all interfacing issues between tools in the flow, slow design cycle, small initial cost.
- Architecture and circuit libraries:
  - many: long design history, many designs on the shelf in up-to-date technology, many layout examples, well documented.
  - hardly available: short design history, few design examples in ancient technology, not documented architectures and circuits.
- Front-end (circuit simulators, etc) and back-end (layout, etc.) tools:
  - high performance: short simulation times, many analysis types, costly, expensive computing servers needed.
  - low performance: public domain, limited application range, runs on standard personal computers.

All design resources have a profound influence on the design process, and are implicitly or explicitly specified. A technology is often explicitly specified by the customer or the manager of the designer. The properties of the design resources enable the oscillator design but at the same time limit the set of solutions. A clear example is the technology that sets the practical boundaries. At the same time, without the technology, a practical implementation would not be possible.

> After capitalization, all design resources can be translated into one single resource **"Money"**, by their contribution to the initial costs of a product.

The importance of these initial costs compared to the final costs, depends on the product. For example, the final product may be aimed at the consumer market (high-volume, low profit margin) or at the professional market (low-volume, high profit margin). If the total budget of an oscillator design is no issue, a large number of experienced designers can be used and state-of-the-art technology and tools can be made available.

3.1.2 Design phases

A break-up into three phases of the overall analog design process, the process from specification to tested hardware, is shown in Figure 3.2 (derived from a similar partitioning in [66]).

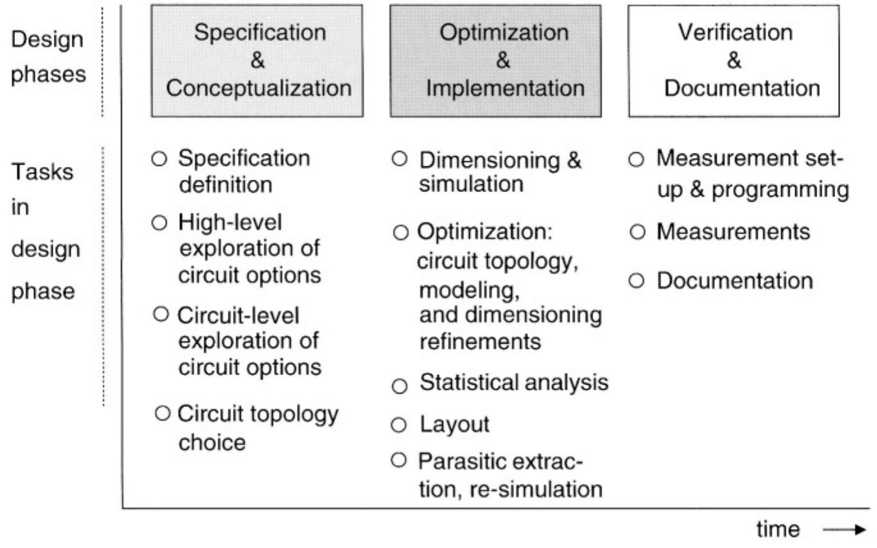
Design phases
Specification & Conceptualization
Optimization & Implementation
Verification & Documentation
Tasks in design phase
○ Specification definition
○ High-level exploration of circuit options
○ Circuit-level exploration of circuit options
○ Circuit topology choice
○ Dimensioning & simulation
○ Optimization: circuit topology, modeling, and dimensioning refinements
○ Statistical analysis
○ Layout
○ Parasitic extraction, re-simulation
○ Measurement set-up & programming
○ Measurements
○ Documentation
time

Figure 3.2 Design phases and tasks in a design process of an integrated analog circuit, such as an oscillator.

The first phase is designated *"specification and conceptualization"*. An important step in this phase is to define the specification unambiguously. Next, an inexperienced designer may start with a literature study to explore circuit options, whereas an experienced designer might instantaneously know a good circuit topology for the job, because he used it successfully in an earlier design job. Behavioral modeling (high-level modeling) can be used advantageously to acquire a good impression of the first order behavior of circuits. Before the second design phase is entered, a circuit topology is selected.

The second phase of the analog design process is the *"optimization and implementation"* phase. All devices are sized, and all specifications are simulated and compared to their target values. In this phase optimization takes place until all specifications are met. The optimization process may include topology changes and modeling refinements, and will certainly involve dimensioning refinements. In order to predict the yield, a statistical analysis usually is part of phase two. When the electrical design of the analog building block is ready, the layout of the circuit has to be made. Interconnect parasitics alter the electrical behavior of the implemented circuit. This can be taken into account by extracting layout parasitics, re-simulating and adjusting the circuit characteristics and the layout, to regain the wanted behavior. The same can be done to include effects of a chip-package. The layout is now ready for mask making and processing.

During and after processing of the analog IC, the third and last phase of the analog design trajectory starts. This phase is designated *"verification and documentation"*.

A measurement set-up or set-ups have to be arranged, and the functionality, performance, yield and reliability of the analog building block have to be verified. If all verification steps match expectation, the documentation can be finalized and the analog design is completed; ready for mass production.

The three grey levels in Figure 3.2 indicate the relative importance of the design phases with respect to the time spent per phase. Phase two is the most dominant phase. Both the novice and the experienced designer spend about 50% to 60% of the total design time simulating, optimizing and realizing the layout[4]. The novice generally spends significantly more time, around 30%, in phase one, compared to the experienced designer (15% to 20 %). Novice designers do not have a large set of previous design examples, circuit options and circuit tricks, which makes it more difficult to reach a first circuit topology. The least amount of time is spent in phase three, about 15% to 20%.

> A bad choice in phase one can have a huge negative impact on the total design time. It forces the designer to repeat phase two several times with many iterations and circuit modifications, until a topology has been found that meets the functional specification.

Obviously, the number of iterations between the phases and between the tasks within a phase should be as small as possible, to minimize the design costs. In other words, although most of the total design time is spent in phase two, design phase one has the dominant influence on the total design time. In practice phase one heavily relies on design heuristics. A number of them are explored in the next section.

3.1.3 Design heuristics

An analog designer makes a number of heuristic decisions during design. This is especially true for the first design phase in Figure 3.2. Several “analog design” heuristics are described by the previously referenced paper of Bowman [66]. An extended and commented list is given below from the perspective of an oscillator designer.

- *Hierarchical circuit design.* The design of an oscillator starts with its specification. At that stage the oscillator is like a black box, of which only the desired properties are known. Next a circuit topology has to be chosen. Finally all devices have to be dimensioned, such that the oscillator meets the specifications. There can be a number of levels between system, circuit and device level, but these three levels already show the natural hierarchy in the design process.

[4] In [66], R. Bowman describes in his paper “Analog integrated circuit design conceptualization” the results of a survey of 75 analog designers. The question was “how much time is spent in each design phase?” A distinction was made between the novice (less than 3 years design experience) and the experienced designer.

- *Specification is prioritized starting with the most difficult specification.* The moment an oscillator designer gets a set of specifications, he begins to assess and prioritize it. For example, if the tuning range should be four decades this immediately attracts his attention. This specification may be the most difficult to achieve and will have the biggest influence in design phase one; see also the next design principle.

- *The first focus is on the few specifications that strongly influence the design decisions made in an early stage of the design process.* To follow up on the example above; if the tuning range of four decades is the dominant one, the designer will focus on this oscillator property first. If this is not realized, the oscillator will not cover the frequency band of interest, and other performance aspects become irrelevant.

- *Rules-of-thumb estimations are applied to simplify the design process.* Common sense aided by experience is an important tool in oscillator design. Part of this common sense, technology awareness and over the years design experience condense in rules-of-thumb. Some examples:

    - A ring oscillator generally has a larger tuning range than an LC oscillator.
    - Bond-wire inductance is about 1nH per mm.
    - A dominant source of $1/f$ noise often is the tail current source of an oscillator.
    - The carrier swing should be maximized.
    - Saturation effects should be avoided.

- *Reuse of knowledge*

    - *Accumulation of circuit options and tricks.* An experienced oscillator designer has made many oscillators for various applications. When he sees the specifications for a new design, he will immediately relate the specifications to a realized design, and circuit options pop up in his mind. During circuit level optimization (phase two in Figure 3.2), the experienced designer opens his backpack of circuit tricks to refine the oscillator design. The backpack of an inexperienced designer still has lots of room.

    - *Optimization and application of circuit tricks are performed at sub-circuit level where only a handful of devices play a role.* A high-frequency oscillator does not usually consist of many transistors. But if a more complex oscillator is designed, the designer usually partitions the oscillator into sub-circuits. This decomposition into (in general) more basic circuit topologies makes the complexity manageable and insight is more easily gained. Most circuit tricks are applied at this level.

- *Similar circuit topologies are used in various technologies.* Oscillator implementations often have the same or similar topologies in different technologies. An LC oscillator implemented with a resonator and a cross-coupled pair or a Colpitts oscillator are basically the same in bipolar, CMOS or, for example, GaAs technology. Of course, this not always the case and can be quite sub-optimal. Each technology also has its own merits and these should be exploited. In bipolar technology a PN-junction varactor may be a good choice, whereas in CMOS technology a MOS-varactor can be preferable, for example.

- *Combinations of known sub-circuits.* Cross-coupled pairs, Colpitts, Hartley, ring oscillators are all quite ancient. On a circuit level, implementations may seem new to an oscillator designer. However, they often are a combination of existing circuit techniques, or are not novel in the sense that they do not add a new circuit to the total pool of known circuit topologies.

The preceding list of design principles and heuristics is certainly not exhaustive. In fact, consciously or unconsciously used heuristics vary from designer to designer. Nevertheless, most common ones are given and the list can be used advantageously to derive a structured FOM-based design approach for oscillators, which blends in with common design practice.

3.2 Structured and automated design methods

Circuit analysis using CAD tools like Spice [67], Pstar[5], Spectre [68l or Aplac[6] is indispensable in modern analog circuit design. As mentioned, most of the design time is spent on computer-aided circuit analysis (part of phase two in Figure 3.2).

A number of structured design methods, heavily based on computer-aided circuit analysis, is discussed in this section.

The list of structured methods illustrates the state-of-the-art in analog design automation. Analog design automation, like the FOM-based structured design of oscillators that will be described in Section 3.3, aims at enhancing the productivity of the designer. Once automated, a design task can be shortened by adding more computing power. The structured methods highlighted in the following sections are applicable to analog circuits in general, including oscillators.

[5] Philips' in-house circuit simulator.

[6] Aplac is a circuit design and simulation tool. See http://www.aplac.com.

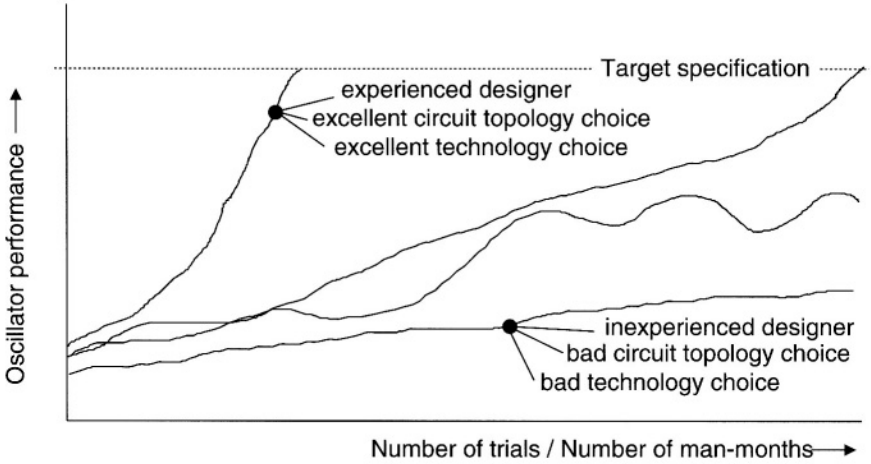

Figure 3.3 Example of possible outcomes of the trial-and-error design process.

3.2.1 Trial-and-error

Every designer is familiar with structured design. The design strategy called "trial-and-error", "cut-and-try" or "turning the knobs" is apparently very important since it is wide-spread. One (or somewhat more unstructured: more than one) design parameter is varied and circuit performance is monitored. If the performance improves, the value of the design parameter is increased until a specific functional specification is met or other design aspects start to move away from the functional specification. All design parameters can be varied in this way until the total circuit performance meets the functional specification. The fact that this process can be performed in a mathematically structured way is demonstrated by today's circuit optimizers, which use sophisticated algorithms in their pursuit of the design requirements. As with any method, practical limits of the chosen circuit topology and technology may prevent the circuit performance from ever reaching the functional specification.

Figure 3.3 illustrates the trial-and-error design process. In combination with the knowledge of an experienced designer, and a good circuit topology and technology choice, the "trial-and-error" method can be a fast way to complete an oscillator design. This case is represented by the upper trace in Figure 3.3. On the other hand, an inexperienced designer may never reach the target specification, regardless of the number of trials.

Arguably, whatever design method will be used for the oscillator design trajectory, there will be always trail-and-error elements in it. The ideal case of a design process of a high-frequency oscillator without any iteration, with an optimally sized circuit in only one simulation run, is not realistic[7]. However, as pointed out by Figure 3.3, with-

[7] Unless it is a copy of an earlier design.

out making sensible circuit and technology choices, this design method has distinct disadvantages:

○ Little or no insight is gained. For every new design the same dimensioning process starts over again. No reuse is made of previously gained experience. Only the increased experience, gained from the final results of a new design, as added value compared to the previous design process. Thus, the reuse of earlier design experience, and the extraction of generic reusable knowledge during a design is very low for trial-and-error design.

○ A very large number of time consuming and costly iterations may be necessary. As illustrated in Figure 3.3, a bad topology choice results in a long design time.

○ Possibly, the design process may not converge.

○ The design knowledge built up by the designer is difficult to transfer and maintain within the design team. When the experienced designer leaves the design group and is replaced by an inexperienced designer, the design time increases a lot.

3.2.2 Optimization tools

Once an oscillator circuit topology is chosen and all devices are given an initial value, automatic optimization tools can be used.

> Automatic optimization tools can perform the tedious task of adjusting all design parameters until specification is met under all process and temperature conditions.

Most electronic design automation (EDA) tools or circuit simulators have an optimizer built in or have an interface to an optimization tool. Examples are "Cadence Analog Circuit Optimizer[8]", the optimizer in Aplac [70], and Adapt[9].

A simplified flow chart of automated circuit optimization is shown in Figure 3.4. The key to the optimization process is the optimization algorithm, which determines in what direction and how much the design values are adjusted, prior to new simulation runs. Examples of used algorithms are simulated annealing [72], genetic optimization [73], minimax optimization [74], and Nelder-Mead optimization [75]. There are a few interesting optimization approaches specifically for LC oscillator optimization. For example, geometric programming can be used to optimize LC oscillators [76]. Another approach uses a graphical optimization method for the same purpose [183].

Similar to manual trial-and-error design, the optimizer may not find a solution to the design problem. Several causes can prohibit a solution. The specification may be unrealistic or the circuit topology in combination with a chosen technology can not

[8] http://www.cadence.com. The EDA market (in 2001) is dominated by Cadence Design Systems. Cadence has 70% market share, Mentor Graphics 20 %, Avant! 8% and others 2% [69].

[9] Philips' in-house circuit optimizer [71].

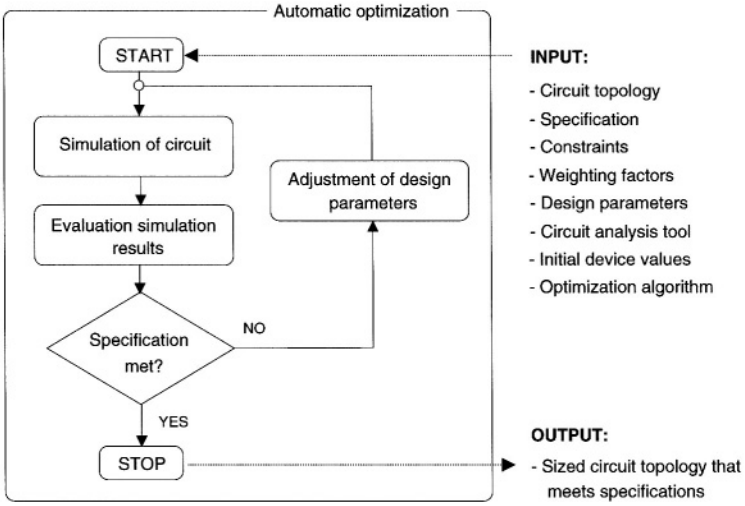

Figure 3.4 Simplified device-sizing flow using automatic circuit optimization.

meet the specification. In other words, the solution to the optimization problem lies outside the practical solution space. Assuming that the specification is realistic and cannot be lowered, the optimization process then has to be reentered with a refined initial topology or with an entirely different circuit topology. Of course, a carefully selected circuit topology choice can prevent or reduce the number of iterations, and will also yield a faster optimization process. Another reason why an optimizer may fail to find a solution, is simply because it can not find it. An optimization method can only search through part of the solution space, and can therefore overlook a solution.

An important remark should be made about the analysis types used for simulating a circuit in the optimization process. The circuit simulator should be able to simulate all specifications[10]. This is not always trivial. For example, simulation of the noise behavior of oscillators is generally not supported by standard circuit analysis types, such as DC, AC, or transient analysis. It is only recently (in the late 90's), that phase noise analysis for oscillators has become available in circuit optimizers. An example is the phase noise analysis implemented in the commercially available circuit simulator SpectreRF [77].

3.2.3 Expert systems and synthesis environments

Expert systems and synthesis environments not only aim to automate and shorten the simulation and optimization task, but also to include topology selection and/or layout

[10] Either that or we need an accurate model to predict the performance of a specific oscillator property.

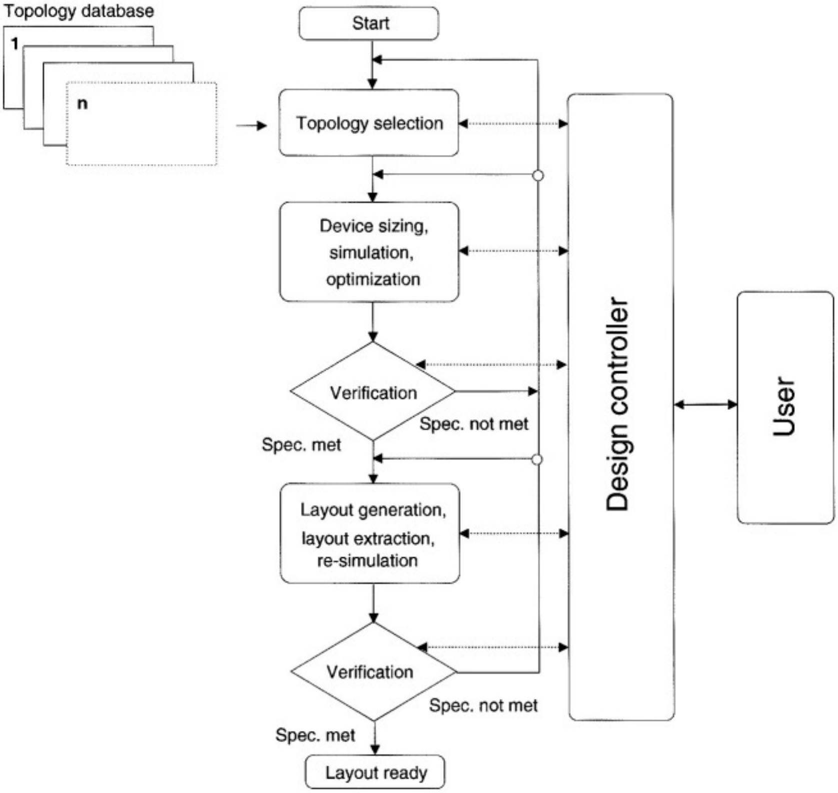

Figure 3.5 Simplified hierarchical design flow of expert systems and analog synthesis environments.

generation. A graphical representation of the most common elements of analog design expert and synthesis systems is shown in Figure 3.5.

> A distinctive difference of an expert system compared to automatic optimization tools is the presence of a topology selection mechanism, which chooses a topology from a topology library.

However, in general this library is small and the selection criterion is as simple as selecting the next topology lined-up if a first topology fails to meet the functional specification. Once the topology is selected the optimization process starts, described in the previous section, resulting in a sized and verified topology. To complete the automatic design flow, layout generation, extraction of parasitics, re-simulation and verification can be part of a synthesis environment. The whole design process and interaction with the user is controlled by the design manager/controller.

There are many publications on analog design optimization environments aiming to implement an automated design flow similar to the one in Figure 3.5. Detailed literature overviews are given in [65,78–80]. All published environments have intriguing names and, interestingly, most often demonstrate their capabilities on operational amplifier (OPAMP) design. Normally, these systems are not available in the public domain or commercially either. A few representative examples follow.

○ BLADES [81].
This automated design approach for analog circuits is expert system based. The use of the system is limited to OPAMPs and it does not include layout generation. A so-called circuit design manager is the design engine of the system. One part of the manager is the circuit topologizer. This is a configuration processor that generates circuit topologies based on sub-circuits present in the knowledge database of BLADES.

○ OPASYN [82].
This synthesis framework is also dedicated to OPAMPs. It includes layout generation. In contrast to BLADES, topologies are not generated from sub-circuits, but selected from a small topology database. The design parameters of the selected circuit topology are first optimized using an analytical model. This is relatively fast but the accuracy is limited. After this, the results are verified and optimized with full accuracy models.

○ AMGIE [65].
This synthesis environment for CMOS analog integrated circuit covers all functionality shown in Figure 3.5. AMGIE can only design circuits that are present in the so-called cell library. This library can contain both standard and custom cells. Designers can add to this library, to extend the possibilities of the synthesis environment. Like OPASYN, device sizing is first performed on analytical expressions of a selected topology, followed by a verification and optimization with accurate device models. After device sizing the layout is automatically generated by the performance-driven place-and-route tool LAYLA [83].

○ CYCLONE [84].
This synthesis tool is an automated layout-aware RF LC-oscillator design tool, capable of delivering an optimized LC-oscillator implementation, starting from specification and including layout. It provides finite element simulations and optimization of RF coils, as well as VCO circuit sizing. Module generation, placement and routing is done within the mentioned tool LAYLA. Two common RF LC oscillator topologies are incorporated. The tool does not support ring oscillator design.

3.3 FOM-based structured design

The overview of structured and automated analog design methods, given in Section 3.2, shows that a number of tools are available, which can shorten the analog design phases pictured in Figure 3.2. However, no structured design method or automated design tool was found in literature that can handle the wide range of LC and ring oscillators within the scope of this work. Furthermore, a practical, insightful and appealing method for topology selection, which has a major influence on the total design time, was not encountered.

In this section FOM-based structured design is described, which aims at reducing the number of iterations between the design phases and design tasks of an oscillator design. FOMs, once formulated, yield fast and insightful oscillator type and topology selection. Moreover, FOMs diminish the disadvantages of the trial-and-error design method, while incorporating design heuristics commonly used by experienced designers.

3.3.1 Structured design requirements

A number of requirements –or maybe better "desired features"– for structured oscillator design can be formulated, keeping in mind the common analog design practice and available tools described in the preceding sections. To be a real guide for the oscillator designer, the structured oscillator design should:

- ○ **Strive for orthogonality.** The structured design method should tell us how the oscillator design parameters (the degrees of design freedom, like quality factor, tail current, number of ring oscillator stages, etcetera) are linked to oscillator properties (frequency, phase noise, etcetera), which determine the performance. If possible, oscillator properties should be orthogonalized so that they can be optimized independently. For high-frequency oscillators complete orthogonalization is less desirable from a performance point of view. In that case, qualitative and quantitative insight into the dependency between parameters and relevant oscillator properties is the next best thing.

- ○ **Fit in a hierarchical approach.** As pointed out, the design process of oscillators already has a natural hierarchy. Within the hierarchy, the number of iterations should be kept to a minimum.

- ○ **Connect with designers.** By focusing on common circuit parameters and definitions to formulate a structured design approach for oscillators, the chances of being useful and appealing to oscillator designers are maximized.

- ○ **Provide qualitative insight.** This helps the designer to prioritize the oscillator specification, to gain insight into circuit options and tricks, and to acquire awareness about the most important oscillator design parameters.

- ○ **Provide quantitative insight.** Since design requirements are in the form $Spec_x \in [S_{lb}, S_{ub}]$, the design method should help the designer to determine oscillator performance relative to quantitative borders. Furthermore, quantitative insight is needed for the oscillator topology selection.

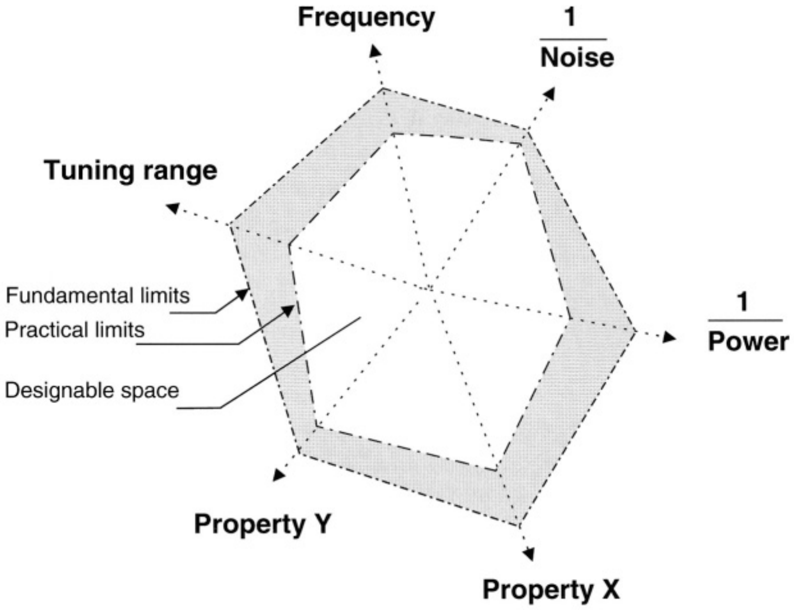

Figure 3.6 Simplified solution space of all oscillator designs with an oscillator property assigned to each axis. The dashed-dotted lines connecting the various properties represent the fundamental en practical limits of property combinations.

○ **Lead to a shorter oscillator design time.** The benefits of using a particular structure design method for oscillator should result in a shorter oscillator design time.

In the following section, structured design with FOMs will be highlighted, which addresses the above requirements.

3.3.2 Figures of merit

> The design space of an oscillator can be visualized by an N-dimensional hyper-cube, in which each oscillator property is assigned to a separate axis [66].

In Figure 3.6, a simplified two-dimensional representation of this hypercube is shown [85]. It is very important to realize that this is a highly suggestive representation in two dimensions of a higher dimensional solution space. The multiple axes connected by lines are drawn to illustrate the bounded character of combinations of properties of the solution space by fundamental limits and practical limits, and by no means represent actual relations between the properties. Certain combinations of oscillator properties will be impossible to realize in an oscillator design.

One oscillator property is assigned to each axis. The arrow at the end of each axis points in the direction of increased performance of an oscillator property. In the direction of the arrow it is increasingly difficult for an oscillator designer to achieve the required performance of one of the properties. Six axes are shown including the most important oscillator properties, but there are more, as there are more properties.

Design solutions always are inside the fundamental boundaries and practical boundaries. The area inside the practical limits can be regarded as designable space [66].

Fundamental limits refer to limits imposed by nature as we know it, for example the speed of light c or the charge of an electron q. All the limits set by nature impose fundamental limits on the oscillator performance. Practical limits are imposed by the design resources, like technology. The IC technology used for oscillator implementation has non-ideal devices and properties, such as finite transition frequencies, interconnect capacitance, etcetera. These non-ideal elements limit practical oscillator performance.

In Figure 3.7 a solution –that is, an oscillator design– is shown within the designable space, which meets the functional specification. There are two other possibilities. If the functional specification is beyond the fundamental limits, there is no oscillator design that can meet this specification. If the functional specification is beyond the practical limits, there is no oscillator design that can meet the functional specification, given the IC technology used for implementation. As another IC technology may be much more advanced, practical limits can shift towards the fundamental limits. Hence, usage of a more advanced technology can bring the oscillator specification back into the designable space.

Functional specifications quantitatively specify the performance of every oscillator property of interest. If the definition of all specifications (these are discussed in Chapter 4) is unambiguous, this clearly defines what is expected of an oscillator design. For example, if the frequency range specification is from 1 GHz to 2 GHz, the oscillator requirement concerning the tuning range of the oscillator is completely clear to the oscillator designer. This is not the case for the fundamental and practical limits. It is very hard to prove where the practical and fundamental limits are exactly located [11] for an oscillator property (e.g. one of properties on the axis in Figure 3.7). However, to meet the requirements we defined for structured oscillator design, we need to have quantitative information about the oscillator solution space.

Figures of merit provide quantitative information about the performance of one or more oscillator properties. By combining more than one property, a FOM can reduce the dimension of the designable space for oscillators. In other words, FOMs can

[11] For example, we would need to know the global optimum of an oscillator design in a certain technology to have the practical limit of the topology/technology-combination quantitatively available.

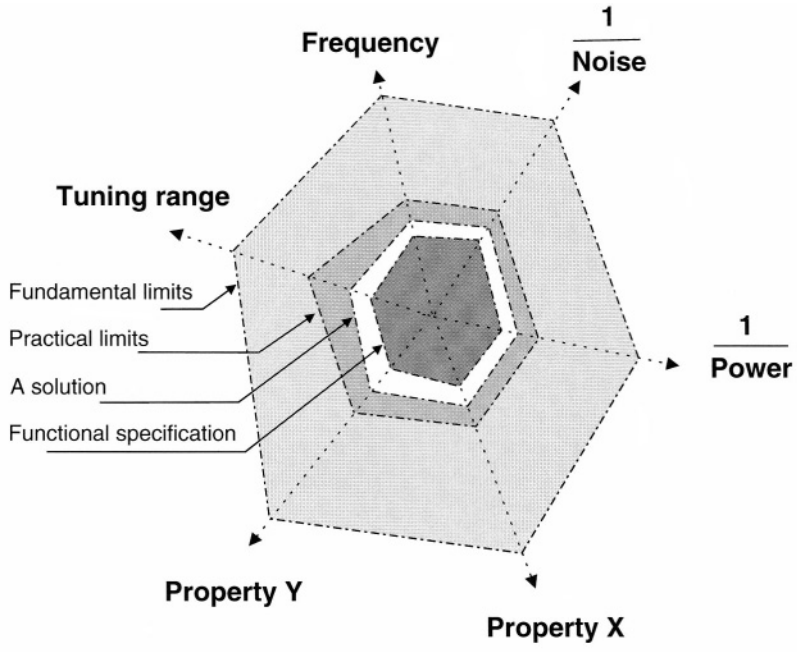

Figure 3.7 Solution space of all oscillator designs. Within the practical limits there is a solution that meets the functional specification.

be regarded as hyper-planes in the N-dimensional hyper-cube, which represents the designable space. Therefore FOMs can limit the solution space making it easier for the designer to find a oscillator implementation that meets the functional specification. It is useful to distinguish between two types of FOMs, design FOMs and benchmark FOMs.

Design FOMs

> Design FOMs provide the oscillator designer with qualitative and quantitative insight into oscillator properties. They predict the design margins for oscillator specifications, and show the trade-offs involved in these margins.

Design FOMs are defined as the quotient of a function f with the design parameters $p_1 \ldots p_m$ as arguments, and a functional specification, represented here by a function g with oscillator properties $q_1 \ldots q_k$ as arguments. Function f, is an analytical function that estimates, when evaluated, the oscillator performance in terms of the functional specification represented by g, making use of behavioral level modeling for example.

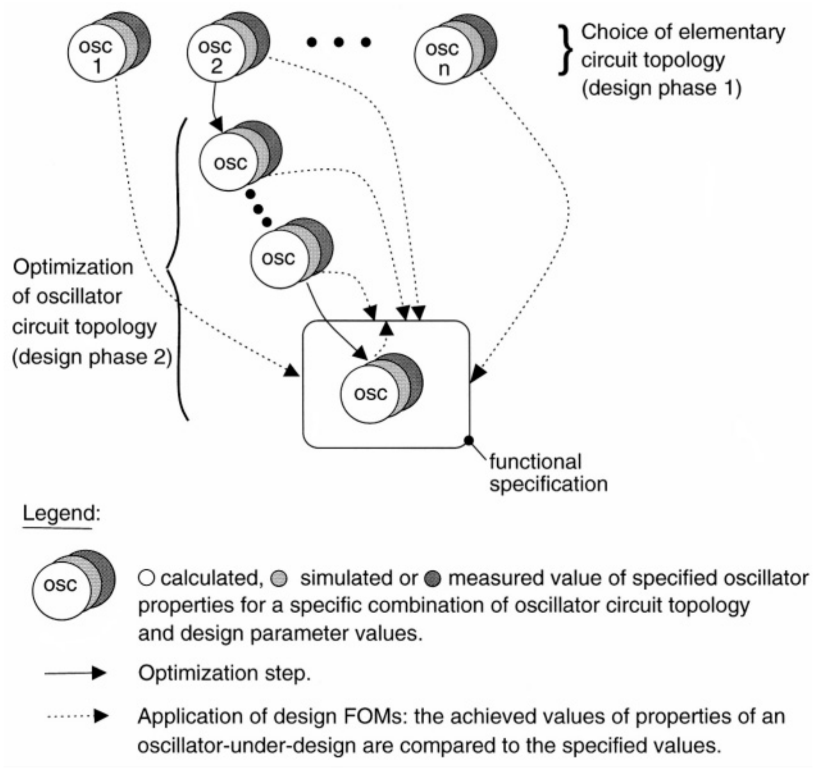
osc 1
osc 2
osc n
Choice of elementary circuit topology (design phase 1)
osc
osc
osc
Optimization of oscillator circuit topology (design phase 2)
functional specification
Legend:
osc
○ calculated, ● simulated or ● measured value of specified oscillator properties for a specific combination of oscillator circuit topology and design parameter values.
Optimization step.
Application of design FOMs: the achieved values of properties of an oscillator-under-design are compared to the specified values.

Figure 3.8 Illustration of the use of design FOMs during the design process. Design FOMs are used to select the most promising circuit topology. During subsequent optimization steps, design FOMs indicate the distance with respect to the functional specification: the design margin.

Alternatively, the function *f* may simply be the simulated or measured oscillator performance, in terms of the functional specification represented by *g*. It follows that a design FOM always is dimensionless. In summary, a design FOM is defined as

$$FOM_{design} = \frac{f_{ana|sim|meas}(p_1 \cdots p_m)}{g_{spec}(q_1 \cdots q_k)}. \tag{3.1}$$

Figure 3.8 illustrates the application of design FOMs during the design of an oscillator. Design phase one and two from Figure 3.2 on p. 41 are represented in Figure 3.8. In design phase one, a choice has to be made between numerous possible circuit topologies. In design phase two, the design parameters of the chosen circuit topology have to be optimized, which can include circuit topology modifications or choosing an entirely new topology, such that the functional specification is met. Design FOMs provide the designer with quantitative information on the performance of an oscillator property with respect to its specified value, and are thus useful in design phase one and two. In dBs, if the realization[12] of a design FOM is negative, the functional specification is not met, if it is 0 dB the specification is exactly met, and if it is positive the amount of positive design margin is given in dBs. All relevant design FOMs for an oscillator will therefore always be zero or positive (when expressed in dBs) when the electrical design of the oscillator is ready. This situation is illustrated in Figure 3.8 by the final design, which is drawn inside the rectangle that represents the functional specification.

Benchmark FOMs

Important design questions like: “how does my realized oscillator design compare against the state-of-the-art, or to a theoretical performance boundary ?”, are answered by benchmark FOMs.

Benchmark FOMs normalize performance aspects of an oscillator, calculated by function g, to allow a fair comparison with other oscillator designs, or to a theoretical performance limit. This type of FOM is defined as the quotient of a function g with the oscillator properties $q_1 \dots q_k$ as arguments, and a normalizing function h with arguments design parameters $p_1 \dots p_m$. Function g is a function of one or more oscillator properties. As an equation we can write

$$FOM_{benchmark} = \frac{g_{performance}(q_1 \cdots q_k)}{h_{norm}(p_1 \cdots p_m)}. \tag{3.2}$$

The application of benchmark FOMs is illustrated in Figure 3.9. As mentioned, benchmark FOMs are used to compare performance aspects of a realized oscillator design with other designs (for example designs found in literature). For example, the normalization function h normalizes the measured phase noise performance for certain design parameters, such as, power and oscillation frequency. This can be considered relative benchmarking. Alternatively, the normalization function h in a benchmark FOM may be a theoretical performance limit. In this case the performance of an oscillator is benchmarked in an absolute sense to a theoretical bound. For example, the normalization function may relate to the best achievable phase noise performance. Similar to relative benchmarking, absolute benchmarking can be used to compare realized oscillator designs. However, unlike relative benchmarking, absolute benchmarking can also be useful during design phase one: to check whether the specified performance is possible to achieve.

[12] The result of the calculation of a FOM with all arguments given a value.

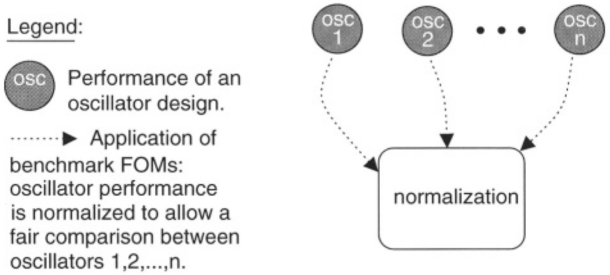
Legend:
OSC Performance of an oscillator design.
Application of benchmark FOMs: oscillator performance is normalized to allow a fair comparison between oscillators 1,2,...,n.
OSC 1
OSC 2
OSC n
normalization

Figure 3.9 Illustration of the use of benchmark FOMs. Performance aspects of a number (n) of oscillator designs are compared after normalization of this performance.

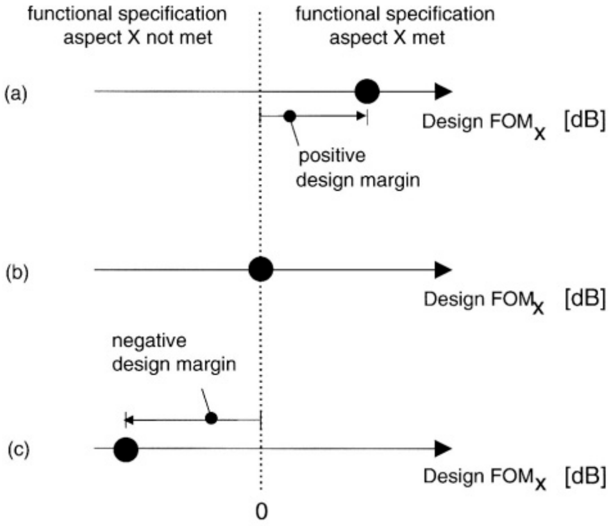

Figure 3.10 Three possible realizations of a design FOM.

Benefits of using FOMs

Figure 3.10 illustrates the use of design FOMs. In this figure, a design FOM, which covers a certain functional specification, is set out on one axis. Three possible locations of the design FOM are shown in Figure 3.10(a), (b) and (c), respectively:

(a) A realization of a design FOM has a positive value. The functional specification X covered by this FOM (for example tuning range) is met and the FOM value indicates the positive design margin.

(b) A realization of a design FOM is 0 dB. Specification aspect X is met and there is no design margin.

(c) A design FOM realization is negative. Specification aspect X is not met: the design margin is negative. If best case values of design parameters $p_1 \dots p_m$ are used in the function f of the design FOM, the oscillator class or topology from which the function f is derived can be can be discarded by the oscillator designer.

Clearly, the situation pictured in Figure 3.10(c) is the most powerful one, assuming best case values are used to calculate the design FOM realization. If the design FOM realization is negative, no feasible oscillator circuit topology exists that is covered by the model, which was used to derive the function f in the design FOM. The oscillator designer can move on to explore more promising solutions to the design problem. The sooner the designer takes a 180 degrees turn in a dead-end street the better, as that saves time. Once the designer has verified that a solution exists with positive design FOMs for each performance aspect, this design can be further optimized. Apart from the ability of design FOMs to guide the designer in the design process, they also provide a great means for transferring and maintaining design knowledge.

> Once a useful design FOM is defined, it enables inexperienced designers to make design decisions as quickly as the experienced designer.

The experienced designer who has formulated the FOM may have left the design group:

> FOMs serve as documentation to transfer qualitative and quantitative design insight.

In the previous section the implicit assumption has been made, that a negative realization of a design FOM with best-case design parameters always is beyond practical limits. In other words, the design FOM overestimates practically achievable performance. The above assumption represents the ideal case: the designable space is limited by the FOMs, and not a single solution is discarded that could potentially meet the functional specification. For many FOMs this ideal case may be valid: if the function f in the FOM neglects effects or parasitics encountered in reality, this FOM will always overestimate practically achievable performance. However the above assumption does not necessarily hold, and is too restrictive. In general, FOMs can limit the designable space but at the cost of discarding potential solutions: part of the actual designable space is not searched. Provided that the FOMs guide the oscillator designer to solutions within the designable space bounded by the FOMs, this is acceptable.

The benefits of using benchmark FOMs are clear: without normalization of oscillator properties like oscillation frequency, power consumption, phase noise performance, etcetera, it would be impossible to make a fair comparison between various realized oscillator designs. In addition, a benchmark FOM can be very useful to a designer during design if the FOM performs an absolute benchmarking. If a realization of this benchmark FOM is positive (the specified, simulated or measured value is greater than the theoretical bound), the designer knows that the given specification is impossible to achieve, or that the simulation or measured value is inconsistent (i.e. highly unlikely).

In Chapter 7 a number of examples will be given of design FOMs and benchmark FOMs.

Clearly, the most difficult step in defining FOMs is the derivation of suitable analytical f-functions in design FOMs or analytical h-functions in benchmark FOMs. To derive a useful function f an oscillator model is required. Once a function based on this model is defined, its accuracy can be assessed by comparing the calculated FOM value with simulated or measured values substituted in the FOM, instead of the realization of a calculated f. To develop the design insights and required knowledge for useful FOMs, the oscillator design space is explored in Chapter 5 and 6. Some conventions and several modeling levels used in these chapters are discussed next.

3.4 Modeling framework

In Section 3.4.1, system level equations will be introduced, which will be used in Chapter 6 to investigate phase noise in linear oscillator models. Section 3.4.2 introduces several behavioral building blocks used in Chapters 5 and 6 . Analysis of the models constructed with these building blocks will lead to the definition of several design FOMs and benchmark FOMs for LC and ring oscillators. A brief discussion on circuit level modeling in Section 3.4.3 concludes the modeling aspects used in this book.

3.4.1 System level modeling

In Chapter 6 we will learn that there are several mechanisms that give rise to the oscillator phase noise sidebands discussed in Chapter 2. Many phase noise generation mechanisms arise from nonlinear oscillator aspects. However, noise shaping of noise sources in an oscillator is always present even in the absence of nonlinearities.

Noise shaping leads to phase noise and can be calculated using a linear feedback system model [26]. This model is reviewed and extended in this section.

The influence of oscillator nonlinearities on the accuracy of the noise shaping equations will be dealt with in Chapter 6.

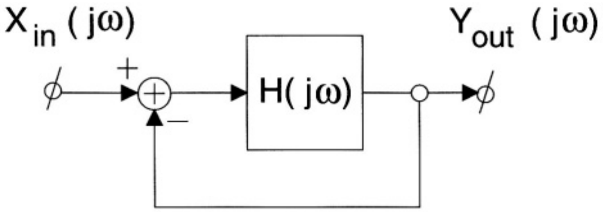

Figure 3.11 An linear oscillatory feedback system with unity feedback.

Single stage feedback system

Consider the linear feedback system in Figure 3.11. Output $Y_{out}(j\omega)$ is the spectral density of the oscillator and input $X_{in}(j\omega)$ is a noise power density, arising from white noise sources in an oscillator for example. The system has unity feedback and an open loop transfer function $H(j\omega)$. The closed loop transfer function is given by

$$\frac{Y_{out}}{X_{in}}(j\omega) = \frac{H(j\omega)}{1+H(j\omega)}. \tag{3.3}$$

When $H(j\omega) = -1$ at $\omega = \omega_{osc}$, the system obeys the oscillation conditions stated in Chapter 2. For frequencies close to the oscillation frequency $\omega = \omega_{osc} + \Delta\omega$, a Taylor expansion can be performed on $H(j\omega)$. Neglecting higher order terms, $H(j\omega)$ can be approximated as

$$H(j\omega) \approx H(j\omega_{osc}) + \Delta\omega\frac{dH(j\omega_{osc})}{d\omega}. \tag{3.4}$$

Substitution of $H(j\omega_{osc}) = -1$ and (3.4) in (3.3), and taking its absolute value leads to [26],

$$\left|\frac{Y_{out}}{X_{in}}(j(\omega_{osc}+\Delta\omega))\right|^2 = \frac{1}{(\Delta\omega)^2\left|\frac{dH(j\omega_{osc})}{d\omega}\right|^2}, \tag{3.5}$$

in which $|\Delta\omega\, dH(j\omega_{osc})/d\omega| \ll 1$ has been assumed. The term $dH(j\omega_{osc})/d\omega$ can be expanded by defining $H(j\omega) = A(\omega)exp[j\phi(\omega)]$. Now (3.5) can be rewritten as [26],

$$\left|\frac{Y_{out}}{X_{in}}(j(\omega_{osc}+\Delta\omega))\right|^2 = \frac{1}{4Q^2}\left(\frac{\omega_{osc}}{\Delta\omega}\right)^2, \tag{3.6}$$

with a quality factor Q defined as

$$Q|_{\omega=\omega_{osc}} = \frac{\omega_{osc}}{2}\sqrt{\left(\frac{dA}{d\omega}\right)^2 + \left(\frac{d\phi}{d\omega}\right)^2}, \tag{3.7}$$

where $A = |H(j\omega)|$ and $\phi = \arg(H(j\omega))$.

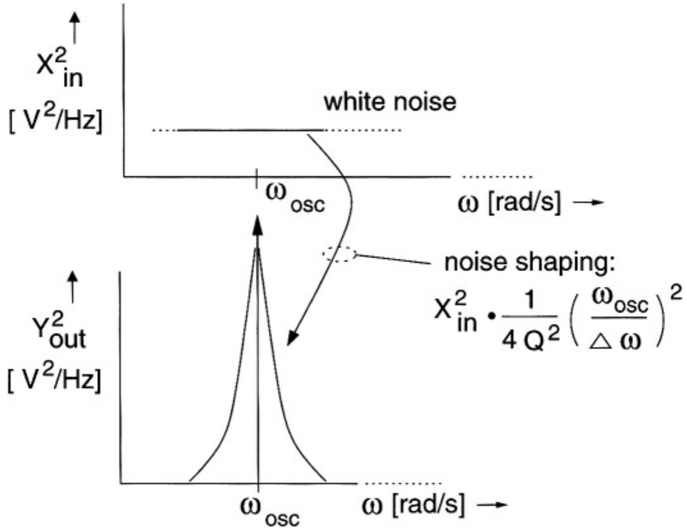

Figure 3.12 White noise is selectively amplified by the closed loop gain of the oscillator, and shaped into phase noise.

Equation (3.6) shows that a noise power density X_{in} will be selectively amplified by the closed loop gain of the oscillator. This process is illustrated in Figure 3.12. The white noise is shaped into phase noise sideband around the carrier atfrequency ω_{osc}.

It is instructive to note that the quality factor definition in (3.7) reduces for a parallel LC resonator at resonance to

$$Q_p = \frac{\omega_{osc}}{2}\left|\frac{d\phi}{d\omega}\right|, \tag{3.8}$$

since $dA/d\omega = 0$ at resonance. Above formula is equivalent to the Q_p definition in (2.13) on p. 24. However, $dA/d\omega$ is for many oscillator types and implementations nonzero, and then the more general Q definition in (3.7) should be used. For example, as we will see in Chapter 6, $d\phi/d\omega$ is zero for the two-integrator oscillator and $dA/d\omega$ dominant is therefore.

Feedback system with *N* identical stages

As shown in the oscillator classification in Chapter 2, many oscillators of interest consist of more than one stage. Specifically, I/Q oscillators have two identical stages in general. If more than two output phases are required, more stages can be added. For N-stage oscillators with identical stages, it is practical to express the quality factor definition in (3.7), in terms of the transfer function of one stage. This simplifies calculation of the quality factor of an N-stage oscillator.

Consider the N-stage linear feedback system shown in Figure 3.13. The system has unity feedback and the open loop transfer function $H_{N-stage}(j\omega)$ is a cascade of N

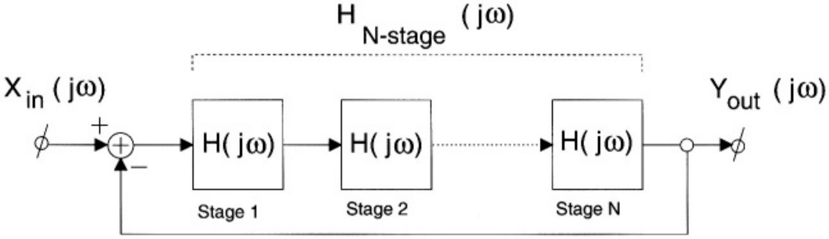

Figure 3.13 An N-stage oscillatory linear feedback system.

identical stages with transfer function $H(j\omega)$. Obviously, the same calculations and assumptions for this system can be made as for the single stage oscillatory feedback system. Hence (3.6) is also valid for this system, provided that (3.7) is redefined. For the system model in Figure 3.13, A in (3.7) should be redefined as

$$A_{N-stage} = |H(j\omega)|^N, \tag{3.9}$$

and ϕ in (3.7) should be redefined in as

$$\phi_{N-stage} = N \cdot \arg(H(j\omega)). \tag{3.10}$$

Substitution of (3.9) and (3.10) in (3.7) yields the quality factor of an N-stage oscillator in terms of sub-system $H(j\omega)$,

$$Q_{N-stage}\big|_{\omega=\omega_{osc}} = \frac{\omega_{osc}}{2}\sqrt{\left(\frac{dA_{N-stage}}{d\omega}\right)^2 + \left(\frac{d\phi_{N-stage}}{d\omega}\right)^2}, \tag{3.11}$$

which can be rewritten in terms of A and ϕ as,

$$\boxed{Q_{N-stage}\big|_{\omega=\omega_{osc}} = N \cdot \frac{\omega_{osc}}{2}\sqrt{\left(A^{N-1} \cdot \frac{dA}{d\omega}\right)^2 + \left(\frac{d\phi}{d\omega}\right)^2}. \qquad (3.12)}$$

Since (3.12) is derived at system level, analyzing $H(j\omega)$ as a black box, (3.12) can be used to assess the noise shaping in N-stage LC and ring oscillators.

3.4.2 Behavioral level modeling

A behavioral model captures part of the behavior of an oscillator circuit, in a model using network elements[13]. The abstraction of an oscillator provided by behavioral modeling is very powerful for several reasons. First of all, calculations on behavioral

[13] Network elements are ideal models of resistors, capacitors, etcetera [86].

models are relatively simple and give insight into the first order behavior of the oscillator. This, of course, requires insight into what oscillator properties are dominant and what properties can initially be neglected, when the behavioral model is constructed. Second, a behavioral level is partially circuit topology independent. Therefore, design FOMs derived with a behavioral model are relevant for a large pool of circuit topologies. To define oscillator behavioral models unambiguously, behavioral building blocks used in this book are described in the following section. Next, conversion from a differential circuit level model to a single-ended behavioral model is discussed.

Behavioral modeling building blocks

In behavioral models, the same symbols for passive network elements are used as in circuit level modeling. However, in circuit level modeling these symbols model physical components. On a behavioral level, passive network elements used in the behavioral models like resistors, inductors, capacitors and varactors, are ideal and noiseless. If an inductor, capacitor, varactor or LC resonator has losses, these losses are explicitly introduced in the behavioral model by adding one or more resistors. In addition, resistors are by convention in this work noiseless in behavioral models. All noise sources will be represented by separate noise current sources, or by one noise current source that represents all noise contributors.

In addition to passive network elements, a number of symbols that represent ideal transconductors, switches, phase shifters, etcetera are required. The symbols used in this work, and their meaning or equivalent circuit are listed in Appendix B. We have already encountered several examples of behavioral models in Chapter 2, for example Figure 2.7(a) on p. 24 and Figure 2.17 on p. 35.

Differential circuits and single-ended behavioral models

An oscillator in an integrated transceiver is only one of the many building blocks of a system on a chip. For maximum robustness and minimum interference generation, an integrated oscillator generally is a differential structure. Since, a differential circuit is truly symmetrical, assuming perfect matching and a symmetrical layout, calculations on one half of the circuit are sufficient to capture its behavior (half-circuit concept [87]). Hence, it is efficient to use single-ended behavioral models, instead of differential behavioral models, to represent differential oscillator circuit topologies.

Obviously, equations and FOMs derived from a single-ended behavioral model should be identical to the results obtained using its differential counterpart, and vice versa. Figure 3.14(a) shows the circuit diagram of a differential LC oscillator. The transconductance of each transistor is g_{m_Q} and the generated output current noise is $i_{n_{diff}}$. The single-ended behavioral model of this oscillator is shown in Figure 3.14(b). Simple calculation shows that the transconductance of the cross-coupled pair is equal to $-g_{m_Q}/2$.

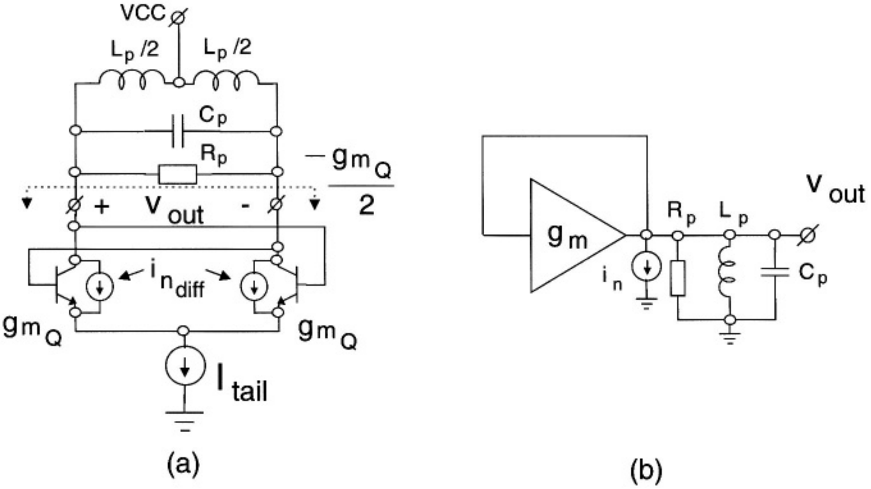

Figure 3.14 Differential LC oscillator circuit diagram (a), and its single-ended behavioral model (b).

The transconductance of the single-ended model , expressed in terms of g_{m_Q}, is equal to

$$g_m = \frac{g_{m_Q}}{2}. \tag{3.13}$$

As explained, noise in behavioral models is explicitly modeled in this work (i.e. the resistor in the behavioral model is noiseless, and represented by i_n). Noise current source i_n in Figure 3.14(b) therefore represents the noise generated by resistor R and current sources $i_{n_{diff}}$. It can be shown that the differential noise power produced by the identical current sources $i_{n_{diff}}$ is $\overline{i_{n_{diff}}^2}/2$ [88].

The single-ended current source i_n models the noise in the differential circuit correctly if

$$\overline{i_n^2} = \frac{4kT}{R_p} + \frac{\overline{i_{n_{diff}}^2}}{2}. \tag{3.14}$$

3.4.3 Circuit level modeling

At circuit level, an oscillator is represented by a circuit diagram, in which each element represents a physical component such as a physical transistor, physical resistor or physical capacitor, etcetera. Although each component is designed to implement one

Bipolar modeling	MOS modeling
Gummel-Poon	BSIM (level 1)
Mextram 503	MOS level 9 / BSIM3
Mextram 504	MOS level 11 / BSIM4

Table 3.1 Three modeling levels of bipolar and MOS transistors.

primary physical function, physical components also exhibit many unwanted properties and second order effects.

At circuit level the accuracy of FOMs derived with behavioral level modeling can be improved. To be able to do this, an equivalent circuit model has to be drawn of the circuit diagram. For example, a resistor in a circuit diagram can be replaced by a resistor model with capacitive losses to ground. The physical resistor is replaced by a model built with network elements, which approximates the physical resistor. Similarly, the transistor symbols in a circuit diagram have to be replaced by a transistors model, before calculations can be performed. In Table 3.1 a number of transistor models are listed. Only the models on the first row qualify for use in hand calculations although even these "simple" models are normally reduced to simplify hand calculations. Bipolar and MOS transistor "back-of-the-envelope"-models can be found in [87,89]. Mextram 504[14], MOS level 11[15] and BSIM4[16] are very advanced and complex transistor models and these should be used for circuit simulations when available. These models are optimized to capture, among other effects, the high-frequency and saturation effects of integrated transistors, whereas Gummel-Poon and BSIM (level 1) simply are not complex enough to predict measured behavior accurately. The values of the parameters in the transistor models are provided by the IC technology of choice.

3.5 Summary

High-frequency oscillator design is an analog design process. Inputs to this process are functional specifications and design resources. The functional specifications specify the performance an oscillator should achieve, whereas the design resources provide the means to meet these specifications. An analog design process, like the design of an oscillator, can be divided into three phases. In the first phase, "specification and conceptualization", the design space is explored and a topology is chosen. In

[14] http://www.semiconductors.philips.com

[15] http://www.semiconductors.philips.com

[16] http://www-device.eecs.berkeley.edu

the second phase, "optimization and implementation", the oscillator is dimensioned and a layout is prepared. The main activity in the final phase is "verification and documentation". Especially in the first phase, design heuristics are used by the analog designer to reduce the complexity of the design problem.

Three structured design methods are discussed. Trial and error, optimization tools and expert systems/synthesis environments. Although all reviewed structured methods aim to speed up design phase one and two, a practical, insightful and appealing method for topology selection, which has a major influence on the total design time, was not encountered in literature. Well-defined figures of merit provide quantitative information of the performance of one or more oscillator properties. They limit the oscillator design space, guiding the oscillator designer in his search for a cost-effective oscillator that meets the functional specification.

The concept of design FOMs and benchmark FOMs is proposed as a structured design method for high-frequency oscillators. A design FOM provides the designer with quantitative information on the performance of a oscillator property with respect to its specified value. If the realization of a design FOM is negative in dBs, the functional specification is not met (negative design margin), if it is 0 dB the specification is met exactly, and if it is positive the amount of positive design margin is indicated. Therefore all relevant design FOMs for an oscillator will always be zero or positive if the electrical design of the oscillator is ready.

Benchmark FOMs are useful to compare performance aspects of an oscillator design with the state-of the-art. This can be viewed as relative benchmarking. Alternatively the normalization function h in a benchmark FOM may be a theoretical performance limit. In this case the performance of an oscillator is benchmarked in an absolute sense to a theoretical bound. If the value of such a benchmark FOM is positive (the oscillator performance is greater than the theoretical limit), the designer knows the given specification is impossible to achieve, or the simulation or measured value is inconsistent.

System, behavioral and circuit level modeling are discussed in the final part of this chapter, as part of the modeling framework that will be used to explore the oscillator design space (in Chapter 5 and 6) and to define FOM examples (Chapter 7).