An Optimization-based Methodology for High-Level Design of Analog Systems

Soumya Pandit

An Optimization-based Methodology for High-Level Design of Analog Systems

Thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Ph.D.)

by
Soumya Pandit

School of Information Technology

Indian Institute of Technology

Kharagpur - 721 302, INDIA

March 2008

Dedicated to
my father, Sri. Sujit Kumar Pandit,
my mother, Smt. Mukul Pandit,
my wife, Smt. Srabanti Pandit,
my brothers and other family members, including in-laws
for their constant support, encouragement and love.

Abstract

The growing complexity of integrated systems being designed today, together with the increasing fragility of analog components brought about by shrinking geometries and reduced power consumption, pose severe challenges to traditional analog integrated circuit (IC) designers to produce satisfactory results in a short time. In order to improve the analog design quality and reduce the design time, the new analog design methodologies are hierarchically divided into several abstraction levels. High-level design of analog systems is an important step in an analog design automation process. This motivates us to develop methodologies which make several tasks of the analog high-level design process fast and accurate. This thesis presents optimization-based methodologies for the task of high-level performance model generation, optimal component-level topology generation and high-level specification translation.

This thesis first presents a non parametric regression-based methodology for the generation of high-level performance models for analog component blocks. The transistor sizes of the circuit-level implementations of the component blocks define the sample space. Performance data are generated through SPICE simulation. Least squares support vector machine (LS-SVM) is used as regression function. Optimal values of the model hyper parameters are determined through grid search technique and a GA-based technique. The constructed performance models are used within a GA-based topology sizing process. The entire methodology has been demonstrated with numerical examples.

This thesis then presents a top-down methodology for the generation of an optimal component-level topology for linear analog systems. The topologies are generated from a transfer function model of the system via state space matrix models. The topology exploration process is modeled as a state space matrix exploration process. Simulated annealing based optimization procedure determines an optimal state space model which is subsequently realized by appropriate analog component blocks to generate an optimal component-level topology. As a case study, the thesis presents a methodology for generation of an operational transconductance amplifier

(OTA)-capacitor (C) based topology for continuous-time $\Sigma\Delta$ modulator.

The thesis finally presents a methodology for the task of high-level specification translation. A meet-in-the-middle approach is followed for the construction of the feasible design space. Least squares support vector machine (LS-SVM) technique is used to identify an accurate geometry of the actual feasible design space. Genetic algorithm (GA) is used to explore the feasible design space. The effectiveness of the procedure is illustrated with numerical examples.

These methodologies form the core of an semi-automated tool for analog high-level design. The methodologies have been implemented under Matlab-Simulink environment. For demonstration of the methodologies, we choose two case studies – interface electronics for MEMS capacitive accelerometer sensor and continuous time $\Sigma\Delta$ modulator system. Optimal topologies for these two systems have been generated and specification parameters of the component blocks have been determined using the present methodologies. Finally they have been implemented at the transistor level and are simulated with SPICE. The SPICE simulation results satisfy the desired specifications of the system and matches closely with the predicted results. This validates the entire procedure.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The growing complexity of integrated systems being designed today, together with the increasing fragility of analog components brought about by shrinking geometries and reduced power consumption, pose severe challenges to traditional analog integrated circuit (IC) designers to produce satisfactory results in a short time. This situation has created a strong interest among the designers in developing new design methodologies and supporting computer aided design tools. The new IC design methodologies such as top-down constraint-driven methodology, platform-based design etc. are based on a hierarchy of abstraction levels – system design, architectural design, cell design, cell layout and system layout design [1, 2, 3]. The design task that is performed at the architecture level of abstraction is referred to as high-level design. This includes decomposition of the system into an architecture consisting of functional component blocks, e.g., amplifiers, filters, ADCs etc. required to realize the specified behavior [3].

An analog high-level design process is formally defined as the translation of analog system-level specifications into a proper architecture of component blocks, in which the specifications of all the component blocks are completely determined so that the overall system meets its desired specifications [4, 5]. The system-level specifications include functional description of the system and desired functional and performance specifications. Examples of component blocks are integrators, adders, mixers etc. The flow of a typical analog high-level design procedure is illustrated in Fig. 1.1 [4]. The analog high-level design process consists of three steps. The first step includes the task of selecting a suitable topology i.e., an interconnection of lower-level component blocks that is capable of realizing the desired behavior.

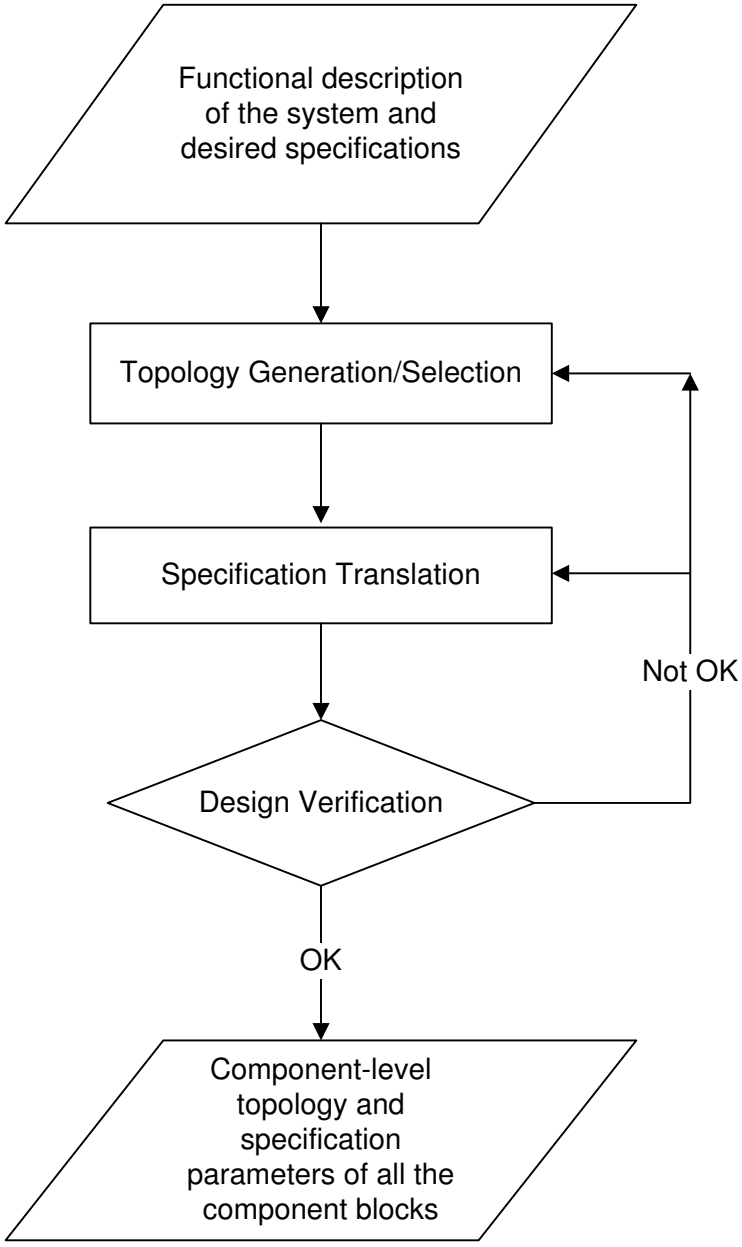


Figure 1.1: Flow of a typical analog high-level design procedure

This phase of the analog synthesis process is called topology selection/generation. At the architecture/high-level design abstraction level, the component blocks of a topology are generally represented by their behavioral models and so at this level the topologies are sometimes referred to as high-level topologies. In the second step, the specifications of the system under design are mapped into individual specifications for each of the component blocks within the selected system topology, so that the complete system meets its specifications, while possibly optimizing the design toward some application-specific design objectives (e.g., minimal power consumption). This process is referred to as specification translation. In the third step, the correctness of the first two steps is verified, generally through behavioral simulation. If the desired specifications of the system are not met, then one or both of the first two steps are repeated.

There are three different approaches for analog high-level design [4]. Of them, two are optimization-based approaches (simulation-based and equation-based) and one is a library-based approach. The optimization-based high-level design methodology is the subject matter of this thesis. The present research work concentrates on different but related aspects of an optimization-based analog high-level design process such as high-level performance estimation, generation of an optimal component-level topology of a system and high-level specification translation.

The rest of the chapter is organized as follows. Section 1 presents a comprehensive survey of literature on the above mentioned topics that led to the motivation behind the present work. Section 2 gives an overview of the work and highlights the specific contributions of the thesis. Finally, section 3 outlines the organization of the rest of the thesis.

1.1 Literature Survey and Motivation

In this section, we present a comprehensive survey of the literature related to the topics – high-level performance estimation, topology generation and specification translation. These provide us the motivation for the present work.

1.1.1 Analog Performance Modeling

An analog performance estimation (APE) tool for high-level synthesis of analog integrated circuits is described in [6, 7]. It takes the design parameters (transistor

sizes, biasing) of an analog circuit as inputs and determines its performance parameters along with anticipated sizes of all the circuit elements. The tool is structured as a hierarchical estimation engine containing performance models of analog circuits at various levels of abstraction like simple analog circuits (current mirrors, V-I converters etc.), operational amplifiers, and analog library cells (integrators, filters, amplifiers etc.). The estimates are propagated through all the levels using symbolic equations that relate them. The estimator is fast to evaluate but the accuracy of the estimated results with respect to real circuit-level simulation results is not good. This is because performance equations are based on simplified MOS models (SPICE level 1 equations). A power estimation model for ADC using empirical formulae is described in [8]. Although this is fast, the accuracy with respect to real simulation results under all conditions is off by orders of magnitude. The same reference [8] describes another approach for estimating power consumption of analog filters. Apart from using generic theoretical formulae, additional information such as the topology and the type of the filter are taken into consideration for improving the accuracy level. The technique for generation of posynomial equations based performance estimation models for analog circuits like opamps, multistage amplifiers, switch capacitor filters etc. is described in [9, 10]. An important advantage of such modeling approach is that the topology sizing process can be formulated as a geometric program, which is easy to solve through very fast techniques. However, there are several limitations with this technique. The derivation of performance equations is often a manual process, based on simple MOS equations. In addition, although many analog circuit characteristics can be cast in posynomial format, this is not true for all characteristics. For such characteristics, often an approximate representations are used. An automatic procedure for generation of posynomial models using fitting technique is described in [11, 12]. This technique overcomes several limitations of handcrafted posynomial modeling techniques. The models are built from data obtained through SPICE simulations. Therefore, full accuracy of SPICE simulations are achieved through such performance models. A limitation of the fitting technique is that a good model template needs to be selected before the construction process and the quality of the estimated results depends upon the chosen template. The model selection process is often a difficult task. A neural network based tool for automated power and area estimation is described in [13]. Circuit simulation results are used to train a neural network model, which is subsequently used as an estimator. Fairly recently, SVM has been used for modeling of performance parameters

for RF and analog circuits [14, 15, 16]. Analog performance models constructed with regression technique is generally fast to evaluate and accuracy with respect to circuit-level simulation results is good. However, a major limitation to this technique is that, the model construction time is often too high which increases the design overhead.

In this work, we have explored the possibility of developing a methodology for generating a high-level performance model with low construction time and good generalization ability using non parametric regression-based technique.

1.1.2 Generation/Selection of an Optimal System Topology

A fairly complete survey of the methodologies for generation/selection of an optimal system topology has been presented in [17]. There are three main classes of techniques for a high-level topology generation and selection process. They are: (i) selection before or after sizing, (ii) selection during sizing and (iii) top-down generation.

There are several approaches for the selection mechanism in the 'selection before or after sizing' technique. In [18], the topology selection tool selects from all topologies in the library only those that are able to satisfy the specifications of the component blocks as determined in the specification sheet and ranks them in order of preference. In [19], a figure-of-merit function is computed using behavioral models which estimates the trade-off between several performances. This function is used for guiding the topology selection process. In [20, 5], optimization techniques are used to optimize the performances of each topology present in a library. From them, the best one is selected. Similar technique has been used for selection of an optimal topology for $\Sigma\Delta$ modulator in [21, 22]. The 'selection during sizing' technique is based on the use of topology templates. For selection of an optimal opamp topology, this approach has been used in [23]. At the architectural level, this has been used in [24] for selection of an optimal topology for $\Sigma\Delta$ modulator system. A drawback of the design strategies based on selection before, during or after sizing is that they select all available topologies from a library, either entirely or as a template with a few binary options for component blocks or interconnections. On the other hand, strategies which create the topology offer a wider design range. The 'top-down generation' technique for topology generation has been followed in [7, 25, 26]. The topology generation methodology starts from a signal flow graph (SFG) descrip-

tion of the system topology. In [7], a branch-and-bound algorithm first generates alternative component-level topologies by mapping SFG components to library elements. For each resultant topology a genetic algorithm based technique is used for constraint transformation. In [25], a tabu search method and heuristic conversion rules are used to find different opamp based topologies for analog circuits. In [26], genetic algorithm has been used for simultaneous architectural and parametric optimization. For all these methodologies, the starting SFG description of the system is a relatively a low level description and is oriented to a specific system topology. These methodologies therefore fail to generate optimal component-level topologies directly from specifications. In [27], component-level topologies for analog filters have been generated from a transfer function model. This is relatively a higher level of description. However, the aspect of generation of an optimal topology has not been addressed.

In this work, we have explored the possibility of developing a methodology for generating an optimal component-level topology of a system directly from desired specifications and a high-level description of the system.

1.1.3 High-Level Specification Translation

In an analog high-level design methodology feasibility models are required for limiting the high-level design space exploration procedure to generate realizable values of the component block specifications. [28] presents a technique for construction of the feasibility model using binary search techniques - radial binary search and vertical binary search. [29] presents two methods for the calculation of the feasible performance values of analog circuits. The first method [30] computes the Pareto-optimal trade-offs of competing performances at full simulator accuracy. The Pareto front is a part of the boundary of the feasible performance region. The second method [31] computes linear polytopal approximations to the feasible performance region. This technique provides only an approximate representation of the feasible design space. Another method that identifies the entire range of feasible performance values using support vector machine principle has been presented in [32]. This technique considers only the circuit realizable space while constructing the feasible design space. Application system specific constraints and mutual influence between the component blocks have not been considered. An approximation to the feasible performance region by box constraints has been presented in [18]. A directed interval based search

space profiling technique and a genetic optimization-based constraint transformation technique is described in [33]. The constructed feasibility models are used within a design space exploration process to implement a specification translation process [29, 18, 34, 35, 33].

In this work, we have explored the possibility of developing a methodology for constructing the feasible design space by incorporating system constraints and mutual influence between the component-blocks. Further, we seek to identify the feasible design space accurately and develop an exploration procedure for high-level specification translation.

1.2 Overview and Contributions of the Thesis

This section first identifies the problems that have been addressed in the thesis and then gives a brief overview of the methodologies adopted to address them. Finally, the major contributions of the thesis are summarized.

1.2.1 Problem Definition

The emphasis of the thesis is on optimization-based methodologies for the different tasks related to analog high-level design. The specific problems that have been addressed in the thesis are as follows:

1. Development of a systematic methodology for construction of high-level analog performance models with good generalization ability and low construction time using non parametric regression-based approach.

2. Development of a methodology for top-down generation of an optimal component-level topology for linear analog systems starting from a high-level description of the system.

3. Development of a methodology for construction and accurate identification of a feasible design space and an exploration technique for high-level specification translation.

The following subsections give an overview of the approaches for solving these problems. The major contributions of the thesis are also highlighted.

1.2.2 Generation of High-Level Performance Models

The thesis presents a methodology for generation of high-level performance models for analog component blocks using statistical learning technique. A sample space is defined by applying a set of geometry constraints on the transistor sizes. A Halton sequence generator is used for extracting samples from the sample space. Performance data are generated through SPICE simulation. For training of the model, only those samples are considered which satisfy a set of functional and performance constraints. Least squares support vector machine (LS-SVM) is used as regression function. Optimal values for model hyper parameters are determined using two techniques - grid search technique and genetic algorithm (GA). Generalization ability of the constructed models is estimated using a hold out method and a 5-fold cross validation technique. Average relative error and correlation coefficients are calculated for measuring the quality of the constructed models.

The methodology is demonstrated with a set of experiments which are as follows:

1. Performance models corresponding to thermal noise, power consumption and output impedance of an operational transconductance amplifier are developed.

2. A comparison between the models constructed with the grid search-based training technique and the GA-based training technique w.r.t. generalization ability and training time is made. It is found that the training time is considerably less for GA-based training technique compared to the grid search-based training technique, with almost the same generalization ability.

3. The constructed performance estimation models are used in a genetic algorithm-based high-level topology sizing process. As an example, the interface electronics for a MEMS capacitive accelerometer has been chosen. The predicted results are compared with SPICE simulation results. The two sets of results match closely.

1.2.3 Top-Down Methodology for Generation of an Optimal Topology for Linear Analog Systems

The thesis then presents a structural synthesis approach for top-down generation of an optimal component-level topology for linear analog systems. The crux of the methodology is that topologies are generated directly from a transfer function model

of a system. The given transfer function model is converted to a state space matrix model. This acts as a basis for topology generation. The topology exploration process is modeled as a state space matrix exploration process. Similarity transformation matrix is used for generation of a new state space model from a given one. These new topologies have identical behavioral properties but they differ in performances. Simulated annealing based optimization technique is used to determine an optimal state space model such that the resultant topology is optimized for a set of performance parameters. The optimized state space model is realized by appropriate analog component blocks to generate an optimal component-level topology. This is then behaviorally simulated to check whether all the specifications are satisfied by the generated topology even in the presence of circuit-level non-idealities. If the test fails, the complete process is repeated and new topologies are generated.

As a case study, the thesis presents a methodology for generation of an operational transconductance amplifier (OTA)-capacitor (C) based topology for continuous-time $\Sigma\Delta$ modulator. The loop filter transfer functions are taken as inputs. The chosen performance metrics are system hardware complexity, sensitivity under parameter variation and relative power consumption. A 3^{rd} order and a 4^{th} order modulator have been chosen as examples for experimentation. The experiments that are carried out for illustrating the methodology described above for each of the examples are as follows:

1. The behavioral equivalence between the newly generated topologies under non-ideal conditions is validated through behavioral simulation. The dynamic ranges are determined. These are nearly equal.

2. The generated topology satisfies the desired dynamic range under non-ideal conditions and overloading does not take place.

3. The performances of the generated topology are compared with that of two standard topologies. Monte Carlo analysis is performed for comparing the sensitivity performances. The yield and performance deviation are computed. It is found that the generated topology is more tolerant to design parameter variations not only in terms of yield but also performance deviations.

It is concluded from the experimental results that the generated topology is better in performances compared to commonly used topologies and satisfy the desired specifications under circuit-level non-idealities.

1.2.4 High-Level Specification Translation

Finally, the thesis presents a methodology for high-level specification translation. A meet-in-the-middle approach is followed for the construction of the feasible design space. This is constructed as the intersection of an application bounded specification space and a circuit realizable specification space. The former is constructed through a top-down procedure using analytical techniques and the latter via a bottom-up procedure through actual circuit simulation. The intersection between these two spaces defines the feasible design space. The tuples that lie within the intersection space are considered to be feasible tuples. Least squares support vector machine (LS-SVM) technique is used to identify an accurate geometry of the actual feasible design space. Genetic algorithm (GA) is used to explore the feasible design space. The final solution point is kept away from the feasible design space boundary, in order to increase the tolerance of component-specifications.

Two case studies, an interface electronics for MEMS capacitive accelerometer sensor and a continuous time $\Sigma\Delta$ modulator have been presented to demonstrate the effectiveness of the procedure. The experiments that are carried out for each case study are as follows:

1. LS-SVM feasibility models are constructed for all the component blocks. A set of performance metrics, viz., sensitivity, specificity and accuracy are computed. These values are found to be close to their ideal values.

2. With the determined specifications of the component blocks, the target systems are implemented at the transistor level and are simulated with SPICE. The SPICE simulation results satisfy the functional specifications of the system, validating the overall procedure.

1.2.5 Contributions

This work has three major contributions as listed below.

1. A methodology is developed for generation of good high-level performance estimation models for analog component blocks using least squares support vector machine (LS-SVM). The models have high accuracy and good generalization ability. The model construction time is low.

2. A methodology is developed for generation of an optimal component-level topology for linear analog systems starting from a transfer function model of the system. The generated topology is ensured to perform satisfactorily under circuit-level non-ideal conditions. Through this methodology, the designer is able to specify the design goal and desired specifications at a higher level of abstraction in terms of transfer functions, while the design tool is able to generate an optimal component-level topology directly from the transfer functions in a highly automated manner.

3. A methodology is developed for high-level specification translation. Through this methodology, it is possible to obtain a set of practically correct circuit level specifications of the component blocks of a system through a fast exploration process in a single pass.

1.3 Organization of the rest of the Thesis

This section presents an outline of the organization of the rest of the thesis.

Chapter 2 : It gives an overview of optimization-based generic methodology for topology sizing and specification translation task. It also discusses in brief the current state-of-the art techniques for construction of high-level models viz., performance models, behavioral models and feasibility models and various optimization methods. The basic principle of the various methods for generation/selection of an optimal system topology are also discussed.

Chapter 3 : This chapter describes in detail the various steps for construction of high-level performance models using least squares SVM technique. It also includes a GA based topology sizing procedure, where the constructed performance models are used. Numerical results have been provided to demonstrate the effectiveness of the methodology.

Chapter 4 : This chapter describes a methodology for top-down generation of an optimal component-level topology for linear analog systems. As a case study, a continuous-time $\Sigma\Delta$ modulator system has been chosen. Numerical results have been provided to illustrate the effectiveness of the methodology.

Chapter 5 : This chapter describes a methodology for high-level specification translation. Feasible design space has been identified accurately using least squares SVM technique. GA-based optimization technique has been used for design space exploration. Numerical results have been provided to demonstrate the effectiveness of the methodology.

Chapter 6 : The major contributions of this thesis are summarized in this chapter and some pointers to future research have been provided.

Chapter 2

Optimization-based Analog High-Level Design Methodology

A general procedure for an analog high-level design process has been shown in Fig. 1.1. As mentioned in the previous chapter, there are three different methodologies for an analog high-level design process. Of them, two are optimization-based methods (one with simulations in the loop, the other with analytical equations) and one is a library-based method. The optimization-based methodology which is the subject matter of this thesis is discussed in this chapter. At the heart of an optimization-based methodology lies several classes of high-level models such as performance models, behavioral models etc. and an optimization procedure. This chapter presents an overview of the basic principle of the state-of-the art techniques for constructing these models and the optimization procedures. It also includes a brief survey of the existing techniques for generation/selection of an optimal component-level topology of a system. This chapter therefore, provides the necessary background required for understanding the contributions of the carried research work.

The chapter is organized as follows. Section 1 presents a generic methodology of an optimization-based high-level design procedure. It discusses the simulation-based approach and the equation-based approach for high-level design. Section 2 discusses the state-of-the art techniques for construction of the different high-level models. Section 3 discusses the various methods of an optimization procedure. Section 4 presents an overview of the various techniques for generation/selection of system topologies. Finally section 5 presents a summary of the chapter content.

2.1 Generic Methodology

In an optimization-based high-level design methodology, the design problem is translated into a function minimization problem that is solved through numerical optimization techniques. These techniques implicitly solve the degrees of freedom of the design problem while optimizing the performances of the circuit/system under the given specification constraints. The optimization-based methodology for the task of specification translation/topology sizing is schematically illustrated in Fig. 2.1. The design variables are the specification parameters of the component blocks used in the topology, e.g. gain, bandwidth etc. of an amplifier. The entire procedure is an iterative process, where design variables are updated at each iteration, until an equilibrium point is reached. The degree of compliance of the design performances with the optimization goals at each iteration is quantified through a cost function. The two important modules for this type of design methodology are a performance estimation module and an optimization engine. The implementation of the design methodology is based upon the flow of information between these two modules. The performance estimation module provides a way to evaluate the optimality of the design with regard to the intended requirements. On the other hand, the optimization engine deals with the cost function and explores the available design space to minimize such function. The cost function being minimized during the optimization process contains two sorts of terms: (1) terms related to the difference between the desired specifications of the system and the evaluated system performance values (for a particular set of specification parameters of the component blocks) and (2) terms related to the general objectives that have to be minimized at the same time, such as power or area.

Depending upon the type of performance evaluation, two different approaches – simulation-based approach and equation-based approach are distinguished. The basic principle of these two approaches are discussed below.

2.1.1 Simulation-based Approach

In this approach, the performance evaluation process inside the optimization loop of Fig. 2.1 is implemented by means of behavioral simulation. Parameterized behavioral models are developed for all component blocks of the system topology. The specification parameters of the component blocks serve as the model parameters.

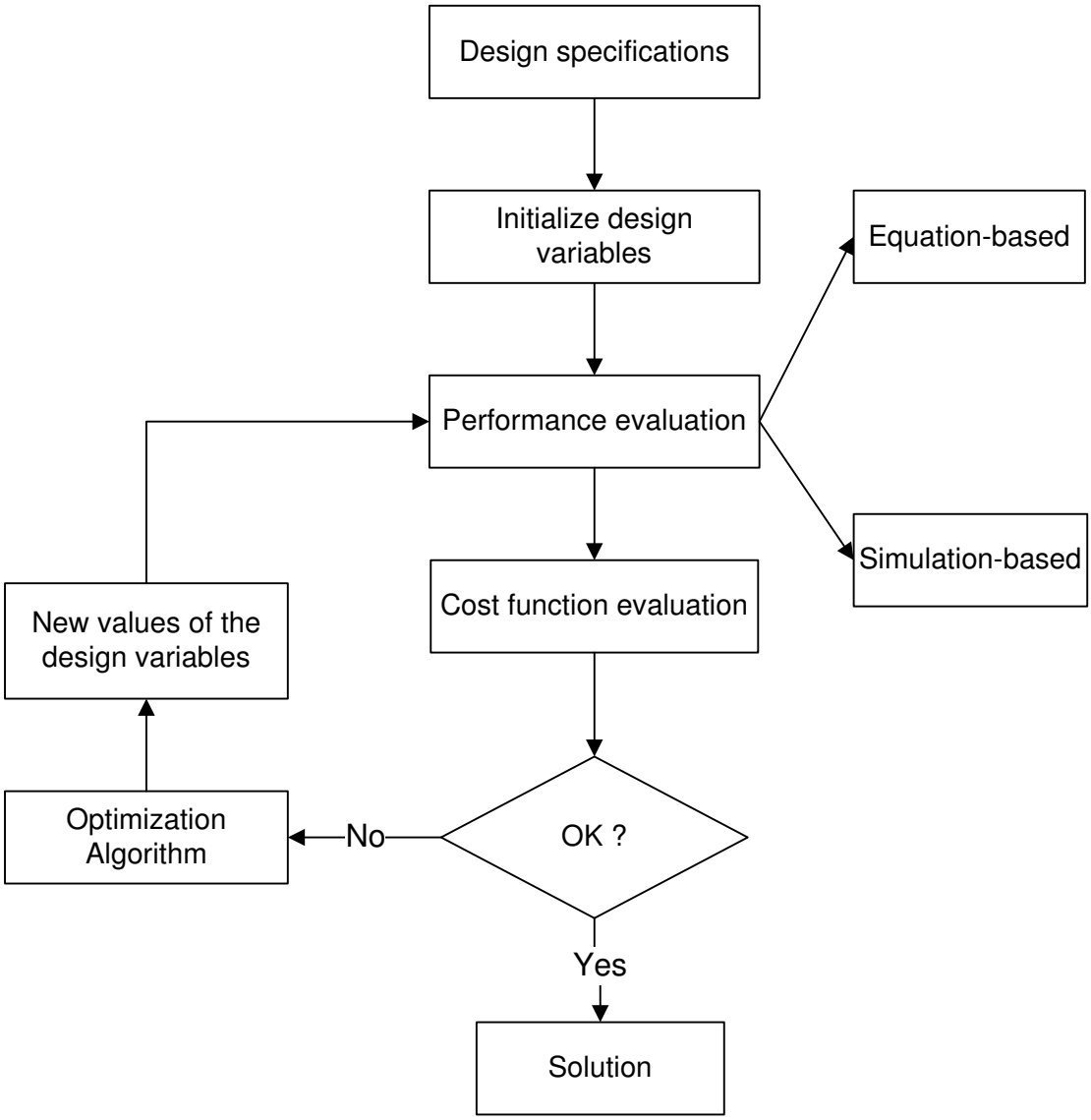


Figure 2.1: Optimization-based design flow

The system topology is described as an interconnection of these behavioral models. Amongst the model parameters, those which will be used as design variables along with the respective bounds are identified by the users. In addition, the user needs to provide the desired specifications of the system and a simulation plan for each of these specifications. Such plan includes the test set up (input sources, loads, feedbacks etc.), the input signals and simulation commands and the required data processing in the simulation results to obtain the desired system performances.

A serious problem of this approach is the required simulation time of one performance evaluation. Since global optimization techniques typically take several thousands of iterations, the evaluation time of one complete run will be unacceptable, if one performance evaluation takes more than a few seconds. On the other hand, an important advantage of this approach is that the process is flexible and user can program a new high-level design problem in a minimum amount of time (if all the component block behavioral models are available). The development of behavioral models for the component blocks is a one-time process and these can be reused whenever such component blocks are part of an architecture.

A simulation-based technique for high-level design and optimization of analog RF receiver front-ends is described in [20]. The design methodology works to evaluate the performance of an RF receiver topology and automatically translates high-level system specifications into a set of specifications for each building block in the topology such that the overall power and/or area consumption of the receiver is minimized. Similar work for $\Sigma\Delta$ modulator design is reported in [21, 22].

2.1.2 Equation-based Approach

This approach uses analytical equations for performance evaluation. These equations directly related the specification parameters of the component blocks with the desired specifications of the system and are generally derived through designer's knowledge or through symbolic analysis approach[3]. An important advantage of this approach is that the evaluation of a set of equations is much faster compared to behavioral simulation. Therefore, the execution time of a complete optimization process is generally small. The disadvantage is the much larger setup time. The user needs to derive all the design equations, which is difficult and time consuming as well as an error-prone task. The accuracy of the performance equations compared to circuit-level simulation results is often not good. In addition, several performance

characteristics cannot be suitably captured by analytical equations. Furthermore, the design equations are often very specific for the system topology and cannot be reused for other topologies.

Symbolic equations based high-level design procedure has been reported in [25, 7]. This technique has also been applied to the high-level design of $\Sigma\Delta$ modulator in the SD-OPT tool [36].

2.2 High-Level Model Generation

The various types of high-level models that are encountered in an analog high-level design process are behavioral models, performance models and feasibility models. In this section, we discuss the basic principle of the various techniques for constructing these models.

2.2.1 Behavioral Model Generation

Let us consider a system S transforming an input signal U into an output signal Y. Suppose the system is governed by a vector of design parameters $\bar{X}$ that influence its behavior. Then

$$Y = \mathcal{B}\left(U, \bar{X}\right) \tag{2.1}$$

Here $\mathcal{B}$ is called as the parameterized behavioral model of the system S. The mathematical modeling of the system's input-output behavior is called behavioral modeling.

Good behavioral models are essential components in a behavioral simulation-based high-level design process. The models need to be good in two senses [37]. First, they must accurately represent all practical circuit behavior. For example, a behavioral model for a voltage amplifier must capture all of the relevant behavior that characterizes an amplifier's transistor-level implementations. Use of inaccurate models lead to wrong high-level design. Second, the models need to be as simple as possible. Complex and overly detailed models compromise efficiency of designs as they result in tedious computations and lengthy simulations. These cause problems in high-level topology optimization and specification translation process, where several component blocks need to be evaluated simultaneously. Therefore, managing the accuracy and simplicity of the behavioral models is the greatest challenge in

behavioral model generation techniques. Systematic generation of good behavioral models is considered as one of the largest problems in an analog high-level design automation process. The commonly used techniques are roughly divided into analytical techniques, fitting or regression techniques, symbolic analysis techniques and model order reduction methods. We discuss the basic principle of each of these techniques in brief below.

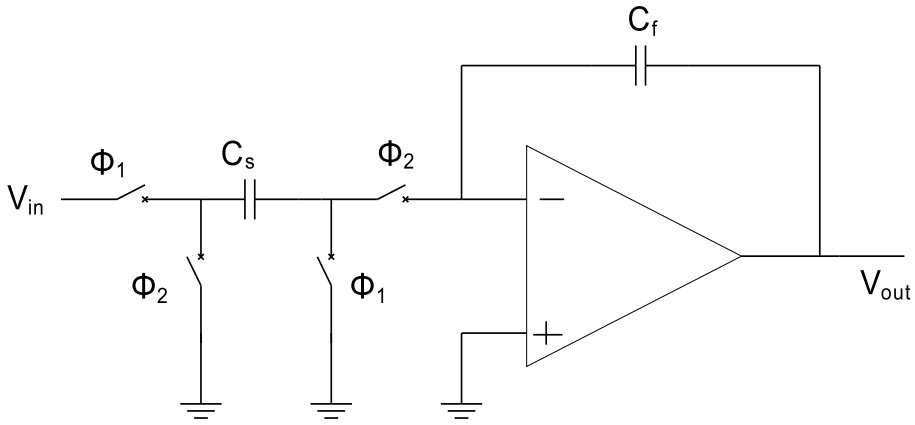
2.2.1.1 Analytical Techniques

The complete behavior of a component block is split up into two parts– fundamental/ideal behavior and nonidealities. For any analog component block, the ideal behavior is generally a simple mathematical operation such as scaling, integration, multiplication etc. The nonidealities are then modeled in terms of the effects they introduce, e.g., distortion rather than in terms of the causes, e.g., transistor sizes or particular topologies. The circuit-level implementation details are not considered. The constructed models are generally simple. Hence this method is suitable for a high-level design and optimization procedure. However, the models are not always perfect, since in many cases they are based on a number of hypothesis which are typically applicable to a particular system. The basic trade-off between accuracy and simplicity of the models is optimized in terms of model simplicity. Simulation frameworks like Simulink, AMS Designer/Verilog-AMS etc. are suitable for implementing the models. In chapter 5 of this thesis, behavioral models constructed through analytical techniques are used for high-level specification translation process.

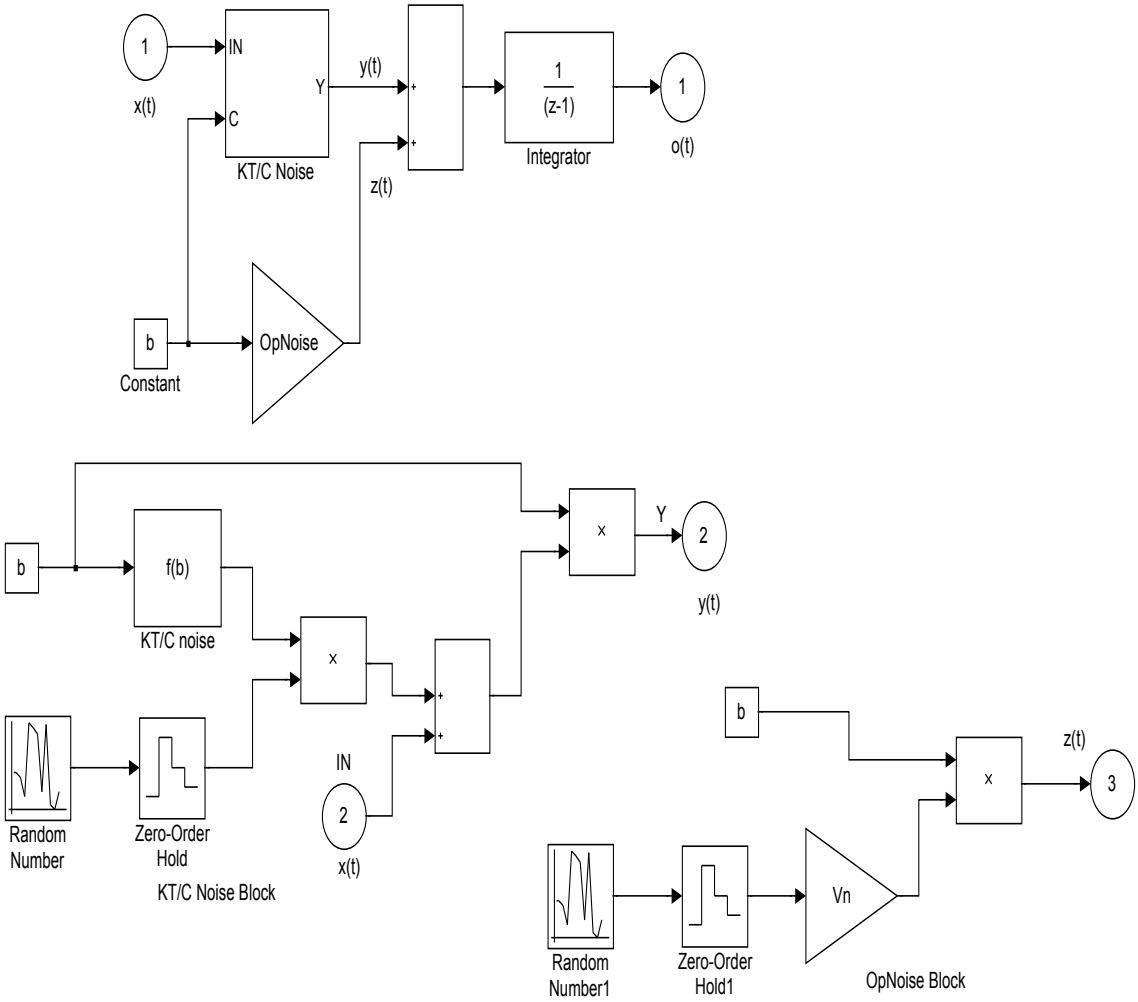
The procedure is illustrated with an example for modeling the transfer function and noise properties of a switch-capacitor (SC) integrator, as shown in Fig. 2.2(a). The z-domain transfer function of the integrator is given by

$$H(z) = \frac{C_s}{C_f} \frac{z^{-1}}{1 - z^{-1}} \tag{2.2}$$

$C_s/C_f = b$ represents the coefficient of the integrator. The most important noise sources affecting the operation of an SC integrator are the thermal noise due to the sampling switches and the intrinsic noise of the operational amplifier. The switch thermal noise voltage e_T (usually called the KT/C noise) is superimposed on the



(a) Single-ended SC integrator.



(b) Model of a noisy integrator.

Figure 2.2: Simulink based behavioral model

input voltage $x(t)$ leading to

$$\begin{aligned} y(t) &= [x(t) + e_T(t)]\, b \\ &= \left[x(t) + \sqrt{\frac{kT}{bC_f}} RN(t)\right] \end{aligned} \tag{2.3}$$

where $RN(t)$ denotes a Gaussian random process with unity standard deviation. The input referred thermal noise of the operational amplifier is modeled as

$$z(t) = bV_n RN(t) \tag{2.4}$$

where V_n represents the total *rms* noise voltage of the operational amplifier referred to the integrator input. The complete Simulink implementation of the integrator behavioral model including transfer function, KT/C noise and opamp noise is shown in Fig. 2.2(b).

Several works are available in literature which adopt this technique for behavioral model generation. Behavioral modeling of switched-capacitor $\Sigma-\Delta$ modulators following this technique using Simulink platform is presented in [22, 38, 39]. Behavioral modeling and simulation of pipelined ADC and PLL following this technique is discussed in [40] and [41] respectively.

2.2.1.2 Fitting or Regression Methods

In fitting or regression methods a parameterized model, e.g., a rational transfer function, a general set of equations is first proposed by the model developers and the values of the unknown parameters are then determined so as to best approximate the known circuit behavior. These methods are generic as they consider the block as a black-box and consider only the simulatable input-output behavior of the block. A drawback of these methods is that a good model template needs to be selected before the construction process and the quality of the estimated results depends heavily upon the chosen template. The selection process is often a difficult task without knowing the underlying circuit-level implementation details. Another possible black-box approach is the use of artificial neural network that is being trained with SPICE simulation results of the real circuit until the response of the network matches closely enough the response of the real circuit. The choice of an appropriate neural network structure is also not an easy task. In chapter 3 of this thesis, this method has been

used for construction of high-level performance estimation models.

A fitting approach for generation of posynomial models for analog circuits has been described in [11]. Neural network based approach has been used in [13, 42].

2.2.1.3 Symbolic Model Generation Methods

Symbolic analysis of an analog circuit is a formal technique to calculate the behavior or characteristic of a circuit with the independent variables (time or frequency), the dependent variables (voltages and currents) and (some or all of) of the circuit elements represented by symbols. This technique is thus complimentary to numerical analysis. Symbolic analysis method tries to generate a behavioral model starting from a circuit netlist. Pure symbolic analysis techniques include three different approaches [43]: determinant-based method, signal flow graph method and tree enumeration method. Determinant-based methods solve the set of linear equations implied by the symbolic analysis procedure [44]. Signal flow graph methods represent a set of linear equations as a weighted graph and use Mason's rule for solving the equation set. Tree enumeration method also describe a network as a graph. Whatever be the approaches followed, the expressions generated can then further be postprocessed in symbolic format. The main limitation inherent to a symbolic analysis method is the large computing time and/or memory storage, which increases very rapidly with the size of the circuit.

Symbolic analysis method has been widely used for automated analog circuit sizing. This approach has been explicitly adopted in the OPTIMAN [45], OPASYN [46], AMGIE [18].

2.2.1.4 Model Order Reduction Methods

The model order reduction methods are mathematical techniques that take a detailed description of a component block, e.g., a SPICE-level circuit netlist and generate, via an automated computational procedure, a much smaller behavioral model. The behavioral model, fundamentally a small system of equations, is usually translated into Matlab/Simulink form for use at the system level. Algorithmic approach for such model generation tackles the problem as the transformation of a large set of mathematical equations to a much smaller one. These reduced order models simulate much more efficiently, while approximating the response of a real circuit. There are two commonly used techniques for model order reduction - asymptotic waveform

evaluation (AWE) technique and Krylov-subspace technique. AWE technique uses explicit moment matching technique for model order reduction. On the other hand, Krylov-subspace technique uses projection matrices for model order reduction. An overview of these techniques is provided in [47, 48].

Although this method has not been used in our work, we present it here to make the discussion complete.

2.2.2 Performance Estimation Model Generation

A performance estimation model is a function that returns an estimated value for the performance of a component block, when given some design parameters of the block as input. Mathematically this is expressed as

$$\bar{\rho} = \mathcal{P}(\bar{X}) \tag{2.5}$$

where $\bar{\rho}$ is a vector of all performance parameters, e.g., bandwidth, slew rate for a component block, $\bar{X}$ is a vector of all design parameters and $\mathcal{P}$ is the performance model.

For the construction of a performance estimation model, there are two possible approaches [8]: a bottom-up approach and a top-down approach. Here we will discuss the basic principle, advantages and shortcomings of both the approaches.

2.2.2.1 Bottom-Up Approach

In the bottom-up approach, a certain circuit-level topology of a component block is selected and from this exactly known schematic, the performance equations are derived. These equations are derived either through symbolic analysis [43] or through regression method [47], discussed above. The advantage of this method is that the constructed estimators are exact and accurate with respect to real designs. The models are 'correct-by-construction'. The disadvantage is that circuit-level details are required for the construction process, which are generally not known precisely during the high-level design process. Moreover, as these methods do not rely on underlying operating principles, extrapolations of the models have no guaranteed accuracy. The bottom-up approach for performance model generation includes regression techniques and symbolic analysis techniques, the basic principle of which have been discussed already.

A fairly complete survey of the various works employing bottom-up approaches for performance model generation is provided in [47]. A fitting approach for generation of posynomial performance equations is described in [11]. Artificial neural network based approach for performance model generation is described in [13, 42]. Support vector machine based performance model generation technique is described in [15]. In chapter 3 of the dissertation, this approach has been followed for constructing the high-level performance models.

2.2.2.2 Top-Down Approach

In the top-down approach, a fundamental relation is derived between the performance parameters of a component block and the input high-level design parameters. The circuit-level topology of the component block is not considered. This is left as open. The result is a set of simple equations that is suitable for implementations in a fast architecture exploration procedure. Therefore, this approach is useful for real system-level design, where nothing is known about the circuit-level implementation details. A drawback of this approach is that good accuracy of the estimators is often difficult to achieve because of the typical nature of analog design where even one transistor can have more or less a great impact of the performances of a block. Another drawback of this approach is that a good knowledge of each component block of the system topology is required in order to make right simplifications. This is not an easy task.

A top-down approach for estimating power of a high-speed CMOS ADC is described in [49]. This approach has been used for estimating performances of analog filters in [25]. In [24], the top-down approach has been followed for estimating performances of $\Sigma\Delta$ modulators. In chapter 4 of the dissertation, this approach has been followed for constructing the performance models.

2.2.3 Feasibility Model Generation

A high-level specification translation procedure often produces overambitious specifications for the component blocks of the system, if the performance capabilities of the underlying analog circuit implementations are not taken into account during the translation process. Feasibility models are needed that limit the specification translation process to determine feasible specifications for the component blocks. The task of identifying the feasible performance region of a circuit is referred to as

performance space exploration (PSE) [29]. A PSE process may compute the whole region of feasible performance values or a set of optimal trade offs between competing performance targets, referred to as Pareto optimal front. The Pareto optimal front is a part of the boundary of the feasible performance region.

An approach for identifying the entire range of feasible performance values using support vector machines is presented in [32]. Using this, a feasibility function is developed, whose output takes only two values – 0 or 1 depending on whether the specifications are realizable at the circuit level or not. An approximation to the feasible performance region by lower and upper bounds of the individual performances is presented in [18]. In [31], feasible performance region is identified using a polytopal approximation technique, which is based on linearized models of circuit performances and structural circuit constraints. A geometric approach for identifying feasible design space using line search techniques is described in [28]. These techniques identify the boundary points of the feasible design space and then these are used to compute feasibility macromodels using radial basis functions. A pareto front computation method using statistical optimization techniques is described in [50]. A pareto front computation technique using normal boundary intersection method is described in [30].

2.3 Optimization Methods

In a parametric optimization procedure, the topology of the circuit/system and the component blocks are fixed. The nominal design problem consists of assigning values to a set of design variables so that the circuit/system performances are optimized, under the constraints that certain specifications are met. The design problem is formulated in terms of mathematical programming problems. There are two types of optimization methods for solving such problems – deterministic methods and stochastic methods. In this section, we briefly describe the working principle of both the methods.

2.3.1 Deterministic Methods

In this approach, the updation of the design variables requires information about the cost function and its derivative. Only changes of design variables that make the cost function to decrease are allowed. Commonly used deterministic methods

are simplex methods, gradient-based methods etc. A limitation of this approach is that the optimization process may be quickly trapped in a local optimum of the cost function. Another problem is the rapid increase of the execution time with the increase of the number of design variables and design space. These techniques are used primarily for the fine tuning of suboptimal sizings.

2.3.2 Stochastic Methods

In this methods, the design variables are varied randomly. The derivatives of the cost function is not required. Greedy stochastic algorithms only accept a new set of variables if there is some improvement. The main advantage of the stochastic methods over the deterministic ones is the capability to escape from local optimum and hence have a higher probability to determine a global optimum. Simulated annealing is an widely used stochastic method. In this method, starting from some point in the design space, a new set of variables is derived by selecting statistically a new point in the neighborhood of the old one or by applying a set of a local optimizer. In this method, the global optimum is theoretically reached after an infinite number of iterations. In an evolution-based stochastic method, e.g., genetic algorithm, a population of individuals is created where the design variables are collected in its genome. Each individual is assigned a fitness value corresponding to the cost function which can be used for ranking and selection. During the optimization process, new generations are built up by selection, mutation and recombination or crossover operators. Similar to simulated annealing algorithm, the global optimum is reached only after an infinite number of generations. The stochastic algorithms are thus computationally expensive.

Some of the key analog CAD tools using SA as the optimization tool are OPTIMAN [45], ASTRX/OBLX [51], ORCA [20], SD-Opt [36] and so on. On the other hand, some of the key analog CAD tools using genetic algorithm are ANTIGONE [26], Watson [52] etc.

2.4 Topology Generation/Selection Methods

The task of generation/selection of an optimal component-level topology of a system is an important step in an analog high-level design process. The distinction between the task of topology selection and generation is important. Topology selection is

the task of selecting the most appropriate topology that can best meet the given specifications out of a set of already known alternative topologies [3]. Topology generation is the task of generating a new component-level topology from a functional description of the system. A component-level topology of a system is defined in terms of component blocks like adder, integrator, multiplier etc. In the high-level design process, the component blocks are often represented by their high-level models. Thus a component-level topology of a system is sometimes referred to as a high-level topology during the high-level design process. There are several methods for generation/selection of an optimal component-level topology of a system. In the following subsections we give a brief overview of the basic principle of these methods. This is based upon the survey paper by Martens and Gielen [17].

2.4.1 Selection before or after sizing

There are two approaches for this method. In one approach, the topology is first selected by a designer based upon his/her experience or a knowledge-assistant tool. The topology parameters are then determined such that system performances are optimized. In the second approach, multiple topologies are sized such that the system performances are optimized and afterwards the best solution is selected. The performances of the topologies are estimated by evaluating performance estimation models. A drawback of this method is that only a limited set of topologies is available in a library and the topology selection process is limited to library elements only. In chapter 3 of this thesis, a genetic algorithm based technique is described for the purpose of topology sizing and optimization.

This methodology has been used for selection of an optimal topology for systems like PLL in [53], RF systems in [5, 20], $\Sigma\Delta$ modulators in [36].

2.4.2 Selection during sizing

In this method, the task of selection and sizing is performed simultaneously. Rather than selecting a specific topology, a template of the topology is chosen explicitly or implicitly. This template defines a topology in terms of various component blocks for which different alternative implementations exist. All implementation choices available in a library fit into the template. During the sizing process, a parameter chosen by the optimizer is translated into a topological choice. The task of optimal topology generation is formulated as a mixed integer nonlinear programming

problem which is solved using a constrained optimization method combined with a combinatorial optimizer like a branch and bound algorithm. A key issue of this technique is thus the definition of the template. This methodology has the limitation that the selected topology is limited to those encompassed within the template.

This methodology has been used for selection of an optimal topology for opamps in [23] and $\Sigma\Delta$ modulators in [24, 39].

2.4.3 Top-Down Generation

The previous two methods select all available topologies from a library, either entirely or as a template with a few binary options for different component blocks or interconnections. With these, new topologies cannot be generated. The top-down methodology which will be discussed in this subsection offers a wider design range and the possibility to generate new topologies. The top-down methodology starts from a functional description of the system. A functional description of a system is a representation of the functionality required from the system, sometimes annotated with information to guide the topology generation process. Hardware description languages like VHDL-AMS are generally used for this purpose. This description is first converted to some internal representation like signal flow graph [25, 7]. This representation is then mapped onto a connection of component blocks to form one or more component-level topology of the system. The mapping process is either implemented through heuristic conversion rules or through optimization process. The latter one involves the determination of the topology parameters such that the system performances are optimized. In chapter 4 of the dissertation, a top-down methodology has developed for generation of an optimal component-level topology for linear analog systems starting from a transfer function description of the system.

This methodology has been used for generation of opamp based topologies for linear analog systems in [25], general analog systems in [7], ADC in [54]. In [26], a genetic algorithm based technique has been described for simultaneous topological and parametric optimization process.

2.5 Summary

An optimization-based methodology for an analog high-level design process has been discussed in detail. The various techniques for constructing the behavioral models

of a system have been discussed. Behavioral models constructed using analytical techniques have been used in chapter 5 for evaluating the functional specifications of a system. The top-down approach as well as the bottom-up approach of constructing high-level performance models have been discussed. The former approach is followed in chapter 3 of the dissertation and the latter approach is followed in chapter 4 of the dissertation. The techniques for constructing the feasibility models have also been discussed. The deterministic as well as the stochastic methods of implementing an optimization procedure have also been discussed. The stochastic method has been followed in the present work. The various methods for generation/selection of an optimal component-level topology of a system have been described in brief. The top-down methodology has been considered in chapter 4 of the dissertation. This chapter therefore provides the required background of the present research work.

Chapter 3

Generation of High-Level Performance Estimation Models

In section 2.2.2 of the dissertation, we introduced the various approaches for construction of analog performance estimation models. In this chapter, we discuss a methodology for generation of high-level performance estimation models for analog component blocks following the bottom-up approach. The models are generated as functions of specification parameters of the component blocks. Non parametric regression technique using least squares support vector machine (LS-SVM) is used for model generation. The LS-SVM models are trained with data generated through SPICE simulation. Trained SVMs are subsequently used as performance models.

The chapter is organized as follows. Section 1 defines the high-level performance estimation models and their essential characteristics. In section 2, SVM based method to generate performance models is described in detail. Section 3 presents a comparison between the present methodology for high-level performance model generation and other existing methodologies. A topology sizing process using SVM models is described in section 4. Experimental results are discussed in section 5. Finally conclusion is drawn in section 6.

3.1 High-Level Performance Estimation Models

A high-level performance estimation model is a function that estimates the performance of a component block when given some high-level design parameters of the block as input [8]. The important requirements for a good high-level perfor-

mance model are : (i) The input parameters must be specification parameters of the component block, i.e.,

$$\bar{\rho} = \mathcal{P}(\bar{X}) \tag{3.1}$$

where $\mathcal{P}$ is the high-level performance estimation model, $\bar{\rho}$ is the set of estimated performances of a component block and $\bar{X}$ is a set of specification parameters for the component block. (ii) The model needs to be low dimensional. Only those specification parameters are to be considered as inputs which have dominant contributions on a performance parameter to be estimated. The task of construction of an accurate high-dimensional performance model is a difficult task [47]. In addition, a topology sizing process involving high-dimensional performance models often becomes difficult because of the wide design space to be explored. (iii) The predicted results need to be accurate. Lower model prediction error reduces the number of iterations of the sizing process. Model accuracy is measured as the deviation of the model prediction from the true function value. The function value in this case is the performance parameter obtained from transistor level simulation. (iv) The evaluation time must be short. This is measured by the CPU time required to evaluate a model. Stochastic global optimization techniques often require several thousands of iterations before convergence. The execution time of a topology sizing procedure becomes unacceptable, when one performance evaluation takes more than few seconds. (v) The time required to construct an accurate model must be small, so that the design overhead does not become high. This is relatively harder to quantify. This process involves both applying design knowledge to setup testbench circuit and design variable selection and computational time needed to use an algorithm to train a model. As a rough estimate, the construction cost can be measured as

$$T_{\text{construction}} = T_{\text{data generation}} + T_{\text{training}} \tag{3.2}$$

where the terms are self explanatory. There exists trade off between these requirements, as a model with lower prediction error generally takes more time for construction and evaluation.

3.2 Regression-based Model Generation

In this section, we describe the various steps of the construction methodology in detail one by one.

3.2.1 Sample Space and Design of Experiments

While choosing the set of inputs, only those specification parameters forming a set $\bar{X}' \subseteq \bar{X}$ which have dominant contributions to specific performance parameters $\bar{\rho} = \{\rho_1, \rho_2, ..., \rho_n\}$ are considered as inputs. This choice of inputs relies on designer's knowledge depending upon the application system and the topology considered. The dominant specification parameters are referred to as the high-level design parameters. For ease of notation, the tick indicating the reduction is omitted in the rest of this chapter. Both the inputs and output of $\mathcal{P}$ are taken to be functions of a set of geometry parameters $\bar{\alpha}$ (transistor sizes) of a component block, expressed as

$$\bar{X} = \mathcal{R}(\bar{\alpha}) \tag{3.3}$$

$$\bar{\rho} = \mathcal{Q}(\bar{\alpha}) \tag{3.4}$$

$\mathcal{R}$ and $\mathcal{Q}$ represents the mapping of the geometry parameters to electrical parameters. The multidimensional space spanned by the elements of the set $\bar{\alpha}$ is defined as circuit-level design space $\mathcal{D}_\alpha$.

A set of geometry constraints are applied on the transistor sizes to enclose a region within $\mathcal{D}_\alpha$, from which samples are extracted for training data generation. These geometry constraints include equality constraints as well as inequality constraints. The equality constraints, expressed as algebraic equations directly correlate the transistor sizes. For example, for matching purpose, the sizes of a differential pair transistors are equal. The equality constraints eliminate elements of the set $\bar{\alpha}$ and therefore reduce the dimension of the circuit-level design space $\mathcal{D}_\alpha$. The inequality constraints exclude additional portion of the reduced design space $\mathcal{D}_\alpha$, (correct notation is $\mathcal{D}_{\alpha'}$, which we avoid for ease of notation) without further reducing its dimension. The inequality constraints are usually given as box constraints i.e., in the form of lower bounds and upper bounds. The lower bounds are determined by the feature size of a technology. The upper bounds are selected such that the transistors are not excessively large. With elementary algebraic transformations, all the geometry constraints can be combined into a single nonlinear vector inequality,

which is interpreted element wise:

$$\bar{C}_g(\bar{\alpha}) \geq 0 \;\; \Leftrightarrow \forall_{i \in \{1 \ldots q\}} C_{gi}(\bar{\alpha}) \geq 0 \tag{3.5}$$

These constraints as functions of $\bar{\alpha}$ define a space, which we call as sample space $\mathcal{D}_g$, defined as

$$\mathcal{D}_g = \{\bar{\alpha} \,|\, \bar{C}_g(\bar{\alpha}) \geq 0\} \tag{3.6}$$

Clearly $\mathcal{D}_g \subset \mathcal{D}_\alpha$. A two dimensional projection of a four dimensional sample space is illustrated in Fig. 3.1.

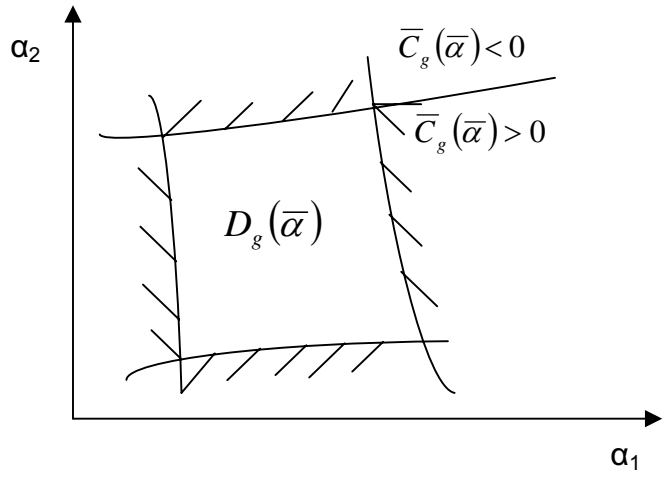


Figure 3.1: 2D projection of a four dimensional sample space

The extraction of training data for $\bar{X}$ and $\bar{\rho}$ of an analog component block is restricted to $\mathcal{D}_g(\bar{\alpha})$. The data generation process is generally an expensive process. Strategies from design of experiments (DOE) provide a mathematical basis to select a limited but optimal set of sample points for training data generation. In the present work, these points are generated using a Halton sequence generator [55]. A Halton sequence generator is a quasi-random number generator which generates a set of uniformly distributed random points in the sample space. This ensures a uniform and unbiased representation of the sample space.

3.2.2 Training Data Generation and Scaling

From (3.3) and (3.4), we see that the inputs ($\bar{X}$) and output ($\bar{\rho}$) of high-level performance models are functions of transistor level parameters $\bar{\alpha}$. The inputs and the

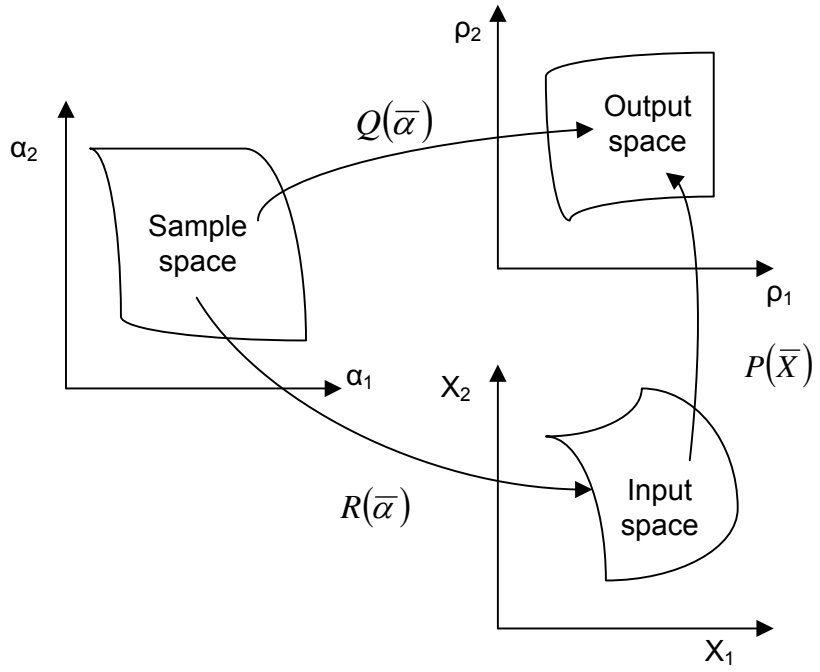


Figure 3.2: Nonlinear relation between sample space and input, output space

outputs are electrical parameters, whereas $\bar{\alpha}$ is a set of geometry parameters. The functions $(\mathcal{R}, \mathcal{Q})$ for mapping the geometry parameters to electrical parameters are complex nonlinear functions, considering deep submicron effects of MOS transistors. In this work, this is achieved element-wise through circuit simulation process, which is accepted to be the most accurate technique. The relationships are illustrated in Fig. 3.2. $\mathcal{R}$ and $\mathcal{Q}$ are used for generating the training data and $\mathcal{P}$ is the performance model to be constructed.

The training data generation process is outlined in Fig. 3.3. For each input sample (transistor sizes) extracted from the sample space $\mathcal{D}_g$, the chosen circuit topology of a component block is simulated using SPICE through Cadence Spectre tool. The BSIM3v3 model is used for simulation, ensuring that important deep submicron effects are considered while generating the training set. Depending upon the selected input-output parameters of an estimation function, it is necessary to construct a set of test benches which would provide sufficient data to facilitate automatic extraction of these parameters via postprocessing of SPICE output files. The commonly used SPICE analysis are ac analysis, transient analysis, dc sweep etc. The voltages and currents at the various nodes of the circuit are also measured. A set of constraints, referred to as feasibility constraints are then applied so that

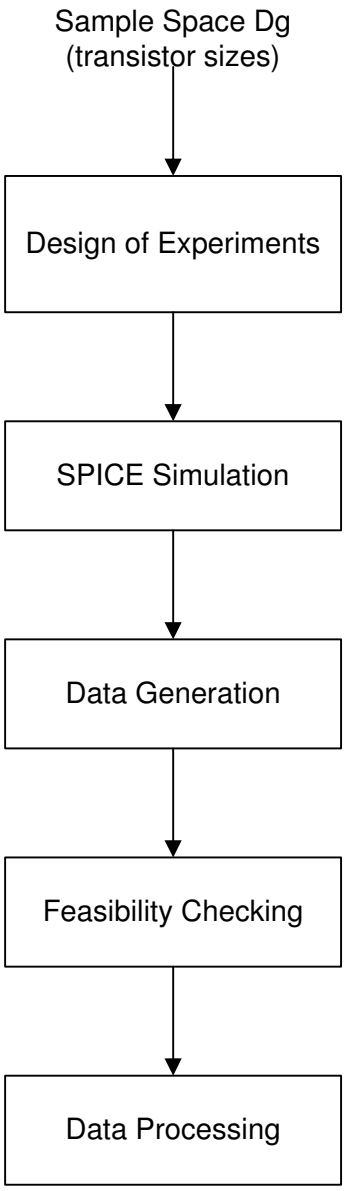


Figure 3.3: An outline of the procedure for generation of training data

only feasible data are considered for training.

The generated input-output data are considered to be feasible, if either they themselves satisfy a set of constraints or the mapping procedures $(\mathcal{R}, \mathcal{Q})$ through which they are generated satisfy a set of constraints. The constraints are as follows [56, 28, 57]:

1. Functionality constraints C_f : These constraints are applied on the measured node voltages and currents. They ensure correct functionality of the circuit and are expressed as

$$C_f = \{f_k(v, i) \geq 0 \;\; k = 1, 2, ..., n_f\} \tag{3.7}$$

   For example, the transistors of a differential pair must work in saturation.

2. Performance constraints C_p : These are applied directly on the input-output parameters, depending upon an application system. These are expressed as

$$C_p = \{f_k(\bar{\rho}) \geq 0 \;\; f_k(\bar{X}) \geq 0 \;\; k = 1, 2, ..., n_p\} \tag{3.8}$$

   For example, the phase margin of an opamp must be greater than 45^0.

The total set of constraints for feasibility checking is thus $C = \{C_f \cup C_p\}$.

Data scaling is an essential step to improve the learning/training process of SVMs. Scaling of the data can be performed on their input and/or output parameters. Commonly suggested scaling schemes are linear scaling, log scaling, and two-sided log scaling. The present methodology employs both linear scaling as well as logarithmic scaling depending upon the parameters chosen. The following formula are used for linear and logarithmic scaling within an interval $[0, 1]$:

$$\text{Linear:} \quad d'_j = \frac{d_j - lb}{ub - lb} \tag{3.9}$$

$$\text{Logarithmic:} \quad d'_j = \frac{\log\left(\frac{d_j}{lb}\right)}{\log\left(\frac{ub}{lb}\right)} \tag{3.10}$$

where d_j is the unscaled j^{th} data of any parameter bounded within the interval $[lb, ub]$. Linear scaling of data balances the ranges of different inputs or outputs. Applying log scale to data with large variations balances large and small magnitudes of the same parameter in different regions of the model.

3.2.3 Regression Using LS-SVR

The detailed theory of LS-SVM based function estimation process has been discussed in the appendix A.1. In principle, LS-SVM always fits a linear relation

$$\bar{\rho} = w\bar{X} + b \tag{3.11}$$

between the output set ($\bar{\rho}$) and the dependent variable set ($\bar{X}$). The best relation is the one that minimizes the cost function $\mathcal{J}$ containing a penalized regression error term [58]:

$$\mathcal{J} = \frac{1}{2}w^T w + \gamma \sum_{k=1}^{N_{tr}} e_k^2 \tag{3.12}$$

subject to

$$\bar{\rho}_k = w^T \varphi(\bar{X}_k) + b + e_k \quad k = 1, 2, ..., N_{tr} \tag{3.13}$$

where N_{tr} denotes the total number of training data set and the suffix k denotes the index of training set, i.e., k^{th} training data. This formulation involves the trade-off between a cost function term (first term in (3.12)) and a sum of squared errors (second term in (3.12)) governed by the trade-off parameter γ. The term $\frac{1}{2}w^T w$ determines the 'smoothness' of the resulting model. γ is referred to as regularization parameter. LS-SVM considers the optimization problem to be a constrained optimization problem and uses dual Lagrangian-based formulation

$$\mathcal{L} = \mathcal{J}(w, e) - \sum_{k=1}^{N_{tr}} \alpha_k \left(w^T \phi(\bar{X}_k) + b + e_k - \rho_k \right) \tag{3.14}$$

and applying 'kernel trick', we arrive at the final model (c.f. appendix A.1)

$$\bar{\rho} = \sum_{k=1}^{N_{tr}} \alpha_k K(\bar{X}_k, \bar{X}) + b \tag{3.15}$$

where $\alpha_k = \gamma e_k$ and $K(\bar{X}_k, \bar{X})$ is called as kernel function. The kernel function maps the sample space to a high-dimensional feature space. It makes the SVM representation independent of the dimensionality of the sample space [58]. There are several choices of the kernel function such as linear kernel, polynomial kernel, radial basis function (RBF) kernel, sigmoid kernel etc. The present methodology

employs RBF function as the kernel. This is defined as

$$K(\bar{X}_k, \bar{X}) = exp\left(-\left|\left|\bar{X}_k - \bar{X}\right|\right|^2 / \sigma^2\right) \tag{3.16}$$

where σ^2 is a parameter of the kernel and controls the width of the kernel function. The reasons for this choice are: First, RBF kernel non linearly maps samples into a higher dimensional space unlike the linear kernel. Furthermore, the linear kernel is a special case of RBF as argued in [59]. In addition, the sigmoid kernel behaves like RBF for certain parameters. Second, the number of hyper parameters influencing the complexity of model selection is comparatively less for RBF kernel. The polynomial kernel has more hyper parameters than the RBF kernel. Finally, the RBF kernel has less numerical difficulties in evaluating. This function has been used as kernel function in constructing circuit-level performance models by many researchers [32, 15].

3.2.3.1 Selection of Hyperparameters

To obtain good performances, some parameters in the SVM models have to be chosen carefully. These parameters include: (i) the regularization parameter γ, which determines the trade off between minimizing the training error and minimizing the model complexity; and (ii) parameter (σ^2) of the kernel function that implicitly defines the nonlinear mapping from the input space to some high-dimensional feature space. These higher level parameters are usually referred as hyper parameters. In general, in any classification or regression problem, if the hyper parameters of the model are not well selected, the predicted results will not be good enough. Tuning of these hyper parameters is usually done by minimizing the estimated generalization error. The techniques used for estimating the generalization error in the present methodology are

1. Hold-out method: This is a simple technique for estimating the generalization error. The data set is separated into two sets, called the training set and the testing set. The SVM is constructed using the training set only. Then it is tested using the test data set. These data are completely unknown to the estimator. The errors it makes are accumulated as before to give the mean test set error, which is used to evaluate the model. This method is very fast. However, its evaluation can have a high variance. The evaluation may depend

heavily on which data points end up in the training set and which end up in the test set, and thus the evaluation may be significantly different depending on how the division is made.

2. 'k'-fold cross validation method: This is a popular method for estimating the generalization error. The training data is randomly split into k mutually exclusive subsets (the folds) of approximately equal size. The SVM decision rule is obtained using $k-1$ of the subsets and then tested on the subset left out. This procedure is repeated k times and in this fashion each subset is used for testing once. Averaging the test error over the k trials gives an estimate of the expected generalization error. The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once, and gets to be in a training set $k-1$ times. The variance of the resulting estimate is reduced as k is increased. The disadvantage of this method is that the training algorithm has to be rerun from scratch k times, which means it takes k times as much computation to make an evaluation.

The present methodology employs two techniques for selecting optimal values of the model hyper parameters. The first one is a grid search technique and the other one is a genetic algorithm-based technique.

A. Grid Search Technique: The basic steps of the grid search-based technique is outlined below:

1. Consider a grid space of (γ, σ^2), defined with $\log_2 \gamma \in \{lb_\gamma, ub_\gamma\}$ and $\log_2 \sigma^2 \in \{lb_{\sigma^2}, ub_{\sigma^2}\}$, where $[lb_\gamma, ub_\gamma]$ and $[lb_{\sigma^2}, ub_{\sigma^2}]$ define the boundary of the grid space.

2. For each pair, estimate the generalization error through hold-out/k-fold cross validation technique.

3. Choose the pair that leads to the lowest error.

4. Use the best parameter to create a model as predictor.

The grid search technique is simple. However, this is computationally expensive since this is an exhaustive search technique. In addition, this is a tricky task since

a suitable sampling step varies from kernel to kernel and the grid interval may not be easy to locate without prior knowledge of the problem.

B. Genetic Algorithm-based Technique:

The task of selection of hyper parameters is same as an optima searching task, and each point in the search space represents one feasible solution (specific hyper-parameters). An outline of a simple GA-based process is shown in Fig. 3.4. The chromosome consists of two parts, $\log_2 \gamma$ and $\log_2 \sigma^2$. Binary encoding scheme is used to represent the chromosome. During the evolutionary process of GA, a model is trained with the current hyper parameter values. The hold-out method as well as well as the k-fold cross validation method are used for estimating the generalization error. The fitness of a chromosome depends on the average relative error (ARE) calculated over the test samples. The fitness function is defined as

$$\text{fitness} = \frac{1}{ARE(\gamma, \sigma^2)} \tag{3.17}$$

Thus, maximizing the fitness value corresponds to minimizing the predicted error. The ARE function is defined as

$$ARE = \frac{1}{N_{te}\rho'} \sum_{1}^{N_{te}} (\rho - \rho') \tag{3.18}$$

Here N_{te}, ρ and ρ' are the number of test data, SVM estimator output and corresponding SPICE simulated value, respectively. The fitness of each chromosome is taken to be average of five repetitions. This reduces the stochastic variability of model training process in GA-based LS-SVM. Roulette wheel selection technique is used for the selection operation. Besides, in order to keep the best chromosome in every generation, the idea of ellitism is adopted. Uniform crossover technique is used in the crossover operation. When the difference between the estimated error of the child population and that of the parent population is less than 0.001 over certain fixed generations, the whole process is terminated and outputs the result.

3.2.4 Quality Measures

Statistical functions are generally used to assess the quality of the generated estimator. The ARE function defined in (3.18) is one measure. Another commonly used

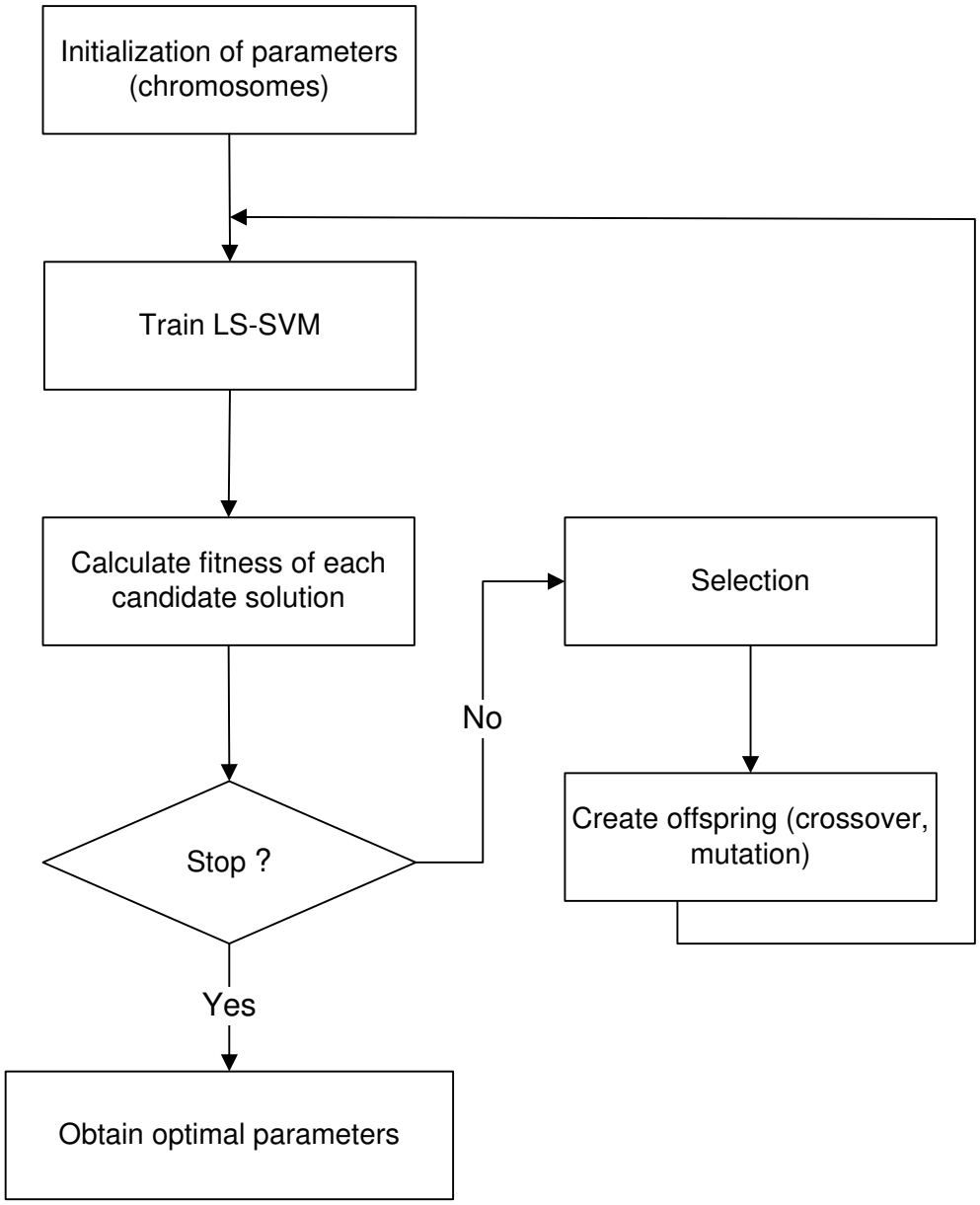


Figure 3.4: Outline of GA-based hyperparameter selection procedure

measure is correlation coefficient. This is defined as follows

$$\text{Correlation Coefficient}(R) = \frac{N_{te}\sum \rho\rho' - \sum \rho \sum \rho'}{\sqrt{\left[N_{te}\sum \rho^2 - \left(\sum \rho\right)^2\right]\left[N_{te}\sum \rho'^2 - \left(\sum \rho'\right)^2\right]}} \quad (3.19)$$

Correlation coefficient is a measure of how closely the LS-SVM output fits with the target values. It is a number between 0 and 1. If there is no linear relationship between the estimated values and the actual targets then the correlation coefficient is 0. If the number is equal to 1.0, then there is a perfect fit between the targets and the outputs. Thus, higher the correlation coefficient, the better it is.

3.3 Comparison with Existing Methodologies

The present methodology uses non parametric regression technique for constructing the high-level performance models. Compared with the other modeling methodologies employing symbolic analysis technique or simulation-based technique, the advantages of the present methodology are: (i) Full accuracy of SPICE simulations and advanced device models, such as BSIM3v3 is used to generate the performance models. The models are thus accurate compared to real simulation results. (ii) There is no need for any *a priori* knowledge about the unknown dependency between the inputs and the outputs. (iii) The generalization ability of the model is high. (iv) The model construction time is low and the construction process does not require any detailed circuit design knowledge.

The EsteMate methodology [13] for constructing high-level performance estimation models using artificial neural network and the SVM-based methodology discussed in [15, 14] are closely related with the present methodology. The methodology that we have developed, however has number of advantages over them. These are

1. Compared to the EsteMate methodology, simpler sampling strategies are required. This is because in EsteMate, the specification parameters of a component block constitute a sample space for training data generation. The specification parameters are electrical parameters and there exists strong nonlinear correlations among them. Therefore, sophisticated sampling strategies are required for constructing models with good generalization ability in the EsteMate methodology. On the other hand in our method, the transistor sizes define the sample space. These being geometry parameters are independently

distributed.

2. In EsteMate, for each samples, a complete circuit sizing task using global optimization algorithm is required for generation of the training data. This is usually prohibitively time consuming. On the other hand in our method, simple circuit simulations using the sampled transistor sizes are required for data generation. Therefore, the cost of training data generation in our method is much less compared to the EsteMate[13].

3. The generalization ability of the models constructed with our methodology is better than that generated through the EsteMate methodology. This is because the latter uses ANN regression technique. Neural network approaches suffer difficulties with generalization, producing models that can overfit the data. This is a consequence of the optimization algorithms used for parameter selection and the statistical measures used to select the 'best' model. SVM formulation on the other hand, is based upon structural risk minimization (SRM) principle [60], which has been shown to be superior to traditional empirical risk minimization (ERM) principle, employed by the conventional neural networks. SRM minimizes an upper bound on the expected risk, as opposed to ERM that minimizes the error on the training data. Therefore an SVM has greater generalization capability.

4. The SVM-based methodology as presented in [15] uses heuristic knowledge to determine the model hyper parameters. The present methodology uses optimization techniques to determine optimal values for them. Genetic algorithm-based methodology for determination of optimal values for the model hyper parameters is found to be faster compared to the grid search technique employed in [14].

3.4 Topology Sizing Methodology using GA

In this section, we discuss a genetic algorithm-based methodology for a topology sizing process employing the constructed LS-SVM performance models.

An outline of the flow is shown in Fig. 3.5. A high-level topology is regarded as a multi dimensional space, in which the topology parameters (specification parameters of the constituent component blocks) are the dimensions. The valid design space for

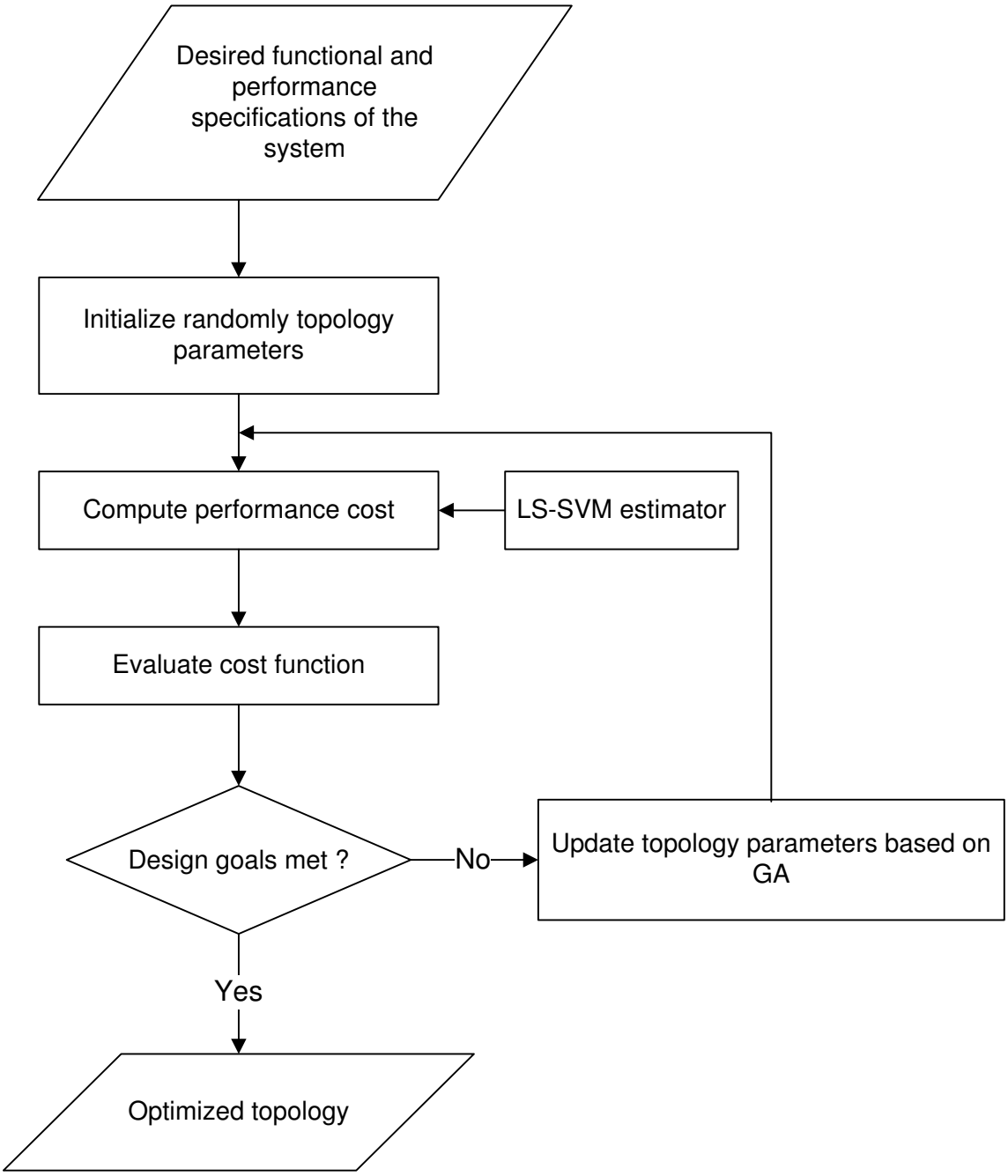


Figure 3.5: Topology sizing methodology using GA optimizer with LS-SVM model

a particular application consists of those points which satisfy the design constraints. The optimization algorithm searches in this valid design space for the point which optimizes a cost function. The optimization targets, i.e. the performance parameters to be optimized and system-specifications to be satisfied are specified by the user. The GA optimizer generates a set of chromosomes, each representing a combination of topology parameters in the given design space. A performance estimation model for estimating the performance of a topology of the entire system is constructed by combining the LS-SVM models of the individual component blocks through analytical formulae. The performance estimation model takes each combination of topology parameters and produces an estimation of the desired performance cost of the topology as the output. A cost function is computed using these estimated performance values. The chromosomes are updated according to their fitness, related to the cost function. This process continues until a desired cost function objective is achieved or a maximum number of iterations are executed.

This topology sizing process can be used for the topology selection process, as discussed in section 2.4.1.

3.5 Experimental Results

In this section, we provide experimental results demonstrating the methodologies described above. The entire methodology has been implemented in Matlab environment and the training of the LS-SVM has been done using Matlab toolbox [61].

3.5.1 Experiment 1:

A two stage CMOS operational transconductance amplifier (OTA) is shown in Fig. 3.6. The technology is $0.18\mu m$ CMOS process, with a supply voltage of $1.8V$. The transistor level parameters along with the various feasibility constraints are listed in Table 3.1. The functional constraints ensure that all the transistors are on and are in the saturation region with some user defined margin. We consider the problem of modeling input referred thermal noise (ρ_1), power consumption (ρ_2) and output impedance (ρ_3) as functions of DC gain (X_1), bandwidth (X_2) and slew rate (X_3). From the sample space defined by the transistor sizes, a set of 5000 samples is generated using a Halton sequence generator. These are simulated through ac analysis, operating point analysis, noise analysis and transient analysis

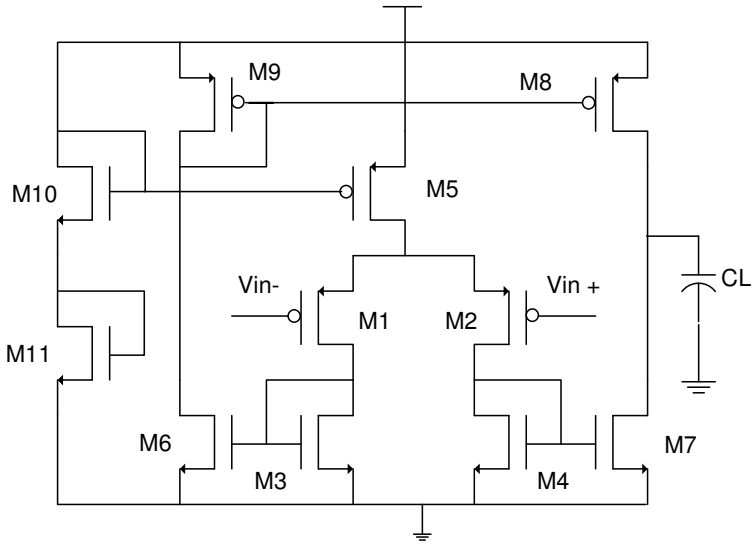


Figure 3.6: PMOS OTA circuit

using SPICE program. Out of all samples, only 1027 samples are found to satisfy the functional and performance constraints listed in Table 3.1.

The estimation functions are generated using LS-SVR technique. The generalization errors are estimated through the hold-out method and a 5-fold cross validation method. The hyper parameters are computed through the grid search and the GA-based technique. In the grid search technique, the hyperparameters (σ^2, γ) are restricted within the range $[0.1, 6.1]$ and $[10, 510]$. The grid search algorithm is performed with a step size of 0.6 in σ^2 and 10 in γ. The determined hyper parameter values along with the quality measures and the training time are reported in Table 3.2 and Table 3.3 for the hold-out method and the cross validation method respectively. From the results we observe that the generalization ability of the models are high when the errors are estimated using the cross validation method. However, the cross validation method is much slower compared to the hold-out method.

For GA, the population size is taken to be ten times the number of the optimization variables. The crossover probability and the mutation probability are taken as 0.8 and 0.05 respectively. These are determined through a trial and error process. The results obtained are reported in Table 3.4 and 3.5. From the results the above observations are also noted.

A comparison between the grid-search technique and the GA-based technique with respect to accuracy (ARE), correlation coefficient (R) and required training time is made in Table 3.6. All the experiments are performed on a PC with PIV

Table 3.1: Transistor Sizes and Feasibility Constraints for OTA

	Parameters	Ranges
Transistor Sizes Geometry Constraints	$W_1 = W_2$	$[280nm, 400\mu m]$
	$W_3 = W_4 = W_6 = W_7$	$[1\mu m, 20\mu m]$
	$W_8 = W_9$	$[280nm, 10\mu m]$
	W_5	$[1\mu m, 50\mu m]$
	C_L	$[1pF, 10pF]$
	Parameters	Range
Functional Constraints	$V_{gs} - V_{th}$	$\geq 0.1V$
	V_{op}	$\approx 0.9V$
	V_{off}	$\leq 2mV$
	Slew rate	$[0.1V/\mu s, 20V/\mu s]$
Performance Constraints	Bandwidth	$\geq 2MHz$
	DC Gain	≥ 70 dB
	Phase margin	$[45^0, 60^0]$

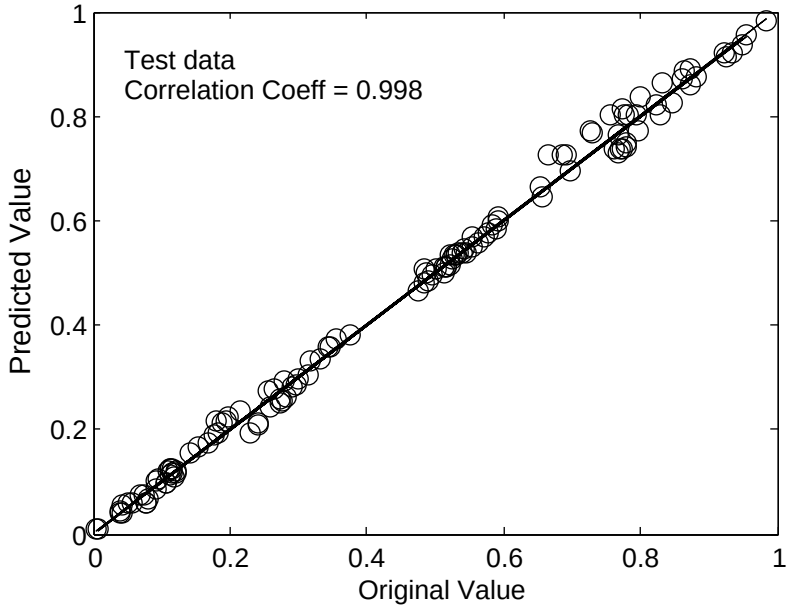


Figure 3.7: Scatter plot of estimated and original values for noise model with normalized test data

Table 3.2: Grid search technique using hold out method

Model	σ^2	γ	$ARE(\%)$		R		T_{tr}
			Training	Test	Training	Test	(min)
ρ_1	3.43	173.26	1.82	2.48	0.999	0.998	118.19
ρ_2	2.10	112.04	2.32	4.18	0.918	0.905	117.83
ρ_3	5.43	387.55	2.02	3.14	0.999	0.937	118.13

Table 3.3: Grid search technique using 5-fold cross validation method

Model	σ^2	γ	$ARE(\%)$		R		T_{tr}
			Training	Test	Training	Test	(min)
ρ_1	4.10	326.32	1.27	1.33	0.999	0.999	583.12
ρ_2	2.76	112.04	2.37	2.42	0.980	0.970	583.62
ρ_3	5.33	142.65	1.82	1.85	0.998	0.998	582.67

3.00 GHz processor and 512 MB RAM. We observe from the comparison that the accuracy of SVM models constructed using the grid search technique and the GA-based technique are almost same. However, the GA-based technique is at least ten times faster than the grid search method. From (3.2), we conclude that the construction cost of the GA-based method is much lower than the grid search-based method, since the data generation time is same for both the methods.

The scatter plots of SPICE-simulated and LS-SVM estimated values for normalized test data of the three models are shown in Fig. 3.7, Fig. 3.8 and Fig. 3.9 respectively. These scatter plots illustrate the correlation between the SPICE simulated and the LS-SVM estimated test data. Perfect accuracy would result in the data points forming a straight line along the diagonal axis.

3.5.2 Experiment 2:

The objective of this experimentation is to quantitatively compare between our methodology and the EsteMate [13]. The power consumption model is reconstructed

Table 3.4: GA technique using hold out method

Model	σ^2	γ	$ARE(\%)$		R		T_{tr}
			Training	Test	Training	Test	(min)
ρ_1	2.38	250.13	2.16	3.38	0.999	0.998	10.06
ρ_2	5.62	480.19	2.12	3.82	0.994	0.961	10.83
ρ_3	5.19	140.15	1.98	2.90	0.999	0.998	10.56

Table 3.5: GA technique using 5-fold cross validation

Model	σ^2	γ	$ARE(\%)$		R		T_{tr}
			Training	Test	Training	Test	(min)
ρ_1	3.98	350.13	1.35	1.36	0.999	0.999	46.66
ρ_2	3.02	150.19	2.12	3.02	0.994	0.980	45.83
ρ_3	5.32	540.15	1.81	1.90	0.999	0.990	46.13

Table 3.6: Comparison between GA and Grid search technique for LS-SVM construction

Model	Algorithm	σ^2	γ	$ARE(\%)$		R		T_{tr}
				Training	Test	Training	Test	(min)
ρ_1	GA	2.38	250.13	2.16	3.38	0.999	0.998	10.06
	Grid Search	3.43	173.26	1.82	2.48	0.999	0.998	118.19
ρ_2	GA	5.62	480.19	2.12	3.82	0.994	0.961	10.83
	Grid Search	2.10	112.04	2.32	4.18	0.980	0.905	117.83
ρ_3	GA	5.19	140.15	1.98	2.90	0.999	0.998	10.56
	Grid Search	5.43	387.55	2.02	3.14	0.999	0.937	118.13

using the EsteMate technique. The specification parameter space is sampled randomly. A set of 5000 samples are considered. For each selected sample an optimal sizing is performed and the resulting power consumption is measured. The sizing is done with a simulated annealing-based optimization procedure and standard analytical equations relating transistor sizes with the specification parameters [62]. Of these, 3205 samples are accepted and the rest are rejected. The determination of the training set took 10 hours of CPU time. The training is done through an artificial neural network structure with two hidden layers. The number of neurons for the first layer is 9, the number of neurons for the second layer is 6. The hold-out method is used for estimating the generalization ability.

A comparison between the two methodologies is reported in Table 3.7. From the results, we find that the data generation time is much less in our method compared to the EsteMate method. This is because, in our approach, transistor sizes are directly used as sample parameters, in contrast to the EsteMate method where these are determined through an optimization procedure. In addition, we find that the generalization ability of our method is better than the EsteMate method. This is due to the use of SVM in our method.

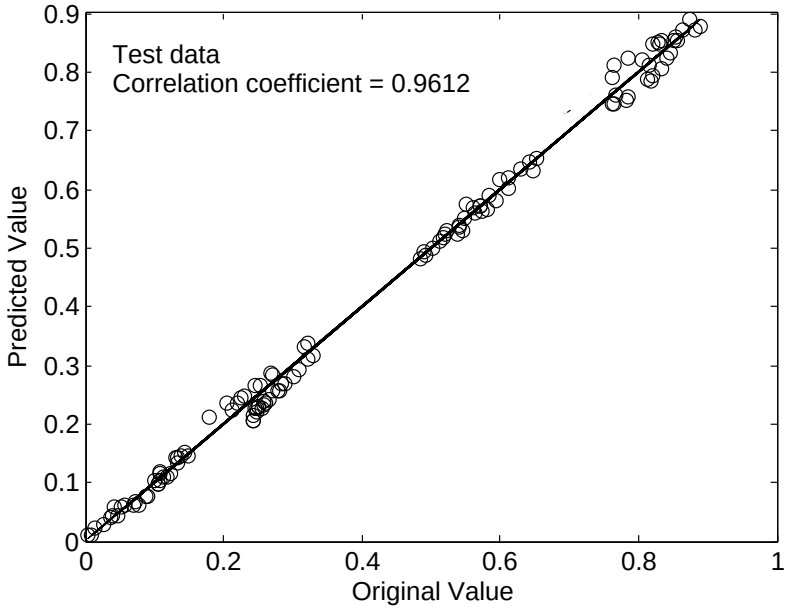


Figure 3.8: Scatter plot of estimated and original values for power model with normalized test data

Table 3.7: Comparison between our methodology and of EsteMate

Method	# Samples		$ARE(\%)$		Generation time	Training time
	Training	Test	Training	Test		
Our	821	206	2.12	3.82	14 min	10.83 min
EsteMate [13]	2564	636	2.88	6.53	10 hour	21 min

3.5.3 Experiment 3:

The objective of this experimentation is to demonstrate the topology sizing process. We choose a complete analog system - interface electronics for MEMS capacitive sensor system as shown in Fig. 3.10. In this configuration, a half-bridge consisting of the sense capacitors C_1, C_2 is formed and driven by two pulse signals with 180^0 phase difference. The amplitude of the bridge output V_x, is proportional to the capacitance change ΔC and is amplified by a voltage amplifier. The final output voltage V_{out}, is given by

$$V_{out} = V_0 \frac{2\Delta C}{2C_0 + C_p} A_v \tag{3.20}$$

where C_0 is the nominal capacitance value, C_p is the parasitic capacitance value at the sensor node, V_0 is the amplitude of the applied ac signal and A_v is the gain of the system, depending upon the desired output voltage sensitivity. The topology

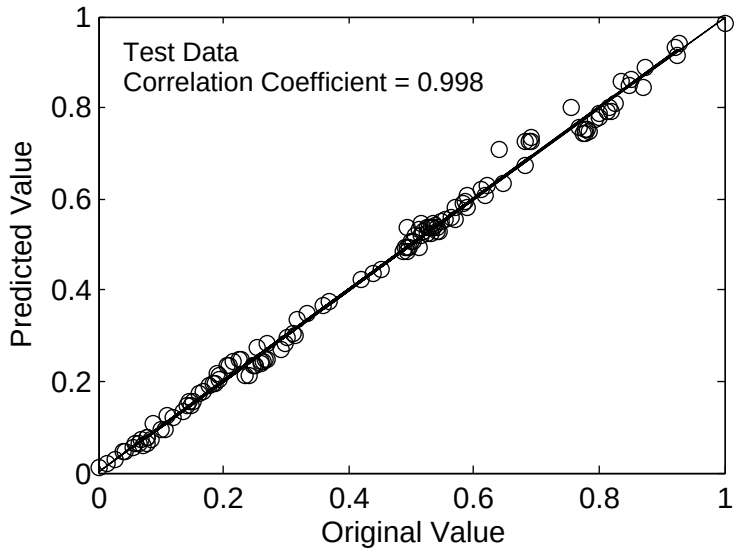


Figure 3.9: Scatter plot of estimated and original values for output impedance model with normalized test data

Table 3.8: Functional Specs and Design constraints

Parameters	Desired Specs
Sensing Capacitance	100 fF
Capacitance Sensitivity	0.4 fF
Linear Range	$\pm$ 6 g
Modulation Frequency	1MHz
Modulation Voltage	500m V
Input Voltage Sensitivity	$\geq$ 1 mV/g
Output Voltage Sensitivity	$\geq$ 100 mV/g
Cut-off frequency	$\leq$ 35 KHz

employs a chopper modulation technique for low $1/f$ noise purpose.

The desired functional specifications to be satisfied are output voltage sensitivity and cut-off frequency of the filter. The performance parameters to be optimized are input-referred thermal noise, total power consumption and parasitic capacitance at the sensor node V_x. The functional specifications and design constraints for the system are based on [63] and are listed in Table 3.8. The synthesizable component blocks are the pre-amplifier (PA), inverter (IN) of the phase demodulator, low pass filter (LF) and the output amplifier (OA). These are constructed using OTAs and capacitors.

High-level performance models for the synthesizable component blocks corre-

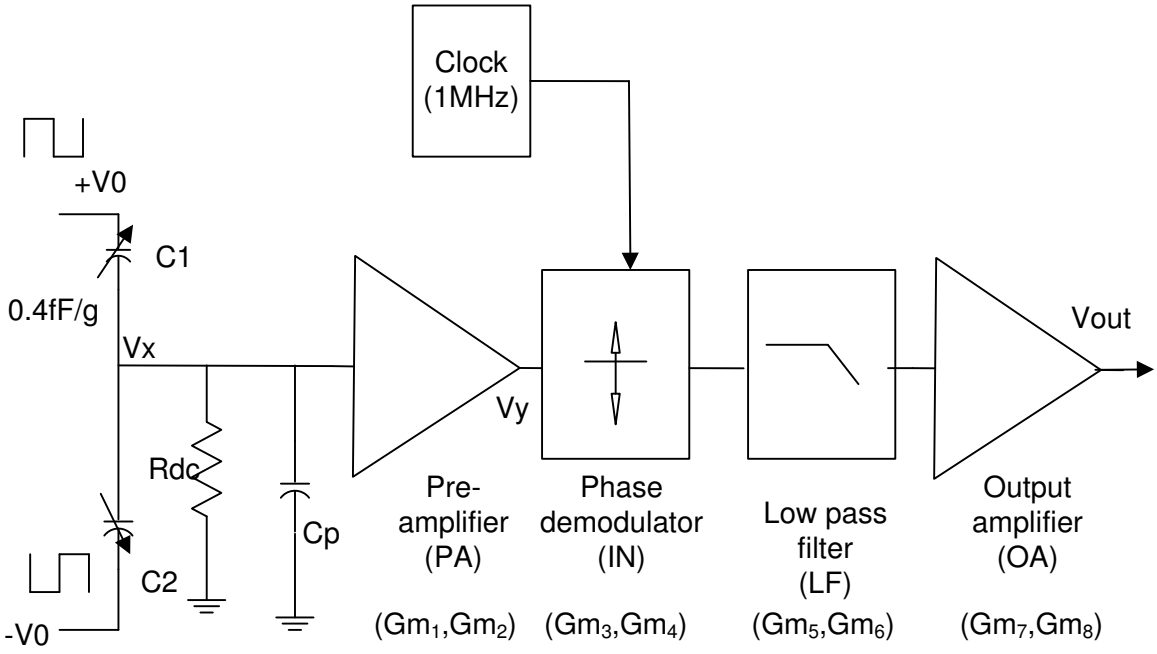


Figure 3.10: Voltage sensing configuration of the interface electronics for MEMS capacitive sensor

sponding to the performance parameters – (i) input referred thermal noise, (ii) power consumption and (iii) sensor node parasitics are constructed. The specification parameters which have dominant influence on the first two performances as well as the functional specification, i.e. the gain and the cut-off frequency are the transconductance values of all the OTAs involved. These are considered as the high-level design parameters. On the other hand for the last performance parameter, i.e. sensor node parasitics, transconductance value of the first OTA of the pre-amplifier block is the single design parameter. The geometry constraints and the feasibility constraints for the PA block of the topology are tabulated in Table 3.9. Similar types of constraints are considered for the other component blocks also. The input-output parameters are extracted through techniques discussed earlier. The sensor node parasitic capacitance is measured utilizing the half-bridge circuit shown in Fig. 3.10, with only one amplifier block. Considering $\Delta C = 5fF, C_0 = 65fF$, a square wave signal with amplitude $V_0 = 500mV$ is applied and transient analysis is performed. Measuring the signal at the node V_x, C_p is calculated using (3.20).

Table 3.10 shows the hyperparameter values, percentage average relative error and correlation coefficient of the constructed performance models for the preamplifier, with respect to SPICE simulated value. The variation of the noise, power

Table 3.9: Transistor Sizes and Feasibility Constraints for Preamplifier

Transistor Sizes	Geometry Constraints	
	Gm_1	Gm_2
$W_1 = W_2$	$[280nm, 400\mu m]$	$[280nm, 200\mu m]$
$W_3 = W_4 = W_6 = W_7$	$[1\mu m, 20\mu m]$	$[1\mu m, 20\mu m]$
$W_8 = W_9$	$[280nm, 10\mu m]$	$[280nm, 10\mu m]$
I_{bias}	$[1\mu A, 40\mu A]$	$[1\mu A, 10\mu A]$
Functional constraints	Parameters	Range
	$V_{gs} - V_{th}$	$\geq 0.1V$
	V_{op}	$\approx 0.9V$
	V_{off}	$\leq 2mV$
Performance constraints	Input linearity	$\geq 15mV$
	Swing	$\geq 750mV$
	Bandwidth	$\geq 2MHz$
	Phase margin	$[45^0, 60^0]$

Table 3.10: Accuracy of Preamplifier block

Models	σ^2	γ	Training		Test	
			ARE	R	ARE	R
Noise	2.88	288.93	1.25	0.9991	1.75	0.9991
Power	1.18	203.18	2.05	0.9989	2.35	0.9989
Parasitics	3.25	189.79	0.58	0.9999	0.62	0.9999

and input parasitics of the preamplifier block with the high-level design parameters (Gm_1, Gm_2) are shown in Fig. 3.11, Fig. 3.12 and Fig. 3.13 respectively. From Fig. 3.11, we see that Gm_1 has the major contribution in comparison to Gm_2 and as Gm_1 increases, the input referred thermal noise decreases. From Fig. 3.12, we see that as Gm_1 and Gm_2 increases, the power consumption increases. From Fig. 3.13, we see that as Gm_1 increases with constant bias current, the parasitic increases. This is due to increase of input transistor width.

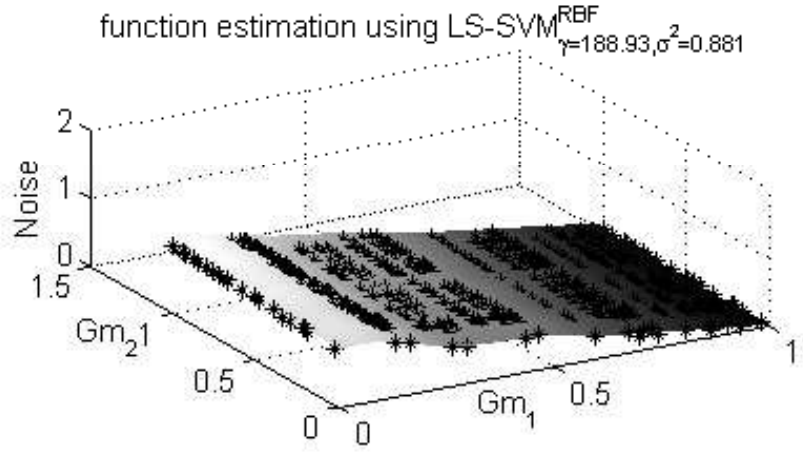


Figure 3.11: Noise as function of Gm_1 and Gm_2

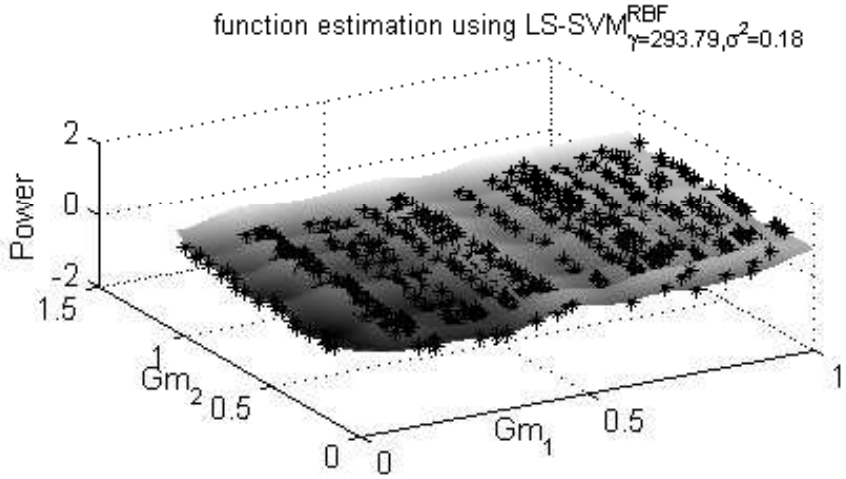


Figure 3.12: Power as function of Gm_1 and Gm_2

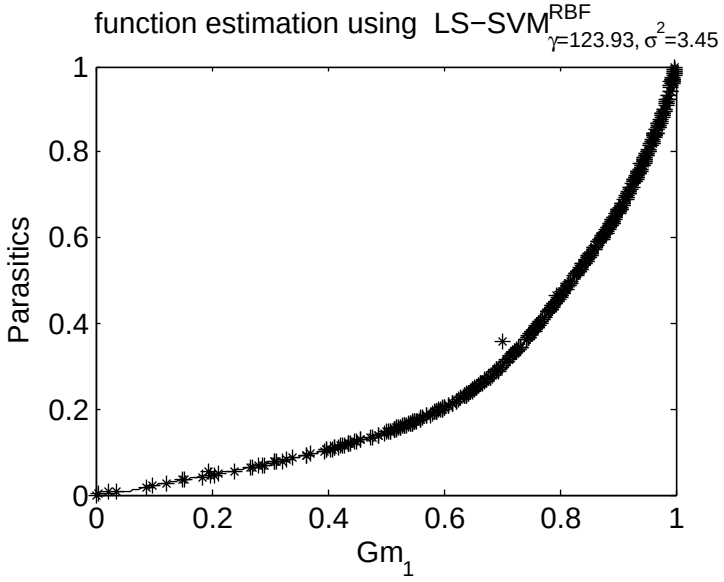


Figure 3.13: Input parasitics as function of Gm_1

The performance models corresponding to the noise and the power consumption for the PA block are reused for the other component blocks. The performances of the individual component blocks are combined to estimate the performances of the total system. The input referred noise and power consumption of the total system is given by

$$\begin{aligned} V_{nT}^2 &= V_{n1}^2(Gm_1, Gm_2) + \frac{V_{n2}^2(Gm_3, Gm_4)}{A_1^2} + \\ &\quad \frac{V_{n3}^2(Gm_5, Gm_6)}{A_1^2} + \frac{V_{n4}^2(Gm_7, Gm_8)}{A_1^2} \qquad (3.21) \\ P_T &= P_1(Gm_1, Gm_2) + P_2(Gm_1, Gm_2) + \\ &\quad P_3(Gm_1, Gm_2) + P_4(Gm_1, Gm_2) \qquad (3.22) \end{aligned}$$

A_1 is the gain of the preamplifier. The sensor node parasitics $P_a = P_a(Gm_1)$ is same as the input parasitics of the preamplifier. With these, the optimization problem

for topology sizing task is formulated as

$$\begin{aligned} \text{Minimize} \quad & \omega_1 V_{nT} + \omega_2 P_T + \omega_3 P_a \\ \text{such that} \quad & (V_{out})_{target} - V_{in}\left[\frac{Gm_1}{Gm_2}\frac{Gm_3}{Gm_4}\frac{Gm_5}{Gm_6}\frac{Gm_7}{Gm_8}\right] \leq \epsilon_1 \\ & f_c - \frac{Gm_6}{2\pi C_L} \leq \epsilon_2 \\ & Gm_{imin} \leq Gm_i \leq Gm_{imax} \\ & C_{Lmin} \leq C_L \leq C_{Lmax} \end{aligned} \tag{3.23}$$

where ω_i are associated weights.

The target output voltage sensitivity of the system (i.e. the total gain of the system) is taken as $145mV/g$ and the cut-off frequency is taken as 35 KHz. The synthesis procedure took 181 seconds on a PIV, 3.00 GHz processor PC with 512 MB RAM. The crossover and the mutation probability are taken as 0.85 and 0.05 respectively. These are determined through a trial and error process. Table 3.11 lists the synthesized values of the topology parameters, as obtained from the synthesis procedure.

To validate the synthesis procedure, we simulate the entire system at the circuit-level using SPICE. Exact values of Gm are not achievable often. In such cases, the nearest neighbouring values are realized. An approximate idea about the transistor sizes required to implement the synthesized Gm values are made from the large set of data gathered during the estimator construction. A comparison between the predicted performances and simulated values is presented in Table 3.12. We observe that the relative error between predicted performances and simulated performances in each case is not much high. However, for the total gain and the cut-off frequency, the error is quite high. This is because the circuit-level non-ideal effects have not been considered in the topology sizing process.

3.6 Conclusion

In this chapter a methodology for generation of high-level performance models for analog component blocks using non parametric regression technique has been presented. The transistor sizes of the component blocks define the sample space. The training data are generated through simple circuit simulation using SPICE. Least square support vector machine is used as the regression functions. The generaliza-

Table 3.11: Synthesized Topology Parameters

Topology Parameters	Synthesized Value
Gm_1	216.30 μS
Gm_2	14.67 μS
Gm_3	17.97 μS
Gm_4	16.80 μS
Gm_5	15.92 μS
Gm_6	13.96 μS
Gm_7	131.73 μS
Gm_8	16.15 μS
C_L	63 pF

Table 3.12: Comparison of Predicted performances and SPICE value

Performances	Pred	SPICE	Error %
Noise ($nV/\sqrt{Hz}$)	19.65	20.32	3.3
Power (μ W)	572.78	592	3.36
Parasitics (fF)	92.05	94.12	2.20
Sensitivity	145.16	138	4.93
Cut-off (KHz)	35.28	38	7.70

tion error have been estimated using a hold-out method and a k-fold cross validation method. The model hyper parameters are determined through a grid search technique and GA-based technique. LS-SVM based performance models have been utilized to develop a topology sizing process with an objective to determine the topology parameters such that the topology performances are optimized. From the experimental results, it has been found that the cross validation method estimates the generalization ability of the constructed models more accurately than the hold-out method. However, it is more time consuming than the latter. It has also been observed that the GA-based training process is faster compared to the grid search based process with almost same accuracy. In addition, it has been experimentally demonstrated that the data generation time in our approach is considerably less compared to the EsteMate approach.

Chapter 4

Top-Down Generation of an Optimal Topology

The various techniques for generation of an optimal component-level topology of analog systems have been discussed briefly in section 2.4 of chapter 2 of the dissertation. This chapter presents a methodology for the top-down technique of generation of an optimal component-level topology for linear analog systems. The topologies are generated from a transfer function model of the system using an analog computation model. Similarity transformation matrix is used as topology transformation operator. A simulated annealing-based optimization procedure selects an optimal topology based upon the performances of the topologies. The entire methodology is illustrated with continuous time $\Sigma\Delta$ modulator system as a case study. The advantage of the methodology is that the designer is able to specify the design goal and desired specifications at a higher level of abstraction in terms of transfer functions, while the design tool is able to generate an optimal component-level topology directly from transfer functions in a highly automated manner. In addition, the generated topology is guaranteed to work satisfactorily even under the presence of circuit-level non-idealities.

The chapter is organized as follows. The generic methodology is outlined in section 1. A review of similar works on $\Sigma\Delta$ modulator is described in section 2. The optimal topology generation methodology for $\Sigma\Delta$ modulator is described in section 3. A comparison between the present and the other existing methodologies is given in section 4. Experimental results are provided in section 5. Finally conclusion is drawn in section 6.

4.1 Generic Methodology for Linear Analog Systems

An outline of the methodology is shown in Fig. 4.1. The input to the topology generation process is a transfer function description of the linear analog system to be designed. The transfer function specifies the input-output behavior of the system in frequency domain. It is transformed into a time domain equivalent, the state space description. A state variable description gives sufficient insight to the system structure. This acts as a basis for generation of a component-level topology of the system using an analog computation model [64]. From a given state space model, infinite number of other state space models can be generated using similarity transformation matrix operation. Therefore, it is possible to generate infinite number of other topologies from an initial topology. These newly generated topologies have same behavioral properties but different performance properties. The task of topology exploration and selection of an optimal topology is performed at the state space model level. An optimal state space model is determined through a simulated annealing based optimization procedure. The performances of the topologies are used as metrics in the selection process. Once an optimal state space model is determined, a component-level topology is generated in two steps. In the first step, the model is realized by several functional component blocks, e.g., adder, integrator etc. In the second step, the functional component blocks are realized following different implementation styles like switch capacitor, active RC etc. This two-step process gives freedom to users in selecting appropriate implementation styles. The generated topology is then behaviorally simulated to check whether it satisfies all of the desired specifications in presence of circuit-level non-idealities. If it fails, the topology exploration and selection process is repeated and a new optimized topology is selected. The final output of the generation process is an optimal component-level topology which satisfies all of the desired specifications under circuit-level non-ideal conditions.

4.2 Related Work on $\Sigma\Delta$ Modulator Synthesis

The existing works on high-level design, synthesis and optimization of $\Sigma\Delta$ modulator are classified into three broad categories. The first category of works concentrate

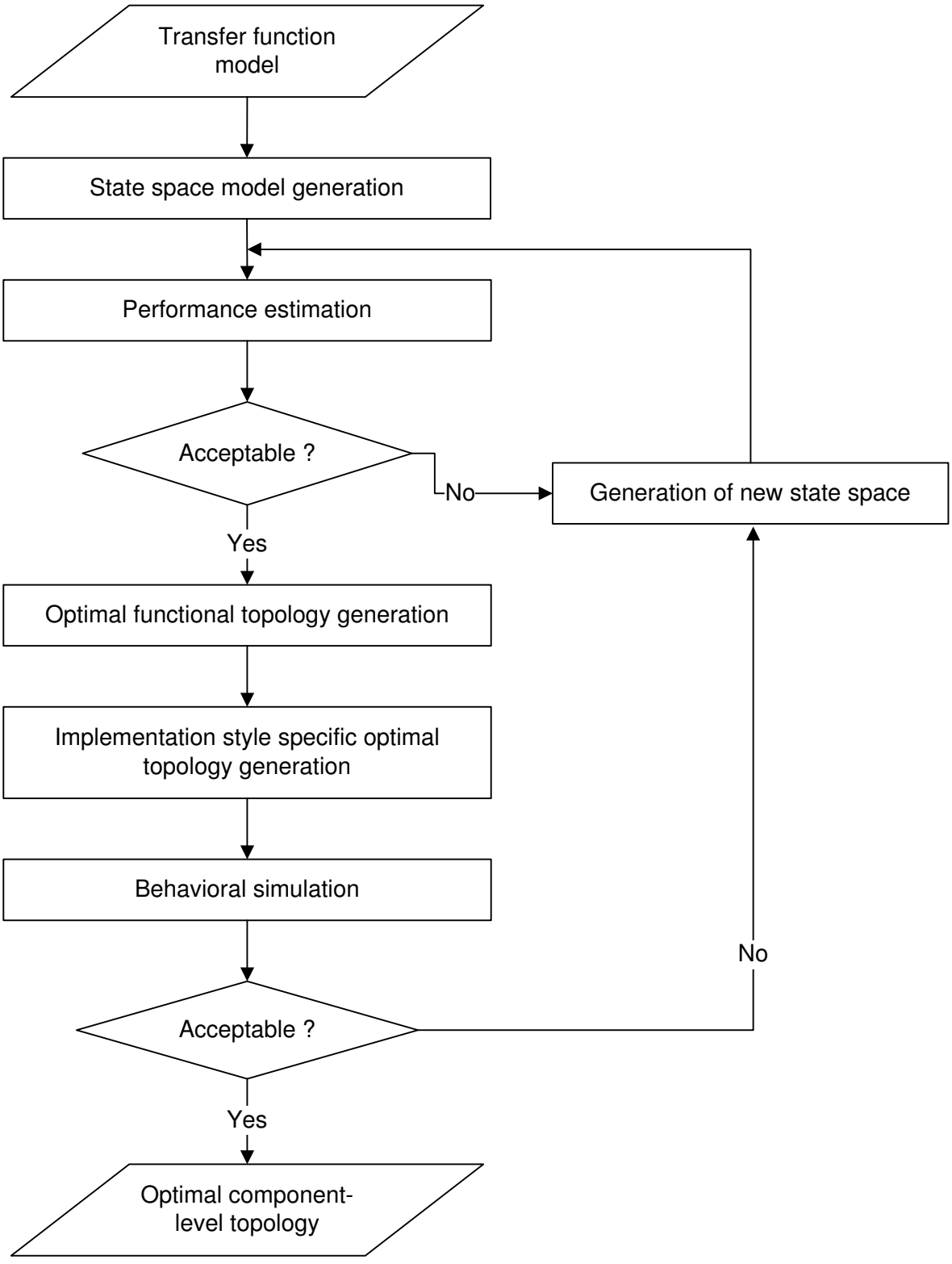


Figure 4.1: Top-down generation of an optimal topology for linear analog systems

primarily on the optimization of modulator coefficients [19, 65, 66, 67]. The basic principle of this type of design approach is to start with a set of popular $\Sigma\Delta$ modulator topologies, including single-loop, single-bit and multi-loop, multi-bit. The topology of the modulator is selected based upon designer's experience. Then the coefficients of the selected topology are calculated to optimize the modulator performances such as peak signal-to-noise ratio (SNR), dynamic range (DR) etc. This category of works follows the 'selection before or after sizing' principle discussed in section 2.4.1. In the second category of works [21, 22, 68], a set of selected topologies with optimized coefficients are stored in a library. For each topology, the specification parameters of the component-blocks are determined such that the SNR and DR are satisfied. Then a topology with the smallest power and/or area consumption is selected. As far as the task of topology generation is considered there is no difference between these two categories of works. The third category of works [24, 39, 69, 70] have come up recently. The tasks of topology generation, exploration and selection are the primary focus of these works. In [24], a generic representation of single-loop, single-bit modulator that describes all possible topologies is considered. A symbolic expression for noise transfer function NTF is derived for the generic topology. By equating the symbolic NTF to the desired NTF, a set of equations in terms of the modulator coefficients are obtained. The topology exploration problem is formulated as a mixed-integer nonlinearly constrained programming (MINLP) problem. This is solved through a standard NLP solver which simultaneously generates and selects a topology optimized with respect to three performance metrics - hardware complexity, sensitivity under parameter variation and power consumption. The procedure performs this optimization process for all combination of integrator types and a set of local solutions are obtained. At the last stage, these are then checked for minimum signal path, sensitivity through Monte Carlo analysis and power consumption, which finally yields a global solution. This basic procedure is extended to develop a systematic methodology for designing reconfigurable continuous time $\Sigma\Delta$ modulator topologies in [39, 69]. This category of works follows the 'selection during sizing' principle discussed in the section 2.4.2.

4.3 Top-Down Generation of an Optimal Topology for ΣΔ Modulator

In this section, we illustrate the top-down generation methodology using ΣΔ modulator system as a case study. The individual steps of the methodology are first discussed in detail one by one in each subsection. The complete flow of the methodology for ΣΔ modulator system is described in the last subsection.

4.3.1 State Space Representation of Continuous Time ΣΔ Modulator Topology

The block diagram shown in Fig. 4.2 describes in general all single-bit modulators [71]. It is splitted into a linear block (the loop filter) and a nonlinear block (the quantizer). The linear block has arbitrary feedforward and feedback transfer functions $L_0(s)$ and $L_1(s)$ from its two inputs $u(t)$ and $y_D(t)$ respectively. For a single-bit modulator system, the loop filter is a two input, one output linear system. The loop filter transfer functions describe the input-output behavior of the system in frequency domain. The general form of a transfer function is given by

$$L(s) = \frac{Y(s)}{U(s)} = K\frac{b_m s^m + b_{m-1}s^{m-1} + ... + b_1 s + b_0}{a_n s^n + a_{n-1}s^{n-1} + ... + a_1 s + a_0}, \quad n \geq m \tag{4.1}$$

A transfer function however, does not provide any information concerning the physical structure of the loop filter. In the time domain, these are equivalently described by the state space model defined as [1]

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \tag{4.2}$$

$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t) \tag{4.3}$$

where $\mathbf{u}(t) \in \Re^p$ are input signals, $\mathbf{y}(t) \in \Re^q$ are the output signals, $\mathbf{x}(t) \in \Re^n$ are the state vectors, $\mathbf{A}(t) \in \Re^{n\times n}, \mathbf{B}(t) \in \Re^{n\times p}, \mathbf{C}(t) \in \Re^{q\times n}$ and $\mathbf{D}(t) \in \Re^{q\times p}$ are the state space matrices. Equation (4.2) models the internal description of the system and (4.3) models the system output in terms of the state vectors and the inputs. The

[1] In this chapter, bold lower hand alphabets represent vectors and bold upper hand alphabets represent matrix, e.g., **a** represents a vector and **A** represents a matrix.

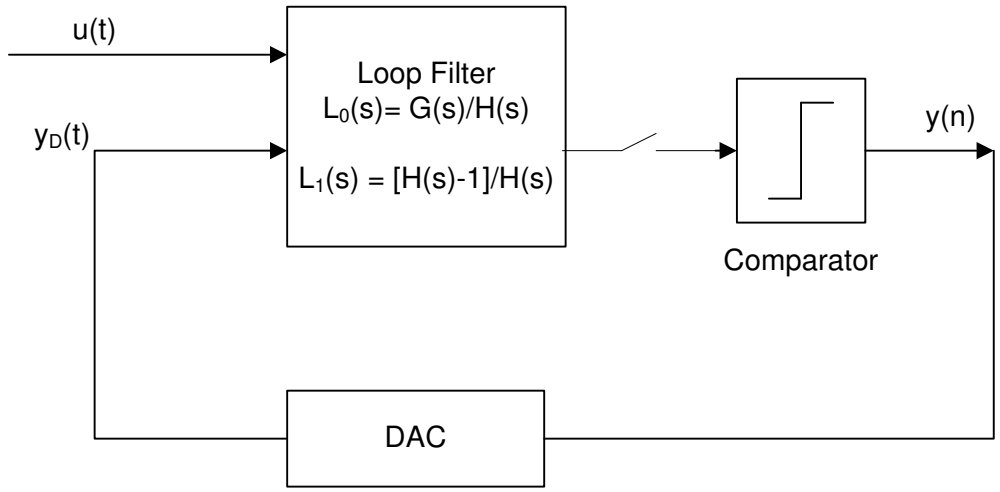


Figure 4.2: Block diagram of a CT $\Sigma\Delta$ modulator

transfer functions and state space equations are the two equivalent ways of modeling a continuous time system. The relationship between the two representations is given as

$$\mathbf{L}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \tag{4.4}$$

where $\mathbf{L}(s)$ is $q \times p$ transfer function matrix between $\mathbf{u}(t)$ and $\mathbf{y}(t)$. In the study of linear systems an important equation, referred to as characteristic equation plays a major role in determing the behavioral properties. From (4.4), this is defined as

$$\mathbf{E}(\mathbf{s}) = |s\mathbf{I} - \mathbf{A}| = 0 \tag{4.5}$$

The roots of the characteristic equation are often referred to as the eigen values of the matrix $\mathbf{A}$. These correspond to the poles of the loop filter.

Many popular $\Sigma\Delta$ modulator topologies are well represented by state space models [71]. Cascaded $\Sigma\Delta$ modulators can also be modeled in the same way by treating each section in the above way [72]. However, in practical cases, there are some designs which cannot be modeled through state space equations [73]. Automated synthesis of these topologies however, do not come under the scope of the present methodology.

4.3.2 Functional Topology Generation

Once the state space model for a modulator is known, a functional topology of the modulator is generated using an analog computation model [64]. A functional topology consists of only functional component blocks like adders, integrators etc. and is independent of any implementation style. The generation procedure is based upon the following two principle [64]:

1. The state space model consists of a set of differential algebraic equations. The functional topology implements these equations through three blocks - adders, integrators and scalars.

2. For an n^{th} order system with p inputs and q outputs, n integrators are required to realize the internal states, a set of $(n+p)$-input weighted adders are required to implement a state x_i and another set of $(n+p)$-input weighted adders are required to implement an output y_i.

The functional topology of an n^{th} order modulator with two inputs and single output, constructed on the basis of these principle is shown in Fig. 4.3. In the figure, the integrators are represented by $1/s$ blocks, $x_1, x_2, ..., x_n$ are the state variables, $a_{11}, ..., a_{1n}, b_{11}, ..., b_{n1}, c_1, ..., c_n, d_1, d_2$ are the signal-path coefficients of the modulator and corresponds to the state space matrix elements. These are realized by scalars. The nodes where the scaled state variables are combined represent addition, realized by adders. It is to be noted that the outputs of the integrators define the state variables.

As a practical example, let us consider the following state space model

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 1 & -f_1 \\ 0 & -f_2 \\ 0 & -f_3 \\ 0 & -f_4 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 0 & b_2 & b_3 & 0 \end{bmatrix} \quad \mathbf{D} = \begin{bmatrix} 0 & 0 \end{bmatrix} \tag{4.6}$$

The functional topology corresponding to this state space model is shown in Fig. 4.4 which is a widely used practical topology (chain of integrator with distributed feedback and distributed feedforward inputs).

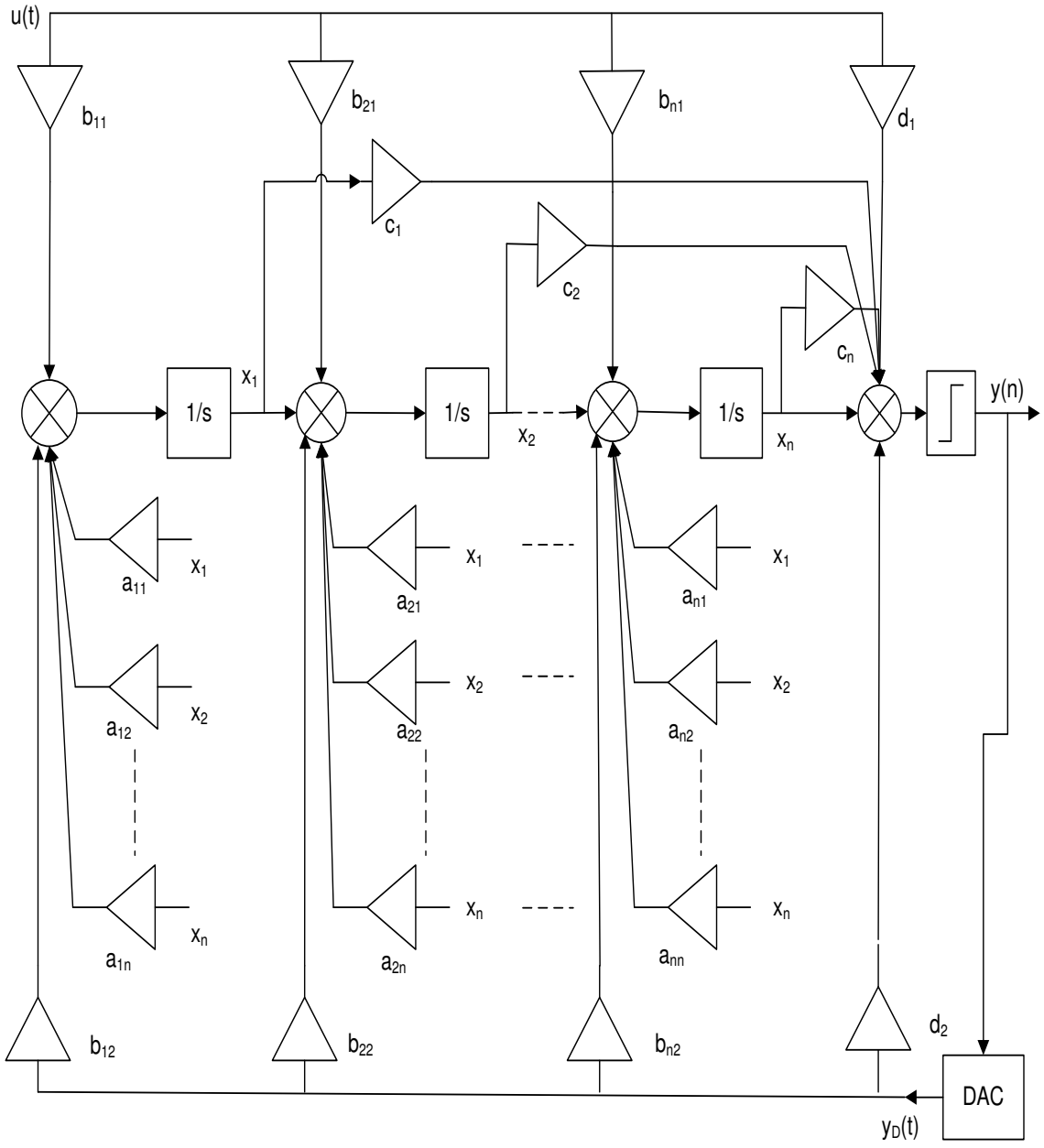


Figure 4.3: Functional topology for a general n^{th} order CT $\Sigma\Delta$ modulator

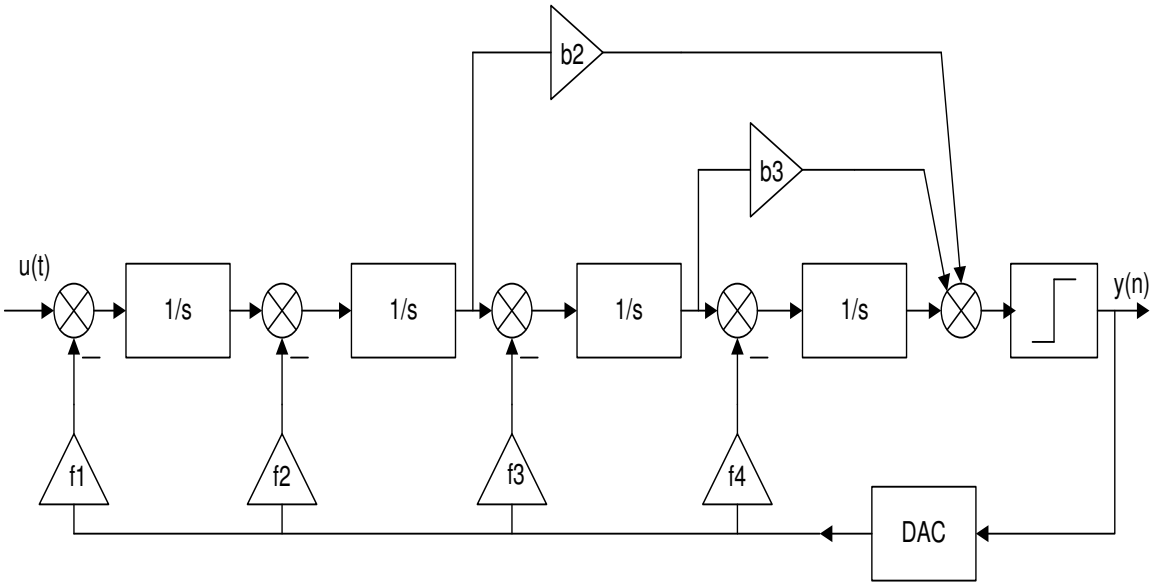


Figure 4.4: Functional topology of a 4^{th} order CT ΣΔ modulator.

4.3.3 Component-Level Topology Generation

A component-level topology is synthesized by replacing each component block of the functional topology with implementation style specific realization. The commonly used implementation styles are operational transconductance amplifier(OTA)-capacitor, active RC and switched capacitor. In the present work, OTA-C implementation style [74] has been chosen. In this style, the integrators are implemented by adding capacitors to the output of OTAs. Adders are simple nodes, where current addition takes place. The scalars are implemented by simple OTAs, converting input voltage signal to output current signal [74]. The mapping of the matrix elements to the transconductance (Gm) values of the OTAs are given by the following relationships [75].

$$|a_{ij}| = \frac{Gma_{ij}}{C_I}, |b_{ij}| = \frac{Gmb_{ij}}{C_I}, |c_i| = \frac{Gmc_i}{Gm_0}, |d_i| = \frac{Gmd_i}{Gm_0} \tag{4.7}$$

where C_I are the integrating capacitors and a_{ij}, b_{ij}, c_i, d_i are the elements of the state space matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ respectively. In terms of OTA and C, the generic 3^{rd} order CT modulator single-loop topology is shown in Fig. 4.5. For behavioral simulation purpose, the OTAs are replaced with appropriate behavioral models. The behavioral models as described in [39, 38] are considered for behavioral simulation purpose. The incorporated nonidealities are listed in Table 4.1.

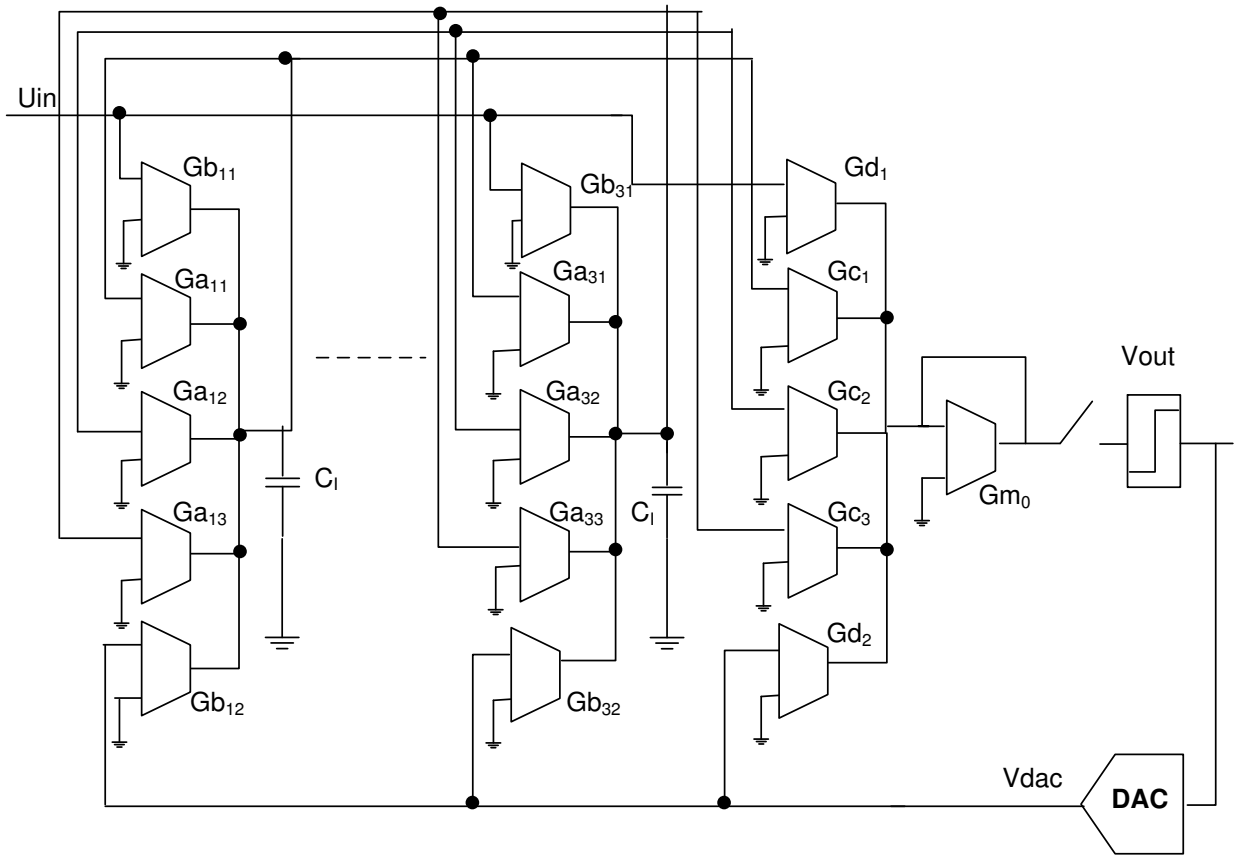


Figure 4.5: A generic 3^{rd} order CT modulator with $Gm - C$ loop-filter.

4.3.4 Topology Exploration

The topology exploration process is implemented at the state space matrix level. The process of generation of a new state space model from a given one physically corresponds to generation of a new topology from a given topology. In this subsection we discuss the theory behind the multiple state space generation process.

The set of all possible state vectors $\mathbf{x}(t)$ forms a finite dimensional real vector space, denoted by $\mathbf{V}(\mathcal{F})$ over the field $\mathcal{F}$ in a generic sense. A non singular matrix $\mathbf{T}$ which changes one state vector to another is defined as follows:

$$\bar{\mathbf{x}}(t) = \mathbf{T}^{-1}\mathbf{x}(t) \tag{4.8}$$

where $\bar{\mathbf{x}}(t)$ is the new state vector, $\mathbf{x}(t)$ is the old state vector and $\mathbf{T}$ is called as similarity transformation matrix in linear algebra [76]. Using this transformation, it is possible to generate a new state space model $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathbf{D}})$ from the old model

Table 4.1: Nonidealities of the OTA-C component blocks

Blocks	Nonidealities
Integrator	Finite and nonlinear gain, finite bandwidth, slew rate, finite output swing, linear input range, offset, thermal noise
Comparator	Offset, hysteresis
DAC	Jitter, Excess loop delay

$(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$. We have from (4.2) and (4.3) by direct substitution, the following relationship between the original and transformed state space matrices,

$$\bar{\mathbf{A}} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T} \tag{4.9}$$

$$\bar{\mathbf{B}} = \mathbf{T}^{-1}\mathbf{B} \tag{4.10}$$

$$\bar{\mathbf{C}} = \mathbf{C}\mathbf{T} \tag{4.11}$$

$$\bar{\mathbf{D}} = \mathbf{D} \tag{4.12}$$

Since a state space model acts as a basis of topology realization, the process of generation of a new state space model from an old one can be identified as generation of a new topology from an old one. The vector space $\mathbf{V}(\mathcal{F})$ is identified as topology space. The similarity transformation matrix is used as a topology transformation operator in the present work. Since there exists infinite number of non singular matrices, infinite number of state space models can be generated from a given one. Accordingly infinite number of topologies can be generated from a given one.

As an example of the multiple state space generation process, consider a 3^{rd} order ΣΔ modulator topology described the following state space model:

$$\bar{\mathbf{A}} = \begin{bmatrix} 0 & 0 & 0 \\ 0.3125 & 0 & -0.02313 \\ 0 & 0.0625 & 0 \end{bmatrix} \quad \bar{\mathbf{B}} = \begin{bmatrix} 5.643 & -7.525 \\ 0.08819 & -1.325 \\ 0 & -0.2339 \end{bmatrix} \tag{4.13}$$

$$\bar{\mathbf{C}} = \begin{bmatrix} 0 & 0 & 4 \end{bmatrix} \qquad \bar{\mathbf{D}} = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

Let the similarity transformation matrix be

$$\mathbf{T} = \begin{bmatrix} -0.9635 & -3.8779 & -2.9476 \\ -2.4620 & 2.1874 & -0.6847 \\ 1.8864 & -1.1728 & -1.6800 \end{bmatrix} \tag{4.14}$$

With this, the newly generated topology is described by the following state space model

$$\bar{\mathbf{A}} = \begin{bmatrix} -0.0405 & 0.0528 & -0.007813 \\ -0.05323 & 0.007813 & -0.02291 \\ 0.08328 & -0.02754 & 0.03269 \end{bmatrix} \quad \bar{\mathbf{B}} = \begin{bmatrix} -0.8802 & 1.228 \\ -1.034 & 1.027 \\ -0.2666 & 0.801 \end{bmatrix} \tag{4.15}$$

$$\bar{\mathbf{C}} = \begin{bmatrix} 7.546 & -4.691 & -6.72 \end{bmatrix} \qquad \bar{\mathbf{D}} = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

Under similarity transformation operation, the values of the state space matrix elements change. This leads to different performances of the corresponding topologies. However, behavioral properties of the topologies remain same as governed by the following invariant properties of similarity transformation.

Invariant Property. *The characteristic equation, eigenvalues, eigenvectors and transfer functions are invariant under the similarity transformations.*

Proof The characteristic equation is written as

$$\left|s\mathbf{I} - \bar{\mathbf{A}}\right| = \left|s\mathbf{I} - \mathbf{T}^{-1}\mathbf{A}\mathbf{T}\right| = \left|s\mathbf{T}^{-1}\mathbf{T} - \mathbf{T}^{-1}\mathbf{A}\mathbf{T}\right| \tag{4.16}$$

Since the determinant of a product matrix is equal to the product of the determinants of the matrices, (4.16) becomes

$$\left|s\mathbf{I} - \bar{\mathbf{A}}\right| = \left|\mathbf{T}^{-1}\right| |s\mathbf{I} - \mathbf{A}| \, |\mathbf{T}| = |s\mathbf{I} - \mathbf{A}| \tag{4.17}$$

With this transformation, the transformed transfer function matrix is defined as

$$\bar{\mathbf{L}}(s) = \bar{\mathbf{C}}(s\mathbf{I} - \bar{\mathbf{A}})\bar{\mathbf{B}} + \bar{\mathbf{D}} \tag{4.18}$$

$$= \mathbf{C}\mathbf{T}(s\mathbf{I} - \mathbf{T}^{-1}\mathbf{A}\mathbf{T})\mathbf{T}^{-1}\mathbf{B} + \mathbf{D} \tag{4.19}$$

which is simplified to

$$\bar{\mathbf{L}}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A}^{-1})\mathbf{B} + \mathbf{D} = \mathbf{L}(s) \tag{4.20}$$

The invariance of behavioral properties of the topologies under ideal conditions is ensured from the above discussion. Under nonideal conditions the invariance property is verified through behavioral simulation as follows. We consider the 3^{rd} order topologies given by (4.13) and (4.15). These are behaviorally simulated considering all the nonidealities listed in Table 4.1. The *SNR* of the modulators are plotted as functions of the normalized input signal amplitude. These are shown in Fig. 4.6(a). We observe that both the curves closely follow each other. The dynamic range of both the topologies under non ideal conditions are almost equal with difference less than 1%. We repeat the same experiment for a 4^{th} order modulator. In this case also, we observe that the *DR* are nearly same for the two modulators. The *SNR* curves for this modulator are shown in Fig. 4.6(b). Therefore the invariant property of similarity transformation holds reasonably good for topologies operating under non-ideal conditions even.

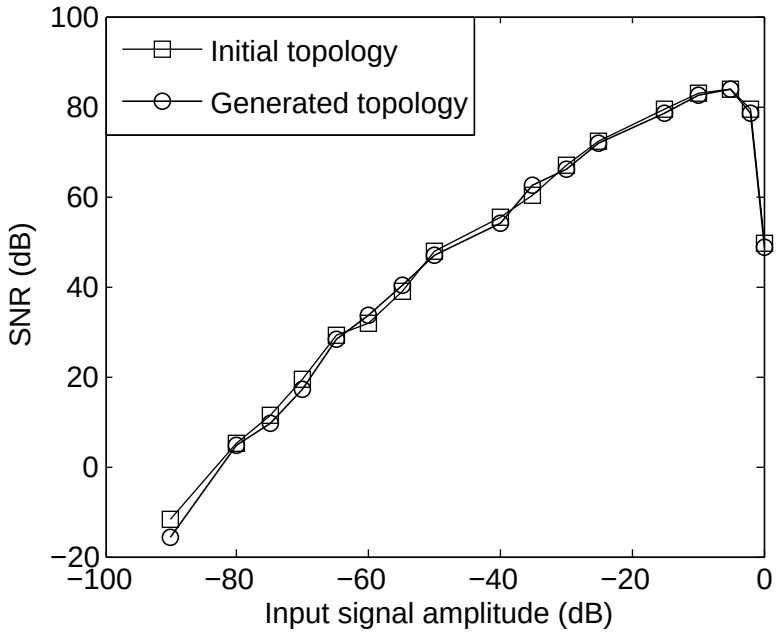
4.3.5 Performance Estimation

Since during a high-level topology generation process, detailed knowledge about the circuit-level implementation of the component blocks of a topology is not known, true estimation of real performances of a topology is difficult. As a result, the topologies are compared on the basis of some heuristic measures of the performances.[2] In this work, three types of performance parameters are chosen for comparison purpose : (1) sensitivity of the modulator response to the variation of the modulator coefficients, (2) hardware complexity and (3) power consumption. The process of estimation for these are discussed below.

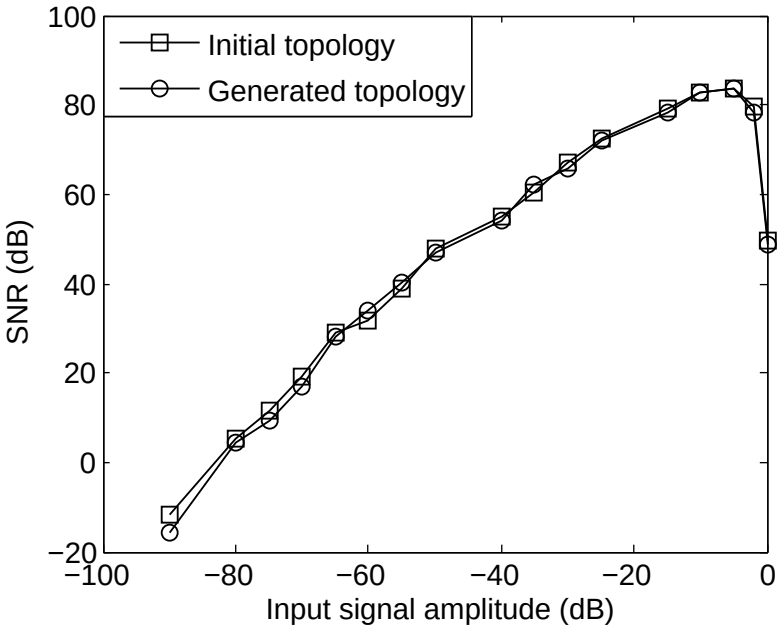
4.3.5.1 Sensitivity minimization

In practice it is not possible to realize the coefficients of the state space matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ exactly at the circuit level. For a topology $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ with transfer function $\mathbf{L}(s)$, we define the sensitivity of $\mathbf{L}(s)$ with respect to the elements of the

[2]In chapter 2 of the dissertation, we mentioned the top-down approach for generation of high-level performance estimation models. In this chapter the estimation models are constructed using this approach.



(a) SNR plot for 3^{rd} order modulator.



(b) SNR plot for 4^{th} order modulator.

Figure 4.6: SNR plot for two topologies generated through similarity transformation, establishing invariant property.

matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\mathbf{D}$ to be the partial derivative of $\mathbf{L}(s)$ with respect to these elements. Considering all the matrix elements in a compact way, we define the sensitivities $S_{\mathbf{A}}(s)$, $S_{\mathbf{B}}(s)$, $S_{\mathbf{C}}(s)$ and $S_{\mathbf{D}}(s)$ of $S_{\mathbf{L}}(s)$ to the coefficients of $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ as

$$S_{\mathbf{A}}(s) \triangleq \frac{\partial \mathbf{L}}{\partial \mathbf{A}} = \mathbf{G}(s)\mathbf{F}^T(s) \tag{4.21}$$

$$S_{\mathbf{B}}(s) \triangleq \frac{\partial \mathbf{L}}{\partial \mathbf{B}} = \mathbf{G}(s) \tag{4.22}$$

$$S_{\mathbf{C}}(s) \triangleq \frac{\partial \mathbf{L}}{\partial \mathbf{C}^T} = \mathbf{F}(s) \tag{4.23}$$

$$S_{\mathbf{A}}(s) \triangleq \frac{\partial \mathbf{L}}{\partial \mathbf{D}} = 1 \tag{4.24}$$

where $\mathbf{F}(s)$ and $\mathbf{G}(s)$ are intermediate transfer functions. $\mathbf{F}(s)$ is the intermediate transfer function from input of the filter to the output of the integrators and $\mathbf{G}(s)$ is the intermediate transfer function from the input of the integrators to the output of the filter. These are defined as

$$\mathbf{F}(s) = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} \tag{4.25}$$

$$\mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1} \tag{4.26}$$

We note that the above sensitivity functions are matrix functions of the complex variable 's'. A further definition is required to be able to measure the collective effects, averaged over all frequencies, of the elements of $S_{\mathbf{A}}(s)$, $S_{\mathbf{B}}(s)$ and $S_{\mathbf{C}}(s)$ respectively. The overall sensitivity measure of the transfer function $\mathbf{L}(s)$ w.r.t the state space matrices is defined in [77] as follows

$$S_{L12} = \left|\left|\frac{\partial \mathbf{L}}{\partial \mathbf{A}}\right|\right|_1^2 + \left|\left|\frac{\partial \mathbf{L}}{\partial \mathbf{B}}\right|\right|_2^2 + \left|\left|\frac{\partial \mathbf{L}}{\partial \mathbf{C}^T}\right|\right|_2^2 \tag{4.27}$$

where $||\mathbf{f}(s)||_p$ denotes L_p norm of the complex matrix function $\mathbf{f}(s)$. This is defined as

$$||\mathbf{f}(j\omega)|| = \left(\frac{1}{2\pi}\int_0^{2\pi} ||\mathbf{f}(j\omega)||_F^p \, d\omega\right)^{\frac{1}{p}} \tag{4.28}$$

where $||\mathbf{f}(j\omega)||_F$ is the Frobenius norm of the matrix $\mathbf{f}(j\omega)$, defined as

$$||\mathbf{f}(j\omega)||_F = \left\{tr\left[\mathbf{f}^T(-j\omega)\mathbf{f}(j\omega)\right]\right\}^{\frac{1}{2}} \tag{4.29}$$

In this equation $tr(\mathbf{Y})$ means trace of the matrix $\mathbf{Y}$. Considering (4.7), S_{L12} can be thought of as the measure of the sensitivity of the loop-filter output to variation of OTA transconductances. It is to be noted that in 4.27, an L_1 norm is used for the sensitivity function of $\mathbf{L}(s)$ w.r.t $\mathbf{A}$ and an L_2 norm for the other two sensitivity functions. This explains the use of the notation S_{L12} for the sensitivity measure. This is referred to as L_1/L_2 sensitivity measure. This definition has become widely acceptable because of its computational simplicity.

Direct evaluation of the term $\left|\left|\frac{\partial \mathbf{L}}{\partial \mathbf{A}}\right|\right|_1^2$ is quite difficult rendering the estimation process computationally expensive. But the process becomes manageable if S_{L12} is replaced by an upper bound containing only L_2 norms [78]. In addition, it was shown [78] that the solution that minimizes the upper bound also happends to minimize the measure S_{L1L2} itself. Using Cauchy-Schwartz inequality the first term in the right hand side of (4.27) is written as

$$\left|\left|\frac{\partial \mathbf{L}}{\partial \mathbf{A}}\right|\right|_1^2 \leq \left|\left|\frac{\partial \mathbf{L}}{\partial \mathbf{B}}\right|\right|_2^2 \left|\left|\frac{\partial \mathbf{L}}{\partial \mathbf{C}^T}\right|\right|_2^2 \tag{4.30}$$

With these the defining equation for the sensitivity becomes

$$\begin{aligned} \tilde{S}_{L12} &= \left|\left|\frac{\partial \mathbf{L}}{\partial \mathbf{B}}\right|\right|_2^2 \left|\left|\frac{\partial \mathbf{L}}{\partial \mathbf{C}^T}\right|\right|_2^2 + \left|\left|\frac{\partial \mathbf{L}}{\partial \mathbf{B}}\right|\right|_2^2 + \left|\left|\frac{\partial \mathbf{L}}{\partial \mathbf{C}^T}\right|\right|_2^2 & (4.31) \\ &= ||\mathbf{G}(s)||_2^2 \, ||\mathbf{F}(s)||_2^2 + ||\mathbf{G}(s)||_2^2 + ||\mathbf{F}(s)||_2^2 & (4.32) \end{aligned}$$

We note that this upper bound now contains only L_2 norms. This allows us to rewrite $\tilde{S}_{L12}$ in terms of the controllability and observability Gramians of the state space model. The observability Gramian matrix $\mathbf{W}$ and controllability Gramian matrix $\mathbf{K}$ are defined as

$$\begin{aligned} \mathbf{W} &= \frac{1}{2\pi} \int_{-\infty}^{+\infty} \mathbf{G}^* \mathbf{G} d\omega & (4.33) \\ \mathbf{K} &= \frac{1}{2\pi} \int_{-\infty}^{+\infty} \mathbf{F}\mathbf{F}^* d\omega & (4.34) \end{aligned}$$

The asterisk (*) denotes the adjoint operator, so if $\mathbf{Y}$ is an arbitrary matrix, $\mathbf{Y}^*$ is the transpose of the matrix of complex conjugates of the elements of $\mathbf{Y}$. These two Gramian matrices are related to the state space matrices through Lyapunov

equations, defined as

$$\mathbf{A}^T\mathbf{W} + \mathbf{W}\mathbf{A} = -\mathbf{C}^T\mathbf{C} \quad (4.35)$$

$$\mathbf{A}\mathbf{K} + \mathbf{K}\mathbf{A}^T = -\mathbf{B}\mathbf{B}^T \quad (4.36)$$

Using these Lyapunov equation, relationship between the intermediate transfer functions and the Gramian matrices are deduced through simple linear algebraic manipulations. These are given as

$$||\mathbf{G}(s)||_2^2 = tr(\mathbf{W}) \quad (4.37)$$

$$||\mathbf{F}(s)||_2^2 = tr(\mathbf{K}) \quad (4.38)$$

Thus we arrive at the final expression for the L_1/L_2 norm of the sensitivity as

$$\tilde{S}_{L12} = tr(\mathbf{W})tr(\mathbf{K}) + tr(\mathbf{W}) + tr(\mathbf{K}) \quad (4.39)$$

With state space exploration by similarity transformation, the Gramian matrices are changed from $(\mathbf{W}, \mathbf{K})$ to $\left(\mathbf{T}^T\mathbf{W}\mathbf{T}, \mathbf{T}^{-1}\mathbf{K}\mathbf{T}^{-T}\right)$. With this, $\tilde{S}_{L12}$ changes to

$$\tilde{S}_{L12}(\mathbf{T}) = tr\left(\mathbf{T}^T\mathbf{W}\mathbf{T}\right) tr\left(\mathbf{T}^{-1}\mathbf{K}\mathbf{T}^{-T}\right) + tr\left(\mathbf{T}^T\mathbf{W}\mathbf{T}\right) + tr\left(\mathbf{T}^{-1}\mathbf{K}\mathbf{T}^{-T}\right) + 1 \quad (4.40)$$

Thus the L_1/L_2 norm sensitivity for various topologies can be estimated by evaluating the expression in (4.40). This is easy to evaluate using Matlab control system toolbox.

4.3.5.2 Hardware Complexity

Hardware complexity is measured by the number of OTAs required to implement a modulator at the circuit-level. This is minimized by minimizing the number of signal paths in the loop filter which means reduction in the number of OTAs. The following cost function is used as the measure of hardware complexity

$$X = \sum_{k=1}^{(N+1)(N+2)} h_k \quad (4.41)$$

N is the order of the filter and (N+1)(N+2) represents the total number of elements in the matrix quadruple. Let m_k represent the elements of the matrix quadruple

$(\mathbf{T}^{-1}\mathbf{AT}, \mathbf{T}^{-1}\mathbf{B}, \mathbf{CT}, \mathbf{D})$. Then h_k is defined as

$$h_k = \begin{cases} 0 & \text{if } m_k < \epsilon \\ 1 & \text{otherwise} \end{cases} \tag{4.42}$$

where ϵ represents very small value,

4.3.5.3 Relative Power Consumption

Without considering the circuit-level implementation details of the component-blocks it is very difficult to accurately estimate the real power consumption of a modulator. In addition, for topology exploration purpose it is not necessary to accurately estimate the power consumption [8]. The estimator needs to measure correctly the change in power consumption of the topology with change in the design parameter values of the topology [8]. Such an estimator is often referred to as a relative power estimator.

A commonly used assumption for constructing a power estimator for $\Sigma\Delta$ modulator is that the largest part of the power consumption of the modulator is determined by the OTAs [21, 36]. This accounts for the static power consumption. This is given as [79]

$$P \approx n_I I_b V_{DD} M \tag{4.43}$$

where n_I is the number of current branches in the OTA circuit-topology, I_b is the bias current for an OTA, V_{DD} is the power supply voltage and M is the total number of OTAs required to implement a topology. For many popular circuit topologies of an OTA, the bias current I_b is related to Gm value as $I_b = Gm^2/(2\beta)$ where $\beta = K'(W/L)$ is the transconductance parameter of the input transistor pairs of the OTA. K' is a process constant and W/L is the aspect ratio for the input MOS transistors. For exact estimation of power consumption, these parameters should be known along with n_I. However, for tracking purpose, the power consumption for a single OTA can be considered to be proportional to Gm^2. With this, the relative power estimator is given by

$$P \approx \sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}^2 C_I^2 + \sum_{i=1}^{N}\sum_{j=1}^{2} b_{ij}^2 C_I^2 + \sum_{i=1}^{N} c_i^2 Gm_0^2 + \sum_{i=1}^{N} d_i^2 Gm_0^2 \tag{4.44}$$

using (4.7) where C_I is the integrating capacitor and Gm_0 is a fixed OTA value,

which remain constant throughout the procedure.

4.3.6 Topology Selection

The topology selection process is formulated as a simulated annealing (SA)-based optimization process. The topology space $\mathbf{V}(\mathcal{F})$ is explored through an optimization algorithm which selects an optimal topology as a solution point. The exploration points are generated through nonsingular similarity transformation matrix $\mathbf{T}$. The elements of this matrix are restricted within a definite range of real numbers. The performance estimators constructed above are used to estimate the qualities of an exploration point. These thus serve as cost functions. Suitable constraints are added within the exploration procedure so that a feasible point is chosen as the final solution point. The details of the SA algorithm has been discussed in the appendix B.2 of the dissertation. Several termination criteria have been incorporated in the algorithm such as maximum iteration count, cost variance threshold and maximum number of consecutive times with no cost decrease.

For a given cost function Φ, the topology selection problem is formulated as

$$\begin{aligned} \text{Minimize} \quad & \Phi(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathbf{D}}) \\ \text{such that} \quad & g_1(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathbf{D}}) \leq 0,\ g_2(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathbf{D}}) \leq 0 \\ \text{and} \quad & det(\mathbf{T}) \neq 0 \end{aligned} \tag{4.45}$$

The cost function Φ is given as the sum of three individual cost functions (4.40),(4.41) and (4.44). $g_i(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathbf{D}})$ is a feasibility constraint. The following feasibility constraints are used in this work.

4.3.6.1 Feasible Gm Value Constraints

Since it is difficult to realize a too high Gm or too low Gm, constraints are added to ensure that for a topology, the required OTA values lie within a certain limit. The feasibility constraints for the state space coefficients are therefore explicitly written as

$$Gm_{min} \leq (a_{ij}C_I, b_{ij}C_I, c_iGm_0, d_iGm_0) \leq Gm_{max} \tag{4.46}$$

4.3.6.2 Non-overload Constraints

For a stable ΣΔ modulator, the integrator outputs must lie below the clipping level L_1 and the final filter output must lie below the corresponding clipping level L_2. Non-overload under normal operating conditions requires that the matrix elements must satisfy certain constraints. These are formulated as from (4.2) and (4.3) following the approach of [80].

$$\left|(j\omega\mathbf{I}-\bar{\mathbf{A}})^{-1}\bar{\mathbf{B}}\begin{bmatrix} U_{max} & V_{ref}\end{bmatrix}^T\right| \leq PL_1 \tag{4.47}$$

$$\left|L_1\bar{\mathbf{C}}+\bar{\mathbf{D}}\begin{bmatrix} U_{max} & V_{ref}\end{bmatrix}^T\right| \leq PL_2 \tag{4.48}$$

where U_{max} is the maximum signal amplitude, V_{ref} is the DAC reference voltage and ω is the frequency of the signal at the integrator output. P is a scaling factor, depending on modulator order and $P > 1$. The reasons to have a scaled version of L_1 and L_2 is to account for over estimation of worst case analysis [24]. A lower P is more likely to avoid overloading, but also more likely to over constrain the coefficient variables. The exact value for P is fixed through few iterations of the optimization process. Exact values of L_1 and L_2 can only be extracted from circuit simulation results. Some heuristic values are taken in this work.

4.3.7 Complete Flow

A complete flow of the topology generation process is shown in Fig. 4.7. The input to the process is a transfer function model of the system. This includes the feedforward and feedback transfer functions of the loop filter. The feasibility constraints are initialized next. The non-overload scaling factor is heuristically initialized to half of the modulator order. From the given transfer function, a seed state space is generated. The performance estimation functions determine the sensitivity response, complexity as well as the relative power consumption of the state space model. A cost function is computed using the estimated performances and the SA algorithm selects a state space model for which the cost function is optimum. The selected state space is realized using OTA-C component blocks. The topology is then behaviorally simulated for a series of input amplitudes, considering the non-idealities listed in Table 4.1. The topology is then tested for two cases. First, the integrator outputs

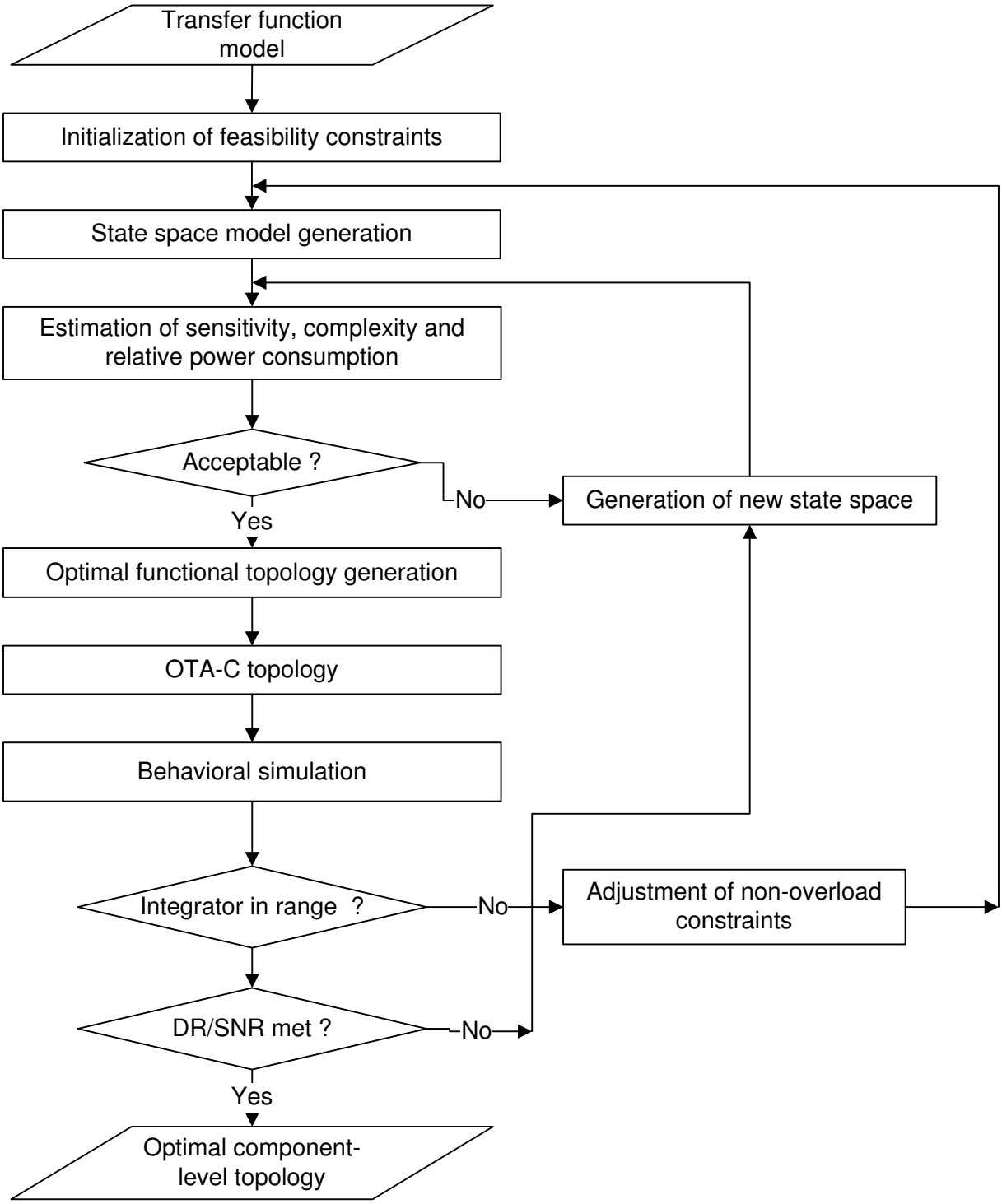


Figure 4.7: Complete flow of the topology generation process for ΣΔ modulator system.

are checked to see whether clipping occurs. If yes, the non-overload condition is adjusted by reducing the scale factor. If the generated topology does not overload, then the DR/SNR of the topology is estimated and compared with the desired value. If the desired specs are achieved, the current topology is taken as the output of the process, otherwise the entire process is repeated and a new topology is generated.

4.4 Comparison with Existing Methodologies

The works closely related with the present work have been described in [7, 25, 27]. In [24], a generic template-based methodology has been described for generation of an optimal topology for discrete time $\Sigma\Delta$ modulator system. We present a comparison between our methodology and the existing methodologies below.

1. The component-level topology generation methodology in [7, 25] starts from an HDL-based functional description of the system to be designed. This is a low level description and is customized to a specific system topology. Such description is then converted to an SFG, from which component-level topologies are generated through mapping process or heuristic conversion rules. These methodologies therefore require an apriori knowledge of the system topology. On the other hand, our methodology generates component-level topologies from a transfer function model. Such model is often easy to compute from the desired specifications of a system through well established techniques and is not specific to any particular topology of the system. Thus our methodology generates component-level topologies from a much higher level of abstraction compared to [7, 25] and do not require any knowledge of the system topology.

2. The ARCHGEN methodology as described in [27] starts from a transfer function model and generates component-level topologies from it via state space models. In spite of its novelty, this methodology fails to generate an optimal topology. This is because the methodology does not include any performance optimization step within it. In addition, circuit-level non-idealities have not been considered to verify the generated topology. Our methodology extends the ARCHGEN methodology by including a performance estimation and optimization procedure. In addition, a behavioral simulation process is included in our methodology to check the performances of the generated topology under circuit-level non-idealities.

3. The major difference between our methodology and that in [24] is that our methodology follows a top-down approach in contrast to template-based approach in [24]. The manual derivation of NTF and STF of the generic topology as required in [24] is not an easy task. The number of symbolic terms grows roughly with the complexity of $4 \times N!$, where N is the modulator order. In addition these are to be recalculated if the modulator order changes. The topologies are generated after solving a set of manually formulated symbolic equations through MINLP programming. On the other hand, in our methodology, the topologies are generated directly from the desired modulator specifications. Thus our topology generation process is free of any manual labour. A fully automated implementation our methodology is developed under Matlab environment, so that it can be used by designers without expert design knowledge. Thus our methodology is advantageous over [24] from the view point of design automation.

Thus the advantage with our methodology is that the designer is able to specify the design goal and desired specifications at a higher level of abstraction in terms of transfer functions, while the design tool is able to generate an optimal component-level topology directly from transfer functions in a highly automated manner. In addition, the generated topology is ensured to work satisfactorily under circuit-level non-ideal conditions.

4.5 Experimental Results

In this section, we provide experimental results demonstrating the methodology described above. The entire methodology has been implemented in Matlab.

4.5.1 Experiment 1

We consider a 3^{rd} order modulator. The design specifications are: (i) dynamic range DR = 85 dB (ideally). (ii) Maximum input signal bandwidth = 100 KHz. (iii) DAC reference voltage (V_{ref}) = 200 mV. The feedforward path and feedback

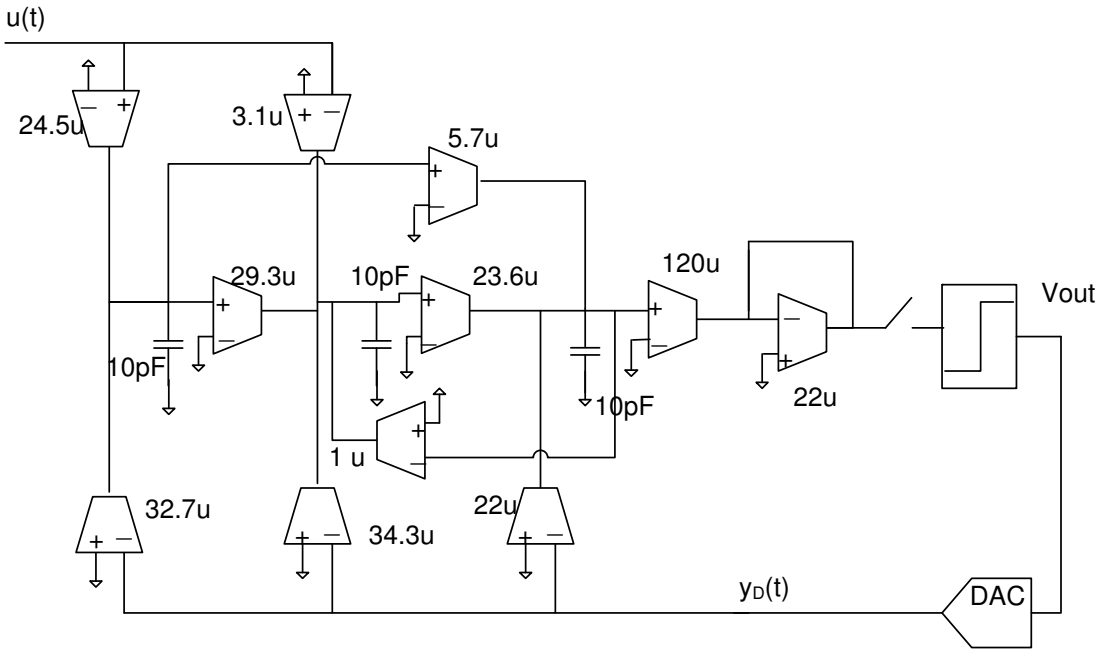


Figure 4.8: Optimized 3^{rd} order topology.

path normalized transfer functions of the loop-filter are

$$L_0(s) = \frac{0.0221\,(s+2)}{s\,(s^2+0.001446)} \tag{4.49}$$

$$L_1(s) = \frac{42.441\,(s^2+0.3541s+0.0628)}{(s+2)} \tag{4.50}$$

These serve as inputs to the process. From this, the seed state space is calculated and is given below.

$$\mathbf{A}_{seed} = \begin{bmatrix} 0 & 0 & 0 \\ 0.3125 & 0 & -0.02313 \\ 0 & 0.0625 & 0 \end{bmatrix} \quad \mathbf{B}_{seed} = \begin{bmatrix} 5.643 & -7.525 \\ 0.08819 & -1.325 \\ 0 & -0.2339 \end{bmatrix} \tag{4.51}$$

$$\mathbf{C}_{seed} = \begin{bmatrix} 0 & 0 & 4 \end{bmatrix} \qquad \mathbf{D}_{seed} = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

Using this state space as the seed state space, an SA-based optimization process is executed to determine an optimal state space which obeys the feasibility constraints. The feasible Gm range is taken as $1\mu S - 530\mu S$. For non-overload constraint computation, the scale factor P is initialized to half of the modulator order,

i.e., $P = \frac{3}{2}$. We assume that $U_{max} = V_{ref}$ and L_1, L_2 are considered as 600 mV each. The frequency at the integrator node is heuristically chosen to be one-fifth of the signal bandwidth. This allows to derive a set of non-overload constraints for the modulator. Behavioral simulation is used to check the integrator clipping. The scaling factor is decreased by 0.2 in each iteration. The topology selection process takes 492 seconds on a PIV, 512 MB PC to determine an optimal topology after 3 iterations. In the process, 5621 topologies have been explored. The state space model corresponding to the final selected topology is given below

$$\mathbf{A}_{opt} = \begin{bmatrix} 0 & 0 & 0 \\ 29.26 & 0 & -1 \\ 5.69 & 23.65 & 0 \end{bmatrix} \times 10^5 \quad \mathbf{B}_{opt} = \begin{bmatrix} 24.52 & -32.69 \\ -3.09 & -34.26 \\ 0 & -21.98 \end{bmatrix} \times 10^5 \tag{4.52}$$

$$\mathbf{C}_{opt} = \begin{bmatrix} 0 & 0 & 5.45 \end{bmatrix} \qquad \mathbf{D}_{opt} = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

For the OTA-C integrator we take the load capacitor values of all the integrators to be $C_I = 10$ pF. Now the Gm values of the OTAs are calculated from (4.7). We assume $Gm_0 = 22\mu S$. The generated OTA-C topology is shown in Fig. 4.8. The OTA values are shown in the figure itself.

Table 4.2: Nonidealities considered

Blocks	Nonidealities	Magnitudes
OTAs	Output impedance	2 MΩ
	High-freq BW	30 MHz
	Low-freq BW	8 KHz
	Swing	500 mV
	Offset	0.1 mV
Comparator	Hysteresis	5 mV
	Offset	10 μ V
Others	Excess loop delay	10 ns
	Clock Jitter	1 ps

To characterize the generated topology, we perform behavioral simulation including the nonidealities as listed in Table 4.2. The FFT-based noise power spectral density is shown in Fig. 4.9. The variation of *SNR* against the input signal amplitude normalized to the maximum allowed, under ideal conditions as well as non-ideal

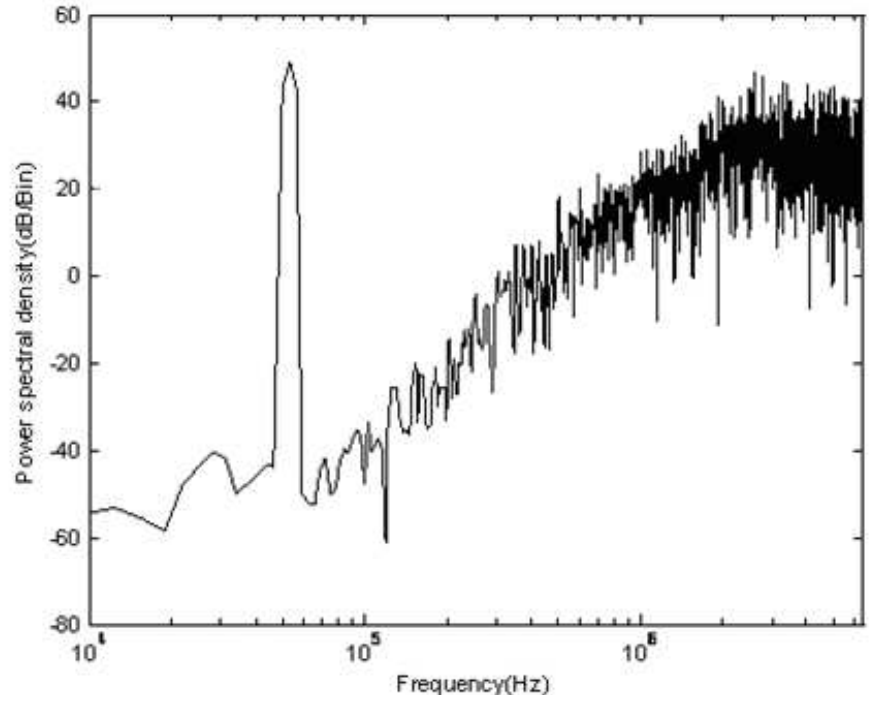


Figure 4.9: Comparison of SNR plot for the generated topology under ideal and non-ideal conditions.

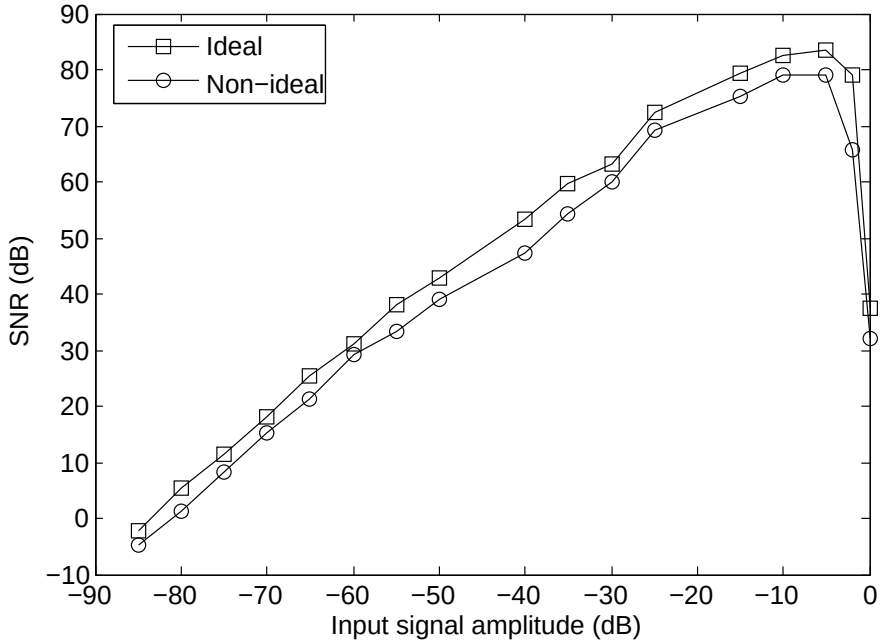


Figure 4.10: Comparison of SNR plot for generated topology under ideal and non-ideal conditions.

Table 4.3: Comparision of behavior under ideal and non-ideal conditions

Condition	DR	SNR peak
ideal	84.84 dB	84.556 dB
non-ideal	82.8 dB	81.3 dB

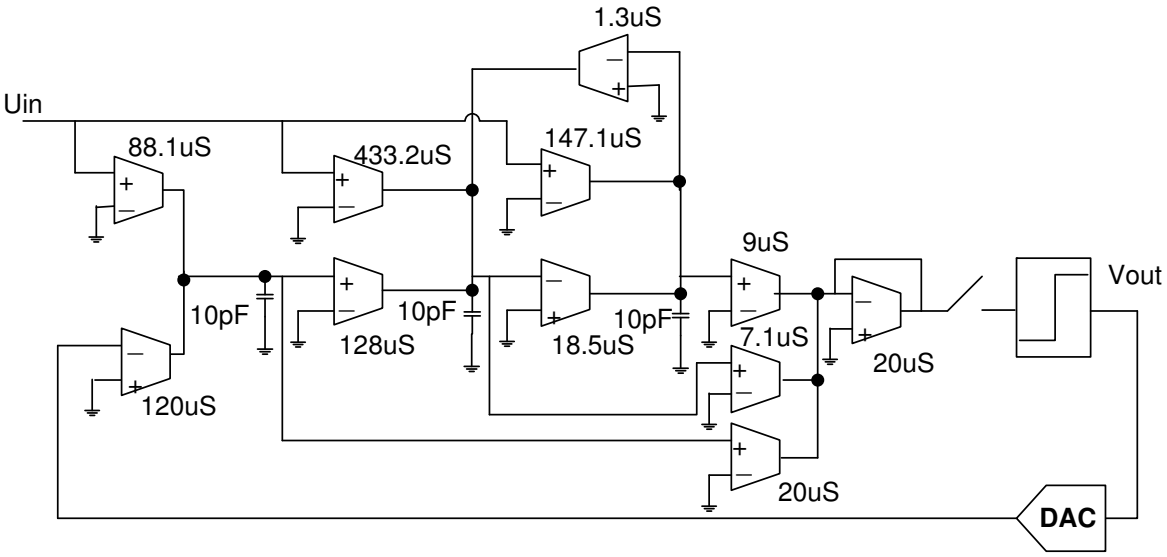


Figure 4.11: Cascaded chain feedforward topology for 3^{rd} order modulator.

conditions is shown in Fig. 4.10. From the figure, we make a comparison between the behavior of the generated topology under ideal and non-ideal conditions of simulation. These are listed in Table 4.3. Thus degradation due to considered nonidealities is approximately 2 dB.

4.5.2 Experiment 2

In order to verify the optimality of the selected topology, we compare the performances of the selected topology with two standard topologies, widely used in manual design process. Of them, one is referred to as cascaded integrator feedforward (CIFF) topology and the other one as distributed feedback (DF) topology [81]. These are shown in Fig. 4.11 and Fig. 4.12 respectively. The coefficients for these topologies are manually calculated using Matlab $\Sigma\Delta$ toolbox [66] and DT-CT technique as described in [85, 71] such that they satisfy the desired specifications and obey the feasibility constraints. A comparison between the SNR plot of the generated topology and the manually designed CIFF and DF topology is shown in Fig.

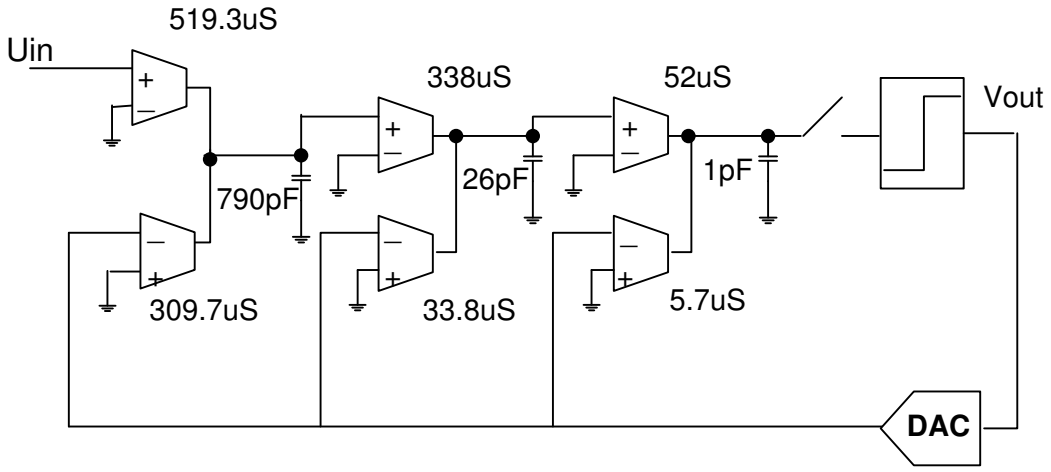


Figure 4.12: Distributed feedback topology for 3^{rd} order modulator.

Table 4.4: Comparison in terms of yield for coefficient variation

Variation	Optimized	CIFF	DF
$\pm 2\%$	1000	982	975
$\pm 5\%$	923	798	320
$\pm 10\%$	536	380	107

4.13. This result shows that all the three topologies satisfy the desired DR/SNR specifications.

To compare the sensitivity characteristics of the optimized topology, the CIFF topology and the DF topology, an easy way is to compare the values of the sensitivity metrics. These values normalized to sensitivity of the seed topology for the three topologies are 1.23 , 2.34 and 3.82 respectively. However, to have better insight in terms of SNR and DR , we run Monte Carlo analysis for the three topologies. We vary the OTA's Gm values by $\pm 2\%$, $\pm 5\%$ and $\pm 10\%$ from their nominal value assuming that the variation follows Gaussian distribution. We compare the change of the SNR value from the nominal value by applying input signal with same amplitude and frequency to all the three topologies. For each topology, we perform 1000 runs of Monte-Carlo simulation. For comparison purpose, we consider two performance metrics, similar to that in [24]. The first one is yield. This is equal to the total number of feasible samples among the 1000 runs. For the feasible samples, the fall of SNR from the nominal value is not greater than 5 dB. The second one is the deviation of SNR of the topologies from the nominal values for a fixed coefficient

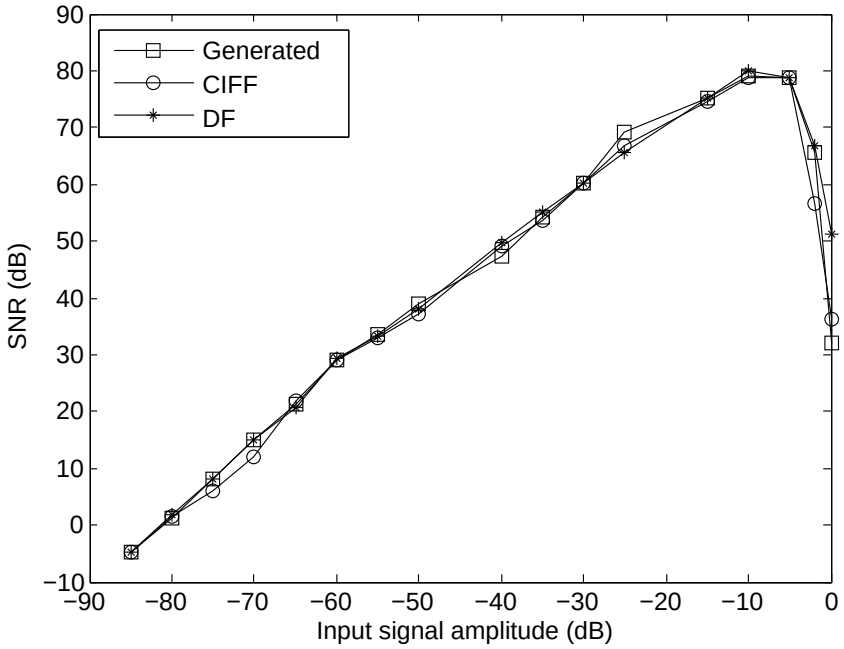


Figure 4.13: SNR/DR comparison between the generated, CIFF and DF topology.

variation. Table 4.4 summarizes the results of yield estimation for all the topologies. For 2 % coefficient variation, we observe that out of 1000 samples, all the samples for the optimized topology are feasible, whereas 982 and 975 samples are feasible for the CIFF and the DF topology respectively. Therefore, for 2 % variation, the performances are comparable. For larger variation, (5 % say) the optimized topology outperforms the other two. Of the three, the DF topology shows the worst performance. This is expected from the values of the sensitivity cost function. The distribution of *SNR* deviation from nominal value is shown in Fig. 4.14 and Fig. 4.15 for 2 % and 5 % coefficient variation respectively. From Fig. 4.14, we observe that *SNR* deviation for the optimized topology is concentrated within |2 dB|. On the other hand, for the other two topologies this may be more than |3 dB|. Similarly from Fig. 4.15, we observe that *SNR* deviation for the optimized topology is concentrated within |3 dB|. Thus we conclude that the optimized topology is more tolerant to coefficient variation not only in terms of yield but also performance deviations.

A comparison between the optimized, chosen and DF topologies w.r.t. relative power consumption and hardware complexity (number of OTAs used) is tabulated in Table 4.5. From this we observe that the distributed feedback topology has less number of hardware components compared to the other two. Despite of this, it has high relative power consumption compared to the other two. The optimized

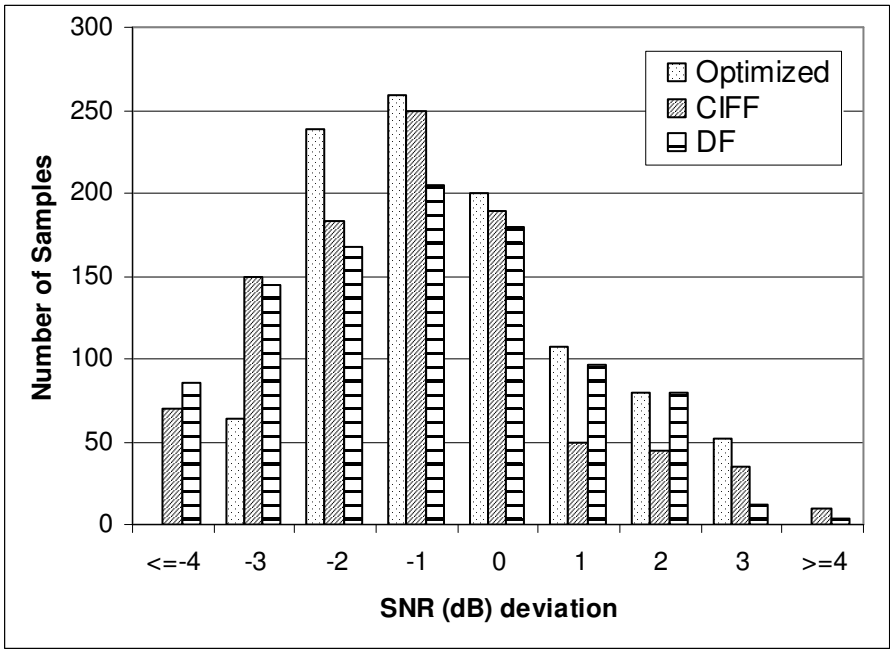


Figure 4.14: Bar diagram of peak SNR deviation for 2% coefficient variation.

Table 4.5: Comparison between the topologies for relative power and complexity

Metrics	Optimized	CIFF	DF
relative power	5.416	249	483
complexity	10	11	7

topology is superior than the other two in terms of relative power consumption.

4.5.3 Experiment 3

In this case, we consider a 4^{th} order CT $\Sigma\Delta$ modulator and repeat the same tasks as before. The design specifications are: (i) dynamic range = 96 dB. (ii) Maximum input signal bandwidth = 100 KHz. (iii) DAC reference voltage (V_{ref}) = 200 mV. The feedforward path and feedback path normalized transfer functions of the loop-filter are

$$L_0(s) = \frac{-0.00103\,(s^2-6)}{(s^2+2.78\times10^{-4})\,(s^2+1.79\times10^{-3})} \tag{4.53}$$

$$L_1(s) = \frac{-911.96\,(s+0.185)\,(s^2+0.1775s+0.04745)}{(s^2-6)} \tag{4.54}$$

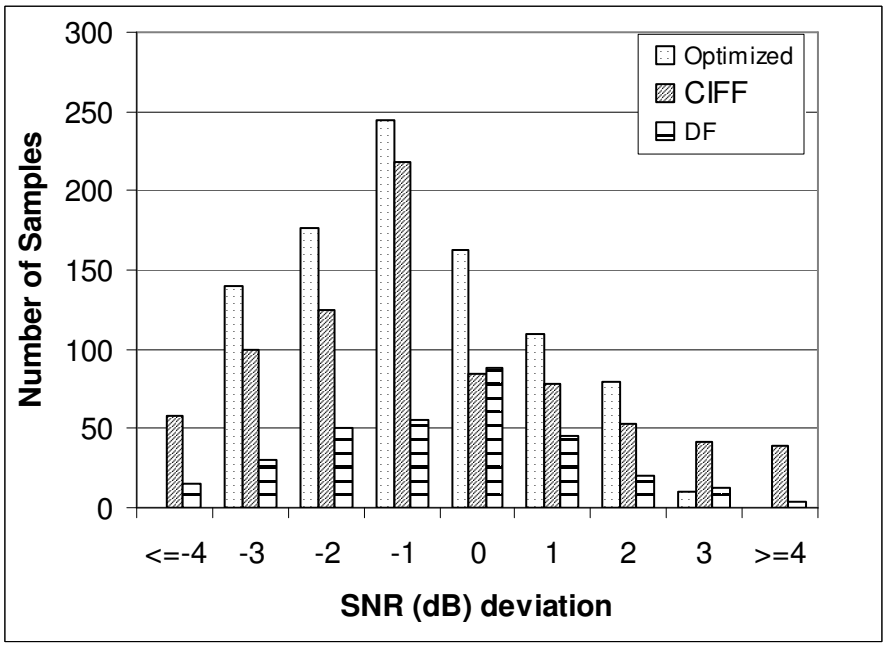


Figure 4.15: Bar diagram of peak SNR deviation for 5% coefficient variation.

Table 4.6: Comparision of behavior under ideal and non-ideal conditions: 4^{th} order modulator

Condition	DR	SNR peak
ideal	95.84 dB	93.85 dB
non-ideal	90.54 dB	91.31 dB

The state space model of the optimized modulator topology obtained through topology exploration and optimization is

$$\mathbf{A}_{opt} = \begin{bmatrix} 0 & -20.5 & 0 & 1.5 \\ 12.1 & 0 & 11.1 & 0 \\ 0 & 22.3 & 0 & -16.3 \\ -6.1 & 0 & 15.2 & 0 \end{bmatrix} \times 10^5 \quad \mathbf{B}_{opt} = \begin{bmatrix} 38.7 & -14.7 \\ 0 & -49.8 \\ 4.8 & -46.6 \\ 0 & -2.8 \end{bmatrix} \times 10^5 \tag{4.55}$$

$$\mathbf{C}_{opt} = \begin{bmatrix} 0 & 2.04 & 0 & 6.71 \end{bmatrix} \qquad \mathbf{D}_{opt} = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

The generated OTA-C topology is shown in Fig. 4.16. The load capacitances are taken as $10pF$. The Gm values are shown in the figure.

A comparison between the behavior of the generated topology under ideal and

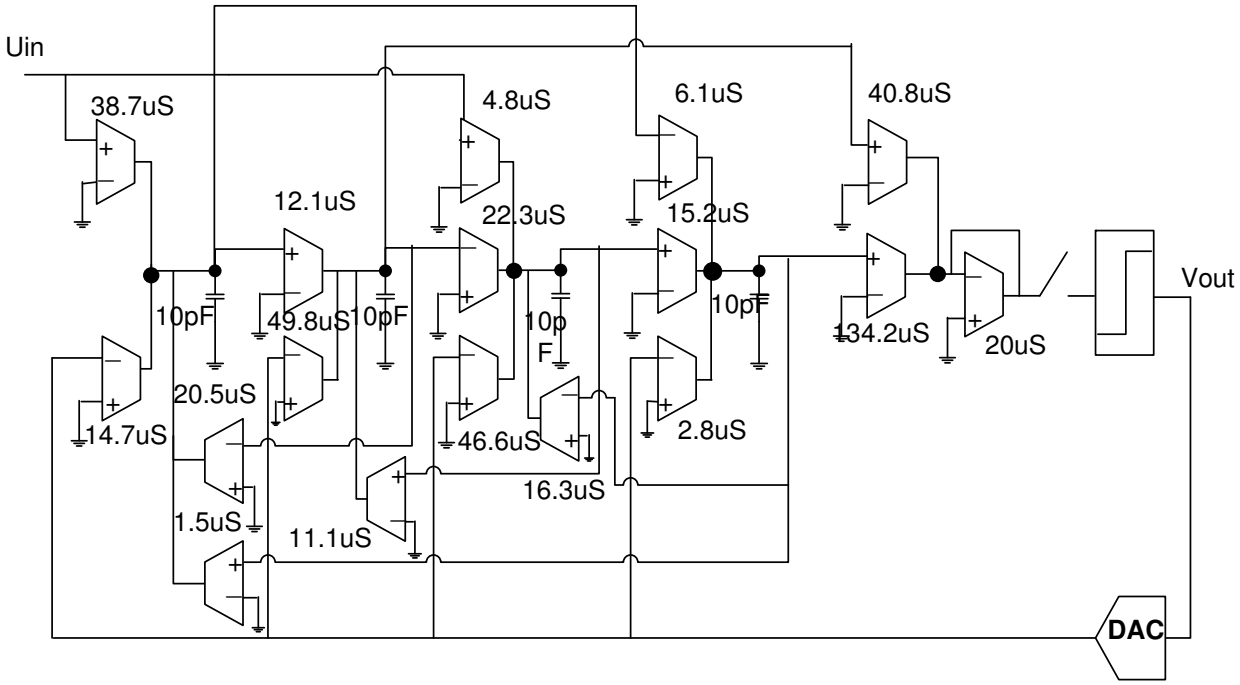


Figure 4.16: Synthesized topology for 4^{th} order modulator.

Table 4.7: Comparison in terms of yield for coefficient variation: 4^{th} order modulator

Variation	Optimized	CIFF
$\pm 2\%$	1000	988
$\pm 5\%$	890	723
$\pm 10\%$	483	305

non-ideal conditions of simulation is reported in Table 4.6.

In order to compare the performances of the optimized topology, we choose a 4^{th} order CIFF topology as shown in Fig.4.18. The coefficients and the corresponding Gm values are calculated following the technique mentioned earlier, such that the desired specifications and the feasibility constraints are satisfied. A comparison in terms of the yield between the two topologies for $\pm 2\%$, $\pm 5\%$ and $\pm 10\%$ variation is reported in Table 4.7. The distribution of the SNR deviation from the nominal value for $\pm 5\%$ coefficient variation is shown in Fig. 4.19. We thus conclude that for the 4^{th} order modulator even, the generated topology is better than the standard topology in terms of the sensitivity performance. A comparison between the relative power cost and hardware complexity of the generated topology and the CIFF topology is reported in Table 4.8. From it, we observe that the relative power cost is significantly

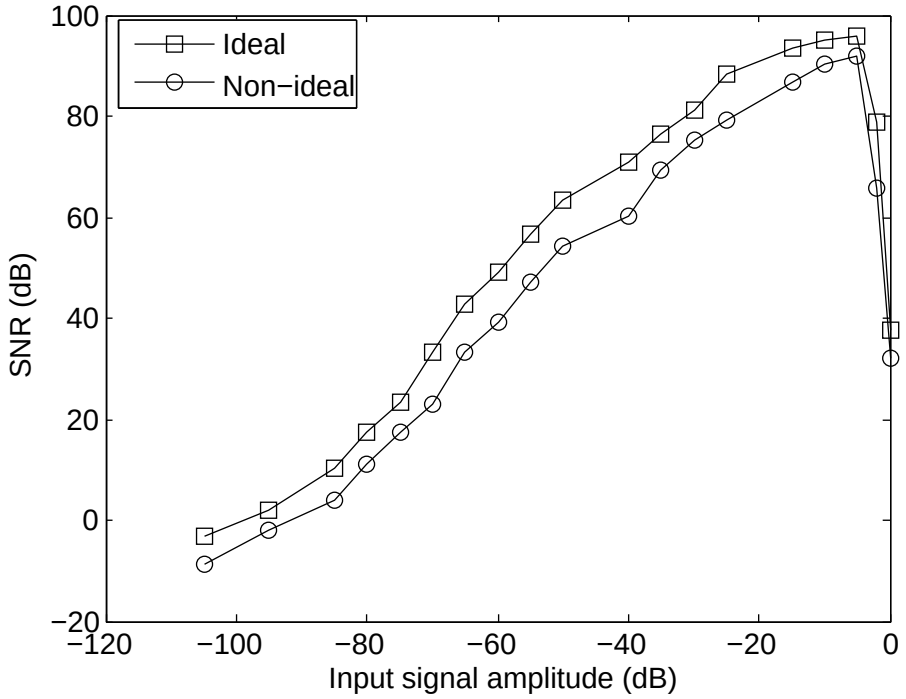


Figure 4.17: Comparison of SNR plot for generated 4^{th} order modulator topology under ideal and non-ideal conditions.

Table 4.8: Comparison between the topologies for relative power and complexity

Metrics	Optimized	CIFF
relative power	1.09×10^{-8}	5.17×10^{-7}
complexity	16	14

less for the generated topology in comparison to the CIFF topology, although the hardware complexity is greater.

4.6 Conclusion

In this chapter a methodology for the top-down generation of an optimal component-level topology for linear analog systems has been presented. The topology generation procedure started from a transfer function description of the system. The component-level topologies are synthesized from the transfer function via state space models using analog computation technique. The topology exploration process is implemented at the state space model level. Similarity transformation matrix is used as the topology transformation operator. An optimal state space model is

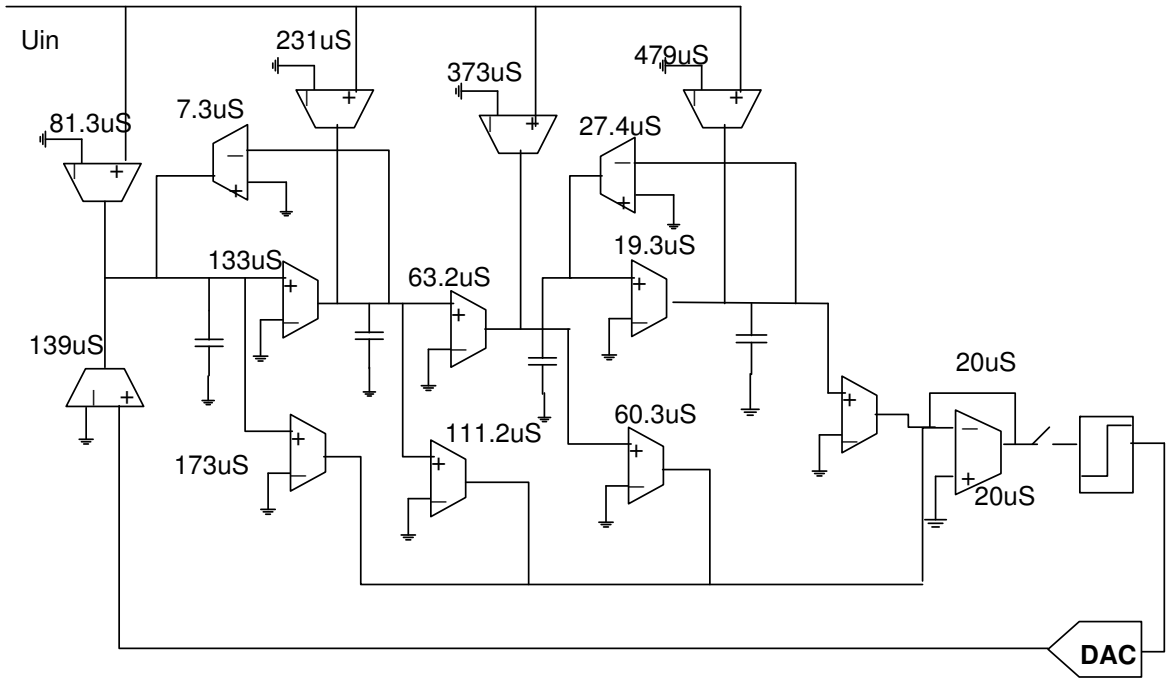


Figure 4.18: 4^{th} order CIFF topology.

selected based upon the topology performances employing a simulated annealing based optimization procedure. It is then realized using OTAs and capacitors. The entire methodology is illustrated with continuous time $\Sigma\Delta$ modulator system as a case study. The synthesized topology is optimized for (i) modulator sensitivity, (ii) hardware complexity and (iii) power consumption. Detailed experimentation has been carried for a 3^{rd} order and a 4^{th} order modulator topology. It is concluded from the experimental results that the generated topology is better in performances compared to commonly used topologies and satisfy the desired specifications under circuit-level non-idealities.

Through this methodology, the designer is thus able to specify the design goal and desired specifications at a higher level of abstraction in terms of transfer functions, while the design tool is able to generate an optimal component-level topology directly from the transfer functions in a highly automated manner. In addition, the generated topology is guaranteed to work satisfactorily even under circuit-level non-ideal conditions.

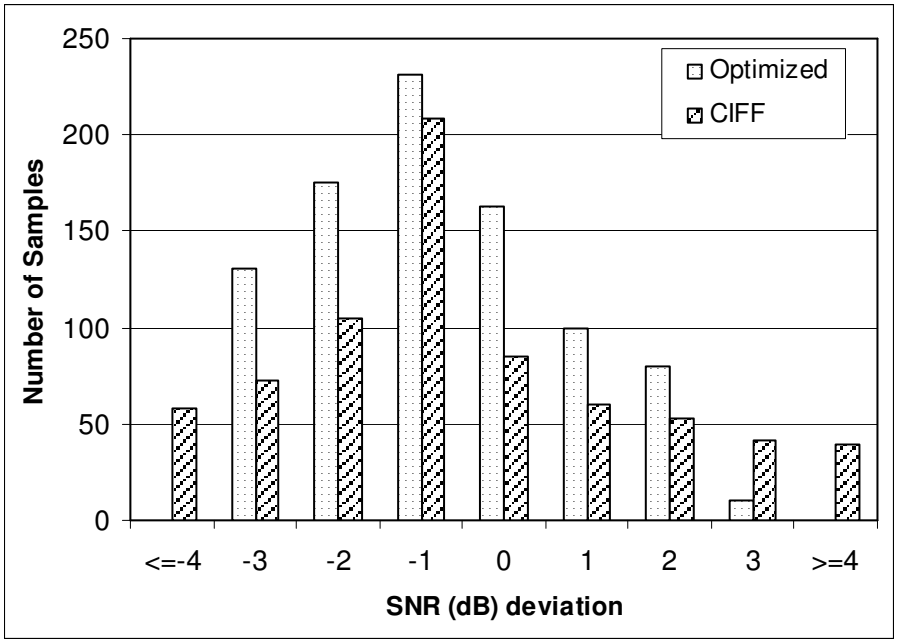


Figure 4.19: Bar diagram of peak SNR deviation for 5% coefficient variation for the 4^{th} order modulator.

Chapter 5

High-Level Specification Translation

The task of high-level specification translation forms an important step in an analog high-level design process. In this chapter we present a design space exploration procedure for high-level specification translation. A meet-in-the-middle approach is followed for constructing the feasible design space. Least squares support vector machine principle is used to accurately identify the actual geometry of the feasible design space. Genetic algorithm is used in the exploration procedure. The benefit of the present methodology is the ability to obtain practically correct circuit-level specifications of the component blocks of the system in a single pass.

The chapter is organized as follows: The problem is formulated in Section 1. The details of the feasible design space construction and identification procedure are provided in sections 2 and 3 respectively. The DSE procedure using GA is described in section 4. A comparison between the present methodology and the existing methodologies have been described in section 5. The case studies are given in section 6. Finally the chapter is concluded in section 7.

5.1 Problem Formulation

In a hierarchical analog design methodology, during the architectural design stage (component-level of abstraction) the overall architecture of the system is first decomposed into several component blocks, defined by their behavioral models. The specifications of these component blocks are then derived from the specifications

of the complete system, so that they can be designed separately. This process is referred to as high-level specification translation [2, 82]. A hierarchical specification translation process is generally implemented through a design space exploration (DSE) procedure.

The specification translation problem is mathematically defined by the following transformation function [29]

$$\Phi_j \leftarrow \Psi_j[\mathcal{B}_1(\bar{X}_1), \mathcal{B}_2(\bar{X}_2), ..., \mathcal{B}_P(\bar{X}_P)] \tag{5.1}$$

where Φ_j is the j^{th} functional specification of the system. $\bar{X}_i$ is the independent specification parameter vector for the i^{th} component block. These are considered as design parameters in the DSE procedure. $\mathcal{B}_i$ is the corresponding parameterized behavioral/performance model in terms of the design parameters. The constraint model Ψ_j for the total system is constructed by combining the parameterized high-level models of the individual component blocks. The transformation process (5.1) is performed through an iterative numerical procedure, generally represented as follows:

$$\begin{aligned} &\text{Minimize} && \textstyle\sum_{j=1}^{J} \omega_j[\Phi_{jt} - \Phi_{js}] \\ &\text{such that} && f_a(\bar{X}) \leq 0 \text{ and } f_c(\bar{X}) \leq 0 \end{aligned} \tag{5.2}$$

where $\bar{X} = \left[\bar{X}_1, \bar{X}_2, ..., \bar{X}_P\right]$ is the set of design parameter vectors for all the component blocks, $f_a(\bar{X}) \leq 0$ defines the application bounded space $\mathcal{D}_a$ and $f_c(\bar{X}) \leq 0$ defines the circuit realizable space $\mathcal{D}_c$. Φ_{jt} is the target value for the j^{th} functional specification and Φ_{js} is the simulated value, obtained after evaluating Ψ_j. ω_j is the associated weight.

5.2 Feasible Design Space Construction

In this section, we describe a meet-in-the-middle approach for constructing the feasible design space. The construction of the application bounded space is described in subsection 5.2.1 and the circuit realizable space is described in the subsection 5.2.2.

5.2.1 Application Bounded Space

The application bounded space is defined by a set of constraints applied on the specification parameters of the individual component blocks due to the desired specifications and design constraints of the chosen application system as well as mutual interaction between the component blocks of the system topology. The set of constraints $f_a(\bar{X}) \leq 0$ is described as a system of equations and inequalities, which are formulated by exploiting circuit knowledge. The formulation is illustrated with a toy example. Consider a system with a voltage amplifier and a low pass filter connected in series. Let the desired specifications of the system be: total gain $A_T \in \left[\underline{\mathrm{A}}_T, \bar{A}_T\right]$, and design constraints 1) maximum input signal = V mV, 2) bandwidth = f_c KHz and 3) input signal frequency = f_{in} MHz. Suppose $\langle A_1, \mathrm{Lin}_1, \mathrm{B}_1\rangle = \bar{X}_1$ and $\langle A_2, \mathrm{Lin}_2, \mathrm{B}_2\rangle = \bar{X}_2$ be the specification parameter (gain, input linearity and bandwidth) vectors for the two component blocks. Then the following equations and inequalities can be derived from circuit knowledge.

$$\begin{aligned}
\underline{\mathrm{A}}_T &\leq A_1 \times A_2 \leq \bar{A}_T && (5.3)\\
A_1 - nA_2 &= 0,\ n = 1, 2, ... && (5.4)\\
\mathrm{Lin}_1 &> V && (5.5)\\
\mathrm{B}_1 &> \mathrm{f}_{in} && (5.6)\\
\mathrm{Lin}_2 &> A_1 \times V && (5.7)\\
\mathrm{B}_2 &= \mathrm{f}_c && (5.8)
\end{aligned}$$

Equation (5.4) captures the interaction between the gain of the two blocks and (5.3) and (5.4) define the mutual interaction relation. Equation (5.7) also captures the interaction between the specification parameters of the two blocks. In (5.4), the exact value of n depends upon designer's experience. Equations (5.5), (5.6) and (5.7), define the lower bounds of the specification parameters. The exact values of the upper bounds depend upon the designer's experience.

The problem of constructing the application bounded space $\mathcal{D}_a$ is thus translated to finding solutions of $f_a(\bar{X}) \leq 0$ over an interval of $\bar{X}$. This is solved in the present work through an interval analysis technique. The interval analysis technique is based on the concepts of interval arithmetic [83]. In interval arithmetic, real numbers are replaced by intervals which are combinations of a lower bound and upper bound on the allowable value range of a variable. In order to perform computations in

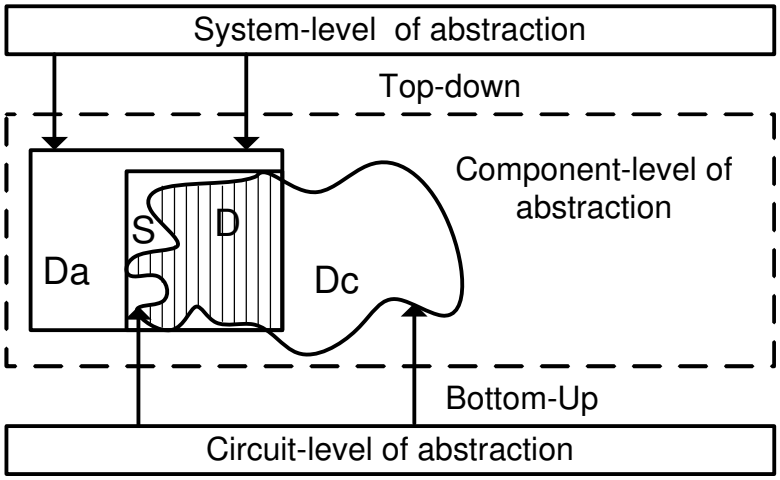


Figure 5.1: Meet-in-the-middle way of constructing $\mathcal{D}$, $\mathcal{D} = \mathcal{D}_a \cap \mathcal{D}_c$, S is the search space.

interval arithmetic, all basic arithmetic operations like addition, multiplication, etc., are replaced by interval versions. While solving equations using interval analysis technique, the solution (or whole set of solutions, if more than one) is enclosed into an interval. Whenever there is more than one variable in the problem, the solution is enclosed into a multidimensional interval rectangle. The commonly used methods for solving equations/inequalities using interval analysis technique are Krawczyk method, Hansen and Sengupta method [83]. In the present work, these have been implemented using Matlab Intlab interval analysis toolbox [84]. In simpler cases, as in (5.5)-(5.8), this can be directly converted to an inclusive/exclusive interval. The application bounded space $\mathcal{D}_a$ for a component block is constructed by combining the interval rectangles corresponding to all the specification parameters. The space $\mathcal{D}_a$ is thus constructed in a top-down fashion and is geometrically represented by a hyperbox as shown in Fig. 5.1. A specification parameter vector $\bar{X}_i$ is said to be application feasible if each element of the vector lies within the space $\mathcal{D}_a$ and satisfies the mutual interaction relation (if exists).

5.2.2 Circuit Realizable Space

A set of discrete tuples of circuit realizable specification parameters constitute the circuit realizable space $\mathcal{D}_c$. This is constructed using the data generation technique discussed in chapter 3 of the dissertation. Each component block is implemented at the circuit-level of abstraction and is simulated through SPICE. The perfor-

mance parameters of a component block, e.g., gain, bandwidth at the circuit-level of abstraction are the specification parameters of the corresponding block at the component-level of abstraction. These are extracted from SPICE simulation results. A set of constraints is applied on the transistor sizes as well as on circuit performances to extract feasible tuples only. The applied circuit performance constraints are taken to be relatively weak compared to the box constraints derived in the top-down phase. This ensures that several extracted parameter tuples lie within the hyperbox $\mathcal{D}_a$. The space $\mathcal{D}_c$ (see Fig. 5.1) is thus constructed using a bottom-up approach.

5.2.3 Feasible Design Space

The feasible design space $\mathcal{D}$ is a subset of the application bounded space, which is circuit realizable, as shown in Fig. 5.1. The tuples of design parameters which lie within $\mathcal{D}$ are considered as feasible tuples and the rest as infeasible tuples. The bounding hyperbox S of $\mathcal{D}$ is used as search space for the DSE procedure. Three important issues are clear from Fig.5.1 - (a) $\mathcal{D}$ is smaller in size than either $\mathcal{D}_a$ or $\mathcal{D}_c$. This speeds up the DSE procedure. (b) To check that a point chosen in S is also in $\mathcal{D}$, an accurate representation of the actual geometry of $\mathcal{D}$ is required. A poor approximate representation may yield inaccurate solutions leading to repetitive design iterations. (c) For the final solution point to be robust, it should not be at the periphery of $\mathcal{D}$. Peripheral solutions may turn out to be infeasible if errors creep in while realizing the solution points at the circuit-level of design. The last two issues are dealt in detail in the subsequent sections.

5.3 Feasible Design Space Identification

A two class LS-SVM technique is used to accurately infer the actual geometry of $\mathcal{D}$. The separating boundary between the two classes of tuples (feasible and infeasible) is implicitly described by a binary classification function $\mathcal{F}_i(\bar{X}_i) \rightarrow \{1, 0\}$. The value '1' signifies the feasible tuples whereas the value '0' signifies the infeasible tuples. Thus the SVM model associated with each component block defines its feasibility model.

The classifier function is constructed using the principle detailed in the appendix A.2 and Matlab *lssvm* toolbox[61]. The training set consists of samples from both

the classes. Large number of infeasible samples are taken compared to feasible samples [32]. The data elements of the sample tuples are logarithmically scaled in the $[0, 1]$ range. Radial basis function is used as the kernel function. A set of two hyperparameters, the regularization constant γ and the kernel parameter σ^2 determines the generalization ability of the classifier which are computed through the genetic algorithm-based optimization procedure discussed in section 3.2.3.1 and the hold-out technique. The objective is to minimize the rate of misclassifications for the test samples.

The SVM classification function is constructed for individual component blocks. If $\mathcal{F}(\bar{X})$ be the feasibility model of the complete system, then $\mathcal{F}(\bar{X})$ is related to individual feasibility models as $\mathcal{F}(\bar{X}) = \mathcal{F}_1(\bar{X}_1) \wedge \mathcal{F}_2(\bar{X}_2) \wedge .... \wedge \mathcal{F}_P(\bar{X}_P)$.

The construction and identification of the feasible design space $\mathcal{D}$ is considered to be a pre processing step of the DSE-based specification translation task. The task of generating circuit realizable specification data is a one-time process. The task of constructing $\mathcal{D}_a$ is much less time consuming compared to $\mathcal{D}_c$. With the addition of any new desired functional/performance specifications of the system or with the change of the desired specification values, the space $\mathcal{D}_a$ needs to be computed fresh and the SVM models need retraining. The circuit realizable data set can however, be reused.

5.3.1 Accuracy Measurement

For evaluating the performances of the SVM classifiers, a set of test samples is identified and three quality metrics, *viz.,* sensitivity (Sen), specificity (Sp) and accuracy (Acc) are measured. These are defined as follows:

Let I denote the entire design space, $\mathcal{D}$ be the feasible design space and $\mathcal{D}'$ be the approximated feasible design space. Thus I is divided by $\mathcal{D}$ and $\mathcal{D}'$ into four subspaces: TP of true positives, TN of true negatives, FP of false positives and FN of false negatives. This is shown in Fig. 5.2. Sensitivity (Sen) is defined as the percentage of true positives relative to all the positive instances.

$$Sen = \frac{|TP|}{|TP| + |FN|} \tag{5.9}$$

Specificity (Sp) is defined as the percentage of true negatives relative to all the

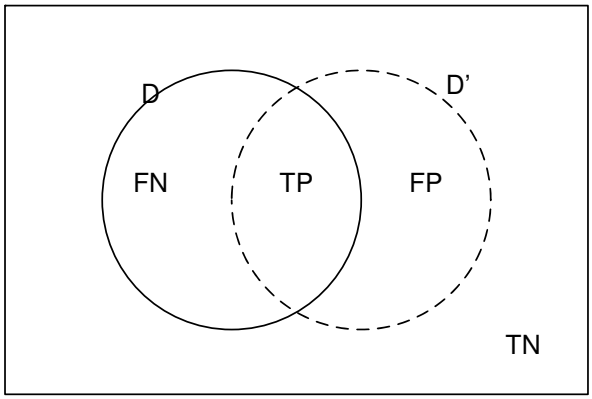


Figure 5.2: Feasible design space and its subspaces.

negative instances.

$$Sp = \frac{|TN|}{|TN| + |FP|} \tag{5.10}$$

Accuracy (Acc) is defined as the percentage of correctly classified instances in the data set.

$$Acc = \frac{|TP| + |TN|}{|I|} \tag{5.11}$$

For a good classifier, these values ideally should be equal to unity.

5.4 Design Space Exploration

The equation-based approach as well as the simulation-based approach discussed in section 2.1 of the dissertation are used for performance model evaluation. The functional specifications of the system are estimated by evaluating the behavioral models of the system and the performance specifications are estimated by evaluating the corresponding high-level performance models. The behavioral models are constructed using the analytical techniques discussed in section 2.2.1.1 of the dissertation. The models are implemented under Simulink environment. The high-level performance models are constructed using the methodology discussed in chapter 3 of the dissertation.

In this section, we discuss the formulation of the cost function and the optimization algorithm that has been used for exploration purpose.

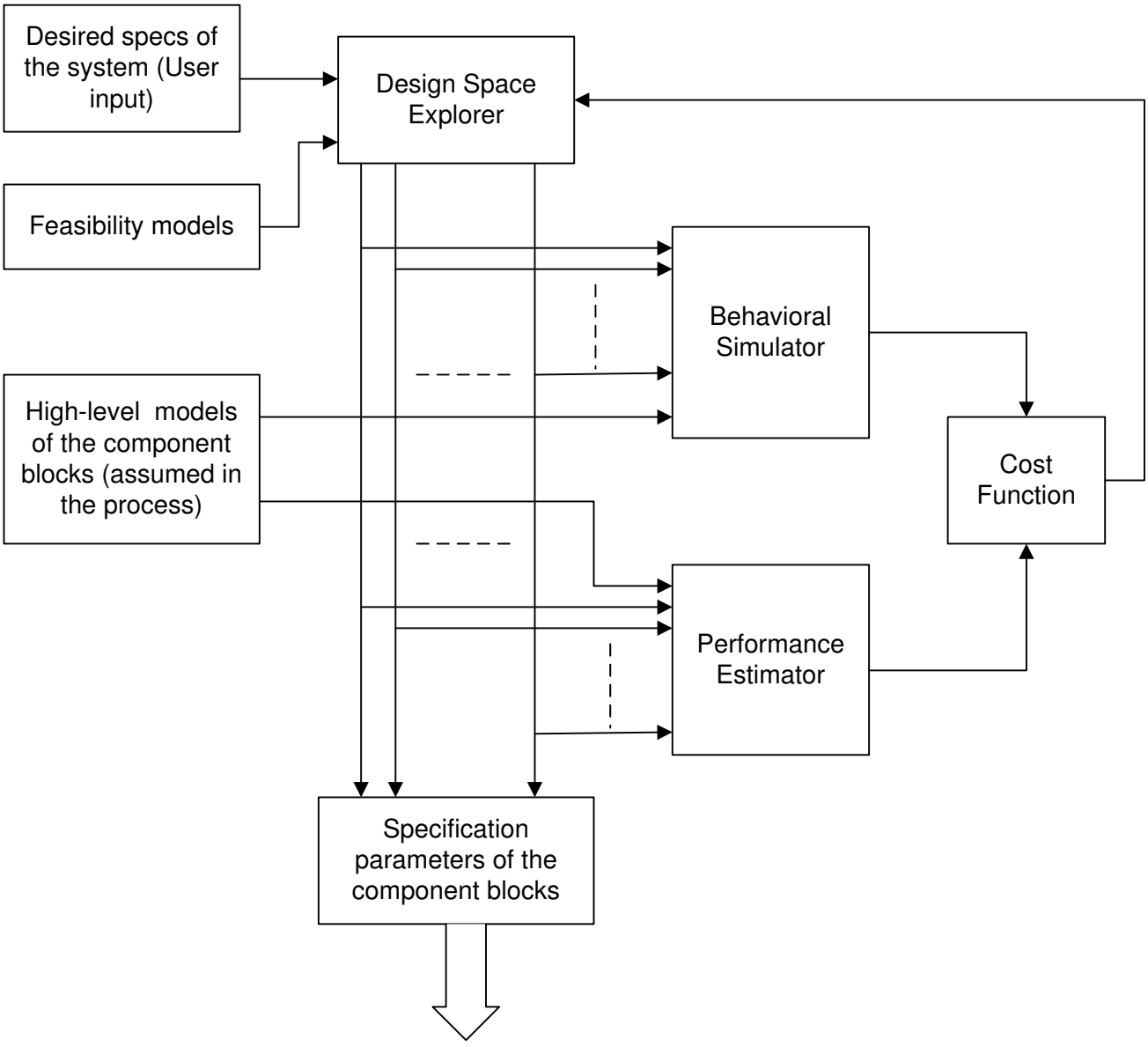


Figure 5.3: DSE mechanism.

5.4.1 Cost Function Formulation

5.4.1.1 Primary Objectives

The primary objectives are: (i) to minimize the relative error between the desired value of the functional specification of the complete system and that obtained by evaluating the system constraint model Ψ in (5.1) through behavioral simulation and/or performance model evaluation and (ii) to ensure that the specification parameter tuples are selected from the feasible region of the search space.

5.4.1.2 Secondary Objectives

Constrained optimization algorithms, in general minimize a cost function by pushing the design variables to the boundary of S. However, points away from the periphery of $\mathcal{D}$ have better tolerance to circuit parameter variations than the peripheral points. The tolerance space for those points have higher probability to be completely within $\mathcal{D}$ [82]. To achieve this, a secondary objective function is added within the cost function. In this formulation, we assume a hyperellipsoid shape for $\mathcal{D}$ [82] for the secondary objective.

The overall cost function is expressed as an weighted sum of the primary and secondary objective functions. This is given as

$$Cost\left(\bar{X}\right) = \frac{\sum_{j=1}^{J} \omega_{1j} \left|\frac{\Phi_{jt}-\Phi_{js}}{\Phi_{jt}}\right| + \omega_2 \left|\left|\frac{\bar{X}-\bar{X}_C}{\bar{X}_C}\right|\right|_2}{\mathcal{F}(X)+\epsilon} \qquad \bar{X} \in \mathcal{D} \tag{5.12}$$

where ω_{1j} and ω_2 are the weights associated with the primary and secondary objectives. X_C is the centre of the search space and ϵ is a very small number (say, 10^{-10}). Note that the first term in the numerator corresponds to the transformation process in (5.2). For infeasible tuples, $\mathcal{F}(\bar{X}) = 0$ and thus the cost is very high. For selecting the weights a trial and error procedure is usually followed. Much higher weights are assigned to the primary objectives.

5.4.2 Exploration Algorithm

Conventional binary coded GA as discussed in the appendix B is used as the exploration algorithm. A random set of twenty chromosomes (specification parameters) constitute the initial population. New generation of chromosomes are created with

the help of crossover and mutation operation (crossover probability 90% and mutation probability 10%). Elitism is used to pass the best chromosome of one generation to its next generation unaltered. Fitness of chromosomes is calculated as the inverse of the cost function (5.12). The GA terminates if the cost function value becomes equal to a predetermined small number or GA iterates for a very large number of loops.

5.5 Comparison with Existing Methodologies

The described methodology is closely related to that described in [56, 29] and [32, 35]. In [56, 29], a linearized polytopal approximation has been used to identify the geometry of the feasible design space. As a result, several runs of the DSE process, caused by over-or underestimation of the true region is required to achieve accurate results. On the other hand in our approach, the actual feasible design space is identified accurately avoiding such overall iterations. In [32, 35], the feasible design space construction process considers only the circuit realizable space. The system constraints and the mutual influence between the component blocks have not been considered. On the other hand, with our meet-in-the-middle approach these are considered systematically while constructing the application bounded space in the top-down construction phase. By reducing the size of the design space through the intersection operation, the DSE process is sped up. These features play a major role in our methodology to obtain a set of practically correct circuit-level specifications of the component blocks of a system through a fast exploration process in a single pass. However, this advantage comes at the cost of increased overhead for construction and accurate identification of the feasible design space. This overhead is much greater than that in [56, 29], and comparable to that in [32, 35]. The present methodology is thus suitable for systems with less number of specification parameters, which may have tight nonlinear coupling.

5.6 Experimental Results

To demonstrate the entire methodology, we choose two complete systems - the readout electronics for a MEMS capacitive accelerometer sensor and the $\Sigma\Delta$ modulator system case studies. The basic block diagram of the topology along with the desired

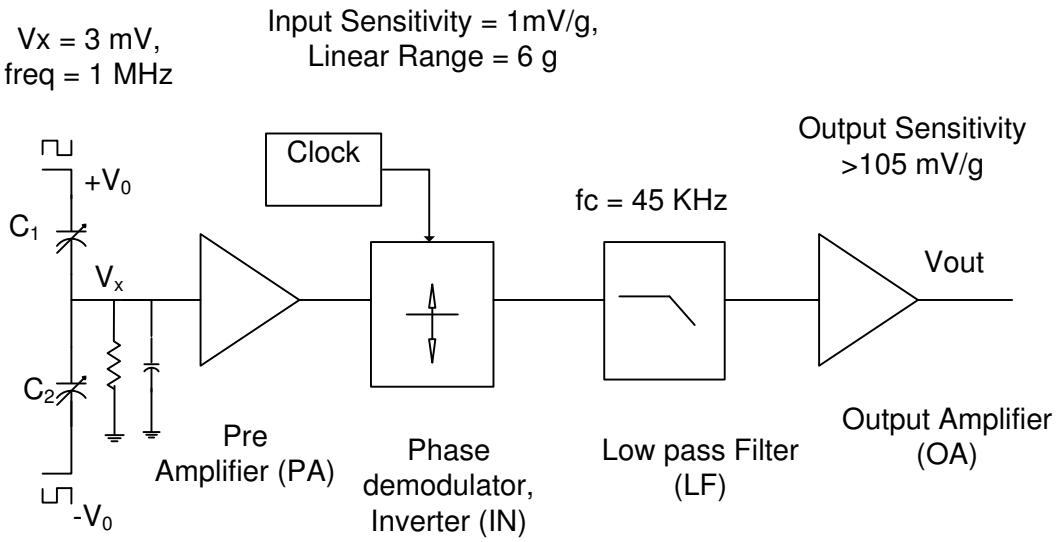


Figure 5.4: Considered system for experimentation.

Table 5.1: Application bounded constraints: Experiment 1

Params	PA	IN	LF	OA
A	$[15, 18.5]$	$[0.9, 1.1]$	$[0.9, 1.1]$	$[6, 7.4]$
Lin (mV)	$[15, 900]$	$[111, 900]$	$[122.1, 900]$	$[134.3, 900]$
BW(MHz)	$[2, 10]$	$[2, 40]$	$[0.0447, 0.0453]$	$[2, 20]$

specifications and design constraints is shown in Fig. 5.4. The synthesizable components are the pre-amplifier (PA), inverter (IN) of the phase demodulator, low pass filter (LF) and the output amplifier (OA). These are designed using OTAs (refer to Fig. 5.5) and capacitors. The chosen design parameters are gain (A), input linearity (Lin), bandwidth (BW) and output swing (OS) of all the synthesizable blocks.

5.6.1 Experiment 1:

The desired functional specification is to achieve an output voltage sensitivity $\geq 105mV/g$ and minimize the performance – input referred thermal noise.

The target output sensitivity A_D is considered as an interval $[108.5, 111.5]$. A_D is related to individual gains as

$$108.5 \leq A_D = A_{PA} \times A_{IN} \times A_{LF} \times A_{OA} \leq 111.5 \tag{5.13}$$

A_{IN} and A_{LF} lie in the interval $[0.9, 1.1]$ and $A_{PA} = 2.5 \times A_{OA}$ Through inter-

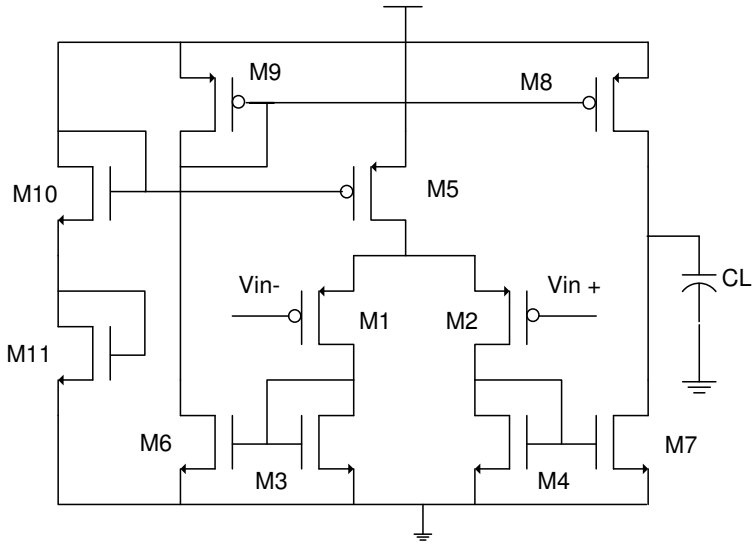


Figure 5.5: PMOS OTA circuit

Table 5.2: Circuit Realizable Constraints: PA block

Sizes (L=1 um)	Ranges
$W_1 = W_2$	[280 nm, 400 um]
$W_3 = W_4 = W_6 = W_7$	[280 nm, 50 um]
$W_8 = W_9$	[280 nm, 50 um]

val analysis method, the intervals of the gain parameter of the individual component blocks are determined. The maximum input signal amplitude is V = input sensitivity $\times$ linear range $= 6mV$. The lower bound of the input linearity parameter of the PA block, i.e., Lin_{PA} is taken to be $2.5 \times V$. The same for the IN, LF and OA block is fixed at $\bar{A}_{PA} \times V$, $\bar{A}_{PA} \times \bar{A}_{IN} \times V$ and $\bar{A}_{PA} \times \bar{A}_{IN} \times \bar{A}_{LF} \times V$ respectively, where $\bar{A}_{PA}$, $\bar{A}_{IN}$ and $\bar{A}_{LF}$ are the upper bound of the interval for A_{PA}, A_{IN} and A_{LF} respectively. The upper bound of the intervals of the linearity parameter for all of them are fixed at half of the supply voltage. The lower bound of the interval for the bandwidth parameter of the PA, IN and OA block is fixed at $2 \times$ input signal frequency $= 2MHz$. The upper bound is taken some higher values. The bandwidth of the LF block is same as the cut-off frequency, which is enclosed within an interval. For the swing (OS) parameter, application constraints have not been imposed. The application bounded constraints for all the component blocks are summarized in Table 5.1. These define the space $\mathcal{D}_a$ for the individual component blocks.

Table 5.3: SVM Performances: Experiment 1

Block	# Test data	σ^2	γ	Sen	Sp	Acc
PA	460	5	800	0.956	0.997	0.993
IN	560	3.5	650	0.982	0.996	0.992
LF	456	4.2	720	0.968	0.995	0.993
OA	528	5	800	0.978	0.993	0.992

A large set of circuit realizable specification data with wide range of values are generated for the PA block through SPICE simulation. The circuit realizable constraints applied on the transistor sizes of the OTAs are reported in Table 5.2. Since all the blocks have identical circuit topology, the data set can be reused. The performances of the constructed SVM classifiers for all the blocks along with the required hyperparameter values are tabulated in Table 5.3. The achieved values of the metrics are close to their ideal values ≈ 1.

The total time required to construct the space $\mathcal{D}_a$ for all the blocks is 3s CPU time for a 512MB RAM, 3.00 GHz PC. The entire data generation process took nearly $4\frac{1}{2}$ hours CPU time. The total training time for constructing the SVM models of all the blocks is about 44 minutes.

GA with parameters (crossover factor β_c and mutation factor β_m) as mentioned in the first row of Table 5.5 are used for DSE. Weights used in (5.12) are also specified. The nominal values of the design parameters for the 'non-peripheral'(incorporating the secondary objective) and 'peripheral' cases, as obtained after DSE are tabulated in Table 5.4. We observe from the results, that the optimization process with secondary objective leaves sufficient margin for most of the parameters. The time complexities are shown in the first row of Table 5.5.

For validation of the results, the topology is implemented at the circuit-level with the determined 'non-peripheral' specifications and is simulated with SPICE. The end results are tabulated in the first two rows of Table 5.6. The SPICE simulated output curve is shown in Fig. 5.6. The satisfaction of SPICE results with the desired functional specification of the system validates the overall procedure.

Table 5.4: Translated Specifications: Experiment 1

Blocks	Parameters	Feasible Range	Nominal Design	
			Non-peripheral	peripheral
PA	A	[15.5,18]	16.55	16.15
	Lin (mV)	[15.3,25.20]	21	18
	BW (MHz)	[2.1, 8]	5.38	7.27
	OS (mV)	[820,850]	831.30	826
IN	A	[0.9,1.1]	1.03	1.04
	Lin (mV)	[280,325]	303	290
	BW (MHz)	[10, 30]	23.65	28.98
	OS (mV)	[410,520]	452	517.82
LF	A	[0.9,1.1]	1.01	1.06
	Lin (mV)	[280,325]	300	320
	BW (KHz)	[44.8,45.1]	45.01	44.95
	OS (mV)	[410,510]	479.2	490
OA	A	[6.2,7.1]	6.57	6.43
	Lin (mV)	[140,170]	150	158
	BW (MHz)	[2.5, 15]	8.65	14.99
	OS (mV)	[700,740]	726	742

Table 5.5: GA results: averaged over 10 runs, 3.0 GHz 512 MB RAM PIV PC

Experiment	β_c	β_m	ω_1	ω_2	ω_3	Time
1	0.9	0.1	0.4	0.4	0.2	7 min
2	0.9	0.1	0.4	0.4	0.2	7.4 min

5.6.2 Experiment 2:

In this experiment, the desired specification is to achieve an output sensitivity $\geq 40mV/g$ with minimum noise. The application bounded space is computed fresh using the same technique as earlier. This is reported in Table 5.7. For SVM construction, the circuit realizable data generated during the previous experiment are reused. Thus the pre-processing time of the procedure is determined solely by the SVM training time. The performances of the constructed SVM models are reported in Table 5.8. The results show that the models are constructed quite accurately. The training time is almost same as the previous experiment. The specification parameters determined through the DSE procedure are tabulated in Table 5.9. The GA parameters used in the procedure as well as the time complexity of the DSE

Table 5.6: End results

Expt	Parameters	Specs	Simulated	SPICE	Error
1	Out Sens (mV/3g)	≥315	331.2	326.8	1.33 %
1	Noise (nV/$\sqrt{Hz}$)	Min	21.32	22.08	3.56 %
2	Out Sens (mV/3g)	≥120	139.4	136.2	2.30 %
2	Noise (nV/$\sqrt{Hz}$)	Min	35.45	36.78	3.75 %

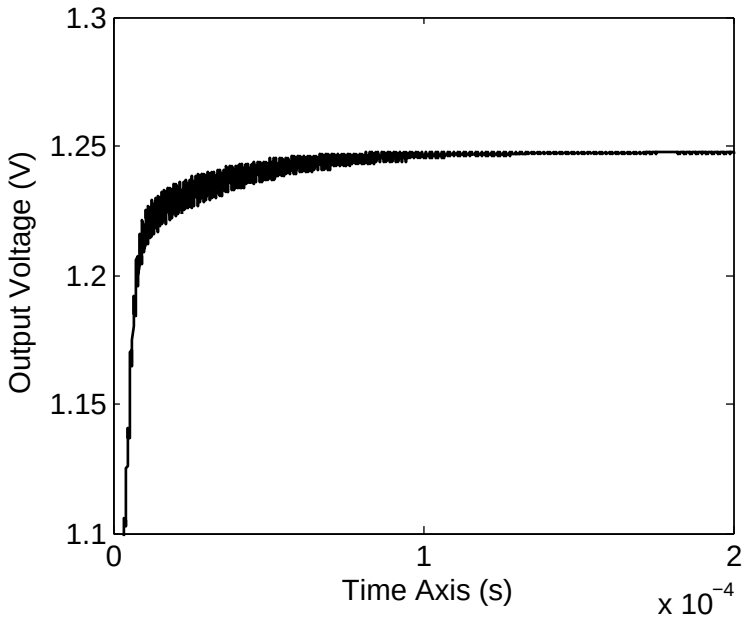


Figure 5.6: SPICE output amplitude with 3 mV input

Table 5.7: Application bounded constraints: Experiment 2

Params	PA	IN	LF	OA
A	$[9, 11]$	$[0.9, 1.1]$	$[0.9, 1.1]$	$[4, 6]$
Lin (mV)	$[15, 900]$	$[66, 900]$	$[72.6, 900]$	$[80, 900]$
BW(MHz)	$[2, 50]$	$[2, 40]$	$[0.0447, 0.0453]$	$[2, 100]$

Table 5.8: SVM Performances: Experiment 2

Block	# Test data	σ^2	γ	Sen	Sp	Acc
PA	520	12	310	0.986	0.991	0.995
IN	560	3.5	650	0.982	0.996	0.992
LF	456	4.2	720	0.968	0.995	0.993
OA	484	5	520	0.945	0.994	0.992

Table 5.9: Translated Specifications: Experiment 2

Blocks	Parameters	Feasible Range	Nominal Design	
			Non-peripheral	peripheral
PA	A	[9.2,10.7]	9.91	10.14
	Lin (mV)	[75.2,98.30]	81.13	94.45
	BW (MHz)	[15.8,38.3]	21.36	37.27
	OS (mV)	[760,810]	768.30	801
IN	A	[0.9,1.1]	1.0	0.96
	Lin (mV)	[280,325]	297.12	321.3
	BW (MHz)	[10, 30]	21.65	26.43
	OS (mV)	[410,520]	466.15	518.82
LF	A	[0.9,1.1]	1.02	0.98
	Lin (mV)	[280,325]	300	320
	BW (KHz)	[44.8,45.1]	45.00	45.05
	OS (mV)	[410,520]	425.3	515.28
OA	A	[4,5.7]	4.80	5.07
	Lin (mV)	[150,210]	170	179.8
	BW (MHz)	[25, 85]	48.15	64.99
	OS (mV)	[540,680]	586	665.2

procedure are reported in the second row of Table 5.5. The end results are reported in the second row of Table 5.6. This experiment demonstrates the fact that with change in desired specification values for the same set of specification parameter vectors, the complexity of the pre processing task reduces considerably, without sacrificing the accuracy of the end results.

5.6.3 Experiment 3:

In this experiment, we choose the $\Sigma\Delta$ modulator system, discussed in chapter 4 of the dissertation. A 3^{rd} order, single bit continuous time (CT) OTA-C $\Sigma - \Delta$ modulator along with the topology parameters and design constraints are shown in Fig. 5.7. The target functional specification is dynamic range $(DR) \geq 80dB$. The design constraints are based on [81].

The synthesizable component blocks are all the OTAs of the CT integrators and the comparator. In order to make the DSE procedure manageable, we assume equal specifications for all the OTAs. Only two SVM feasibility models are thus required, one for all the OTAs and the other for the comparator. The OTAs are implemented

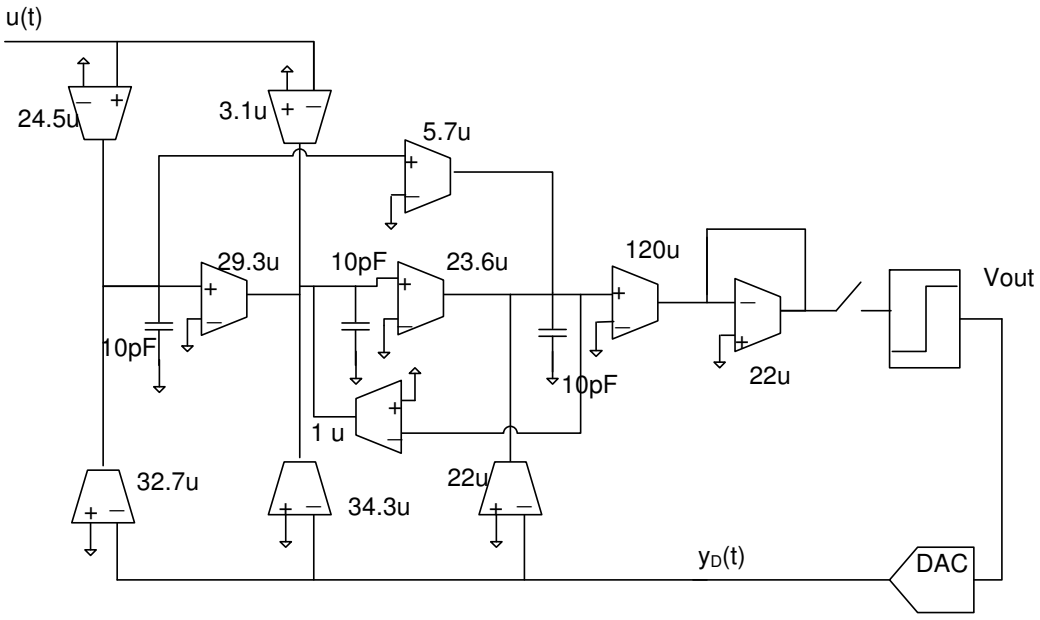


Figure 5.7: 3^{rd} order OTA-C based $\Sigma - \Delta$ modulator topology

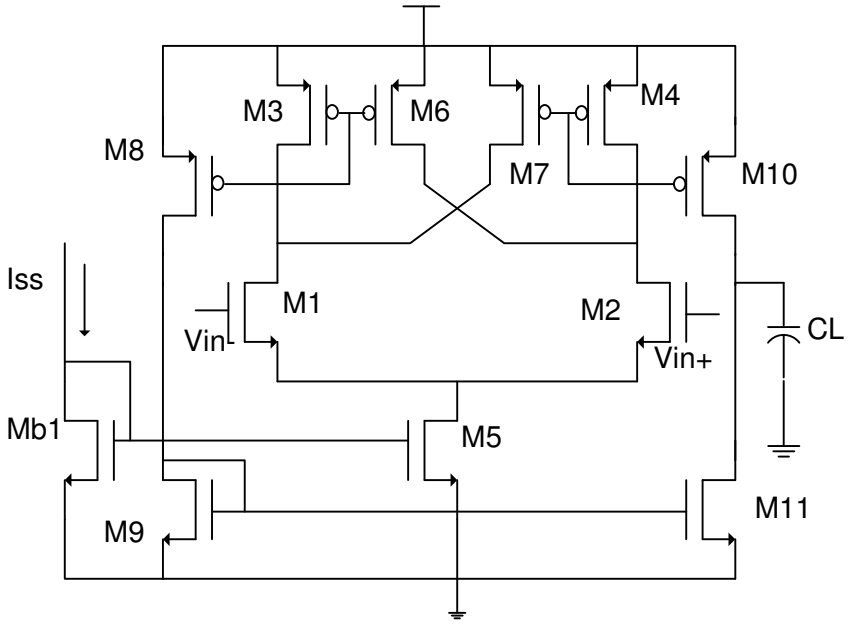


Figure 5.8: Comparator circuit

Table 5.10: Feasibility constraints for OTA and Comparator: Experiment 3

	Quantities	Range
Application Constraints (OTA) (Comparator)	Swing	$\geq$ 300 mV
	p_n	$\geq$ 13 MHz
	Ro	$\geq$ 1 MΩ
	Delay	$\leq$ 15.6 ns
Circuit params OTA	$W_1 = W_2$	280 nm, 200 um]
	$W_3 = W_4 = W_6 = W_7$	[280 nm, 50 um]
	$W_8 = W_9$	[280 nm, 50 um]
	I_b	[1 uA, 40 uA]
Circuit params Comparator	$W_{b1} = W_5$	3 um
	$W_3 = W_4 = W_8 = W_{10}$	[1 um, 10 um]
	$W_9 = W_{11}$	[280 nm, 50 um]
	$W_6 = W_7$	[1 um, 10 um]
	$W_6/W_3 = W_7/W_4$	$\geq$ 1
	I_b	50 uA

Table 5.11: Specifications for the component blocks of Experiment 3

Blocks	Parameters	Feasible Range	Nominal Design	
			Non-peripheral	Peripheral
OTA	Swing (mV)	[300,700]	537.0	332.84
	p_n (MHz)	[20, 80]	36.65	20.89
	Ro ($M\Omega$)	[1,10]	6.88	9.81
Comparator	Hysteresis (mV)	[0.5,10]	4.54	8.40
	Offset (mV)	[0.5,10]	6.30	1.1
	Delay (ns)	[4,15]	9.11	4.54

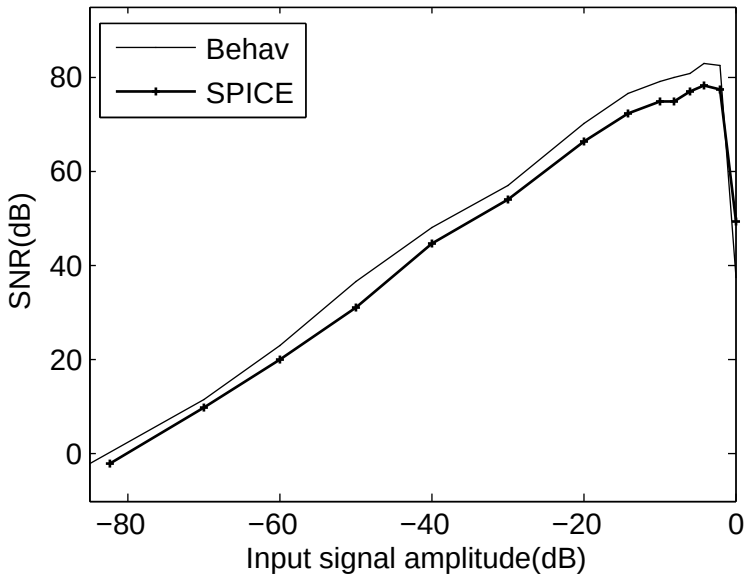


Figure 5.9: Behavioral and SPICE simulated SNR Plot of the modulator.

using the circuit shown in Fig. 5.5 and the comparator by the circuit shown in 5.8. The system bounded constraints as well as the sizing constraints for the OTAs as well as the comparator are tabulated in Table 5.10. The swing constraint for the OTAs is determined based on the maximum input signal amplitude or the DAC reference voltage. The non-dominant pole (p_n) of the OTA-C integrator needs to be greater than the sampling frequency [85]. The maximum value of the comparator delay is limited by the minimum value of the excess loop delay, which is taken as 20% of the sampling period [85]. Efficient SVM models are constructed as earlier. The specification parameters obtained through DSE for both 'non-peripheral' and 'peripheral' cases are reported in Table 5.11. The crossover and mutation probabilities are taken as same as earlier. The DSE process took 47 minutes. For validation of the results, the modulator is implemented at the circuit level as per the specifications obtained. The SNR curve, as obtained from the electrical simulation is provided in Fig. 5.9. The end results are tabulated in Table 5.12. We observe that the designed circuit satisfies the desired functional specification. This again validates the procedure.

5.7 Conclusion

This chapter presents a DSE procedure for determining specifications for individual component blocks of an analog system such that the given functional specifications

Table 5.12: End results: Experiment 3

Parameters	Specs	Behavioral	SPICE
Input signal bandwidth	100 KHz	100 KHz	100 KHz
DAC reference voltage	200 mV	200 mV	200 mV
Sampling frequency	13 MHz	13 MHz	13 MHz
Dynamic range	$\geq$ 80 dB	84.87 dB	83.2 dB

of the system are satisfied with optimized performances. A 'meet-in-the-middle approach is followed for the construction of the feasible design space. This is constructed as intersection of an application bounded space and a circuit realizable space. The reduced design space speeds up the exploration. LS-SVM principle is used to accurately identify the actual geometry of the feasible design space. GA is used for exploring the design space. The effectiveness of the procedure is illustrated with two practical systems. For verification of the results, the systems are implemented at the circuit-level based on the determined specifications. Final SPICE simulation results satisfy the desired specifications, validating the overall procedure. The benefit of the methodology is the ability to obtain practically correct circuit-level specifications of the component blocks of the system through a fast exploration procedure in a single pass.

Chapter 6

Conclusion and Directions for Further Research

In this chapter we summarize the major contributions of this thesis and discuss some of the directions for future scope of research.

6.1 Summary and Conclusions

The emphasis of the thesis is on optimization-based methodologies for the different tasks related to analog high-level design. The specific tasks for which the methodologies have been developed are: (i) high-level performance model generation, (ii) generation of an optimal component-level topology and (iii) high-level specification translation.

The thesis presents a methodology for generation of high-level performance models for analog component blocks using statistical learning technique. The transistor sizes of the circuit-level implementations of the component blocks define the sample space. Performance data are generated through SPICE simulation. For training of the model, only those samples are considered which satisfy a set of functional and performance constraints. Least squares support vector machine (LS-SVM) is used as regression function. The generalization ability of the constructed models have been estimated through hold-out method and a 5-fold cross validation method. Optimal values of the model hyper parameters are determined through grid search technique and a GA-based technique. The constructed performance models are used within a GA-based topology sizing process. The entire methodology has been im-

plemented under Matlab environment. The SVM models has been trained using Matlab toolbox. The methodology has been demonstrated with a set of experiments. Performance models corresponding to thermal noise, power consumption and output impedance of an operational transconductance amplifier have been developed. A comparison between the models constructed with the grid search-based training technique and the GA-based training technique w.r.t. generalization ability and training time is made. It is found that the training time is considerably less for GA-based training technique compared to the grid search-based training technique, with almost the same generalization ability. To demonstrate the topology sizing process, the interface electronics for a MEMS capacitive accelerometer has been chosen as an example. High-level models corresponding to required performances of all the component blocks have been developed. These are evaluated to estimate the performances of the topology. The predicted results have been compared with SPICE simulation results. The two sets of results match closely.

The thesis then presents a structural synthesis approach for top-down generation of an optimal component-level topology for linear analog systems. The topologies are generated from a transfer function model of the system via state space matrix models. The topology exploration process is modeled as a state space matrix exploration process. Similarity transformation matrix is used for generation of a new state space model from a given one. Simulated annealing based optimization technique is used to determine an optimal state space model such that the resultant topology is optimized for a set of performance parameters. The optimized state space model is realized by appropriate analog component blocks to generate an optimal component-level topology. As a case study, the thesis presents a methodology for generation of an operational transconductance amplifier (OTA)-capacitor (C) based topology for continuous-time $\Sigma\Delta$ modulator. The loop filter transfer functions are taken as inputs. The chosen performance metrics are system hardware complexity, sensitivity under parameter variation and relative power consumption. A 3^{rd} order and a 4^{th} order modulator have been chosen as examples for experimentation. The behavioral equivalence between the newly generated topologies under non-ideal conditions is validated through behavioral simulation. The dynamic ranges are determined. These are nearly equal. The generated topology satisfies the desired dynamic range under non-ideal conditions and overloading does not take place. The performances of the generated topology have been compared with that of two standard topologies. Monte Carlo analysis has been performed for comparing the

sensitivity performances. The yield and performance deviation has been computed. It is found that the generated topology is more tolerant to design parameter variations not only in terms of yield but also performance deviations. It is concluded from the experimental results that the generated topology is better in performances compared to commonly used topologies and satisfy the desired specifications under circuit-level non-idealities.

Finally, the thesis presents a methodology for high-level specification translation. A meet-in-the-middle approach is followed for the construction of the feasible design space. This is constructed as the intersection of an application bounded specification space and a circuit realizable specification space. Least squares support vector machine (LS-SVM) technique is used to identify an accurate geometry of the actual feasible design space. Genetic algorithm (GA) is used to explore the feasible design space. Two case studies, an interface electronics for MEMS capacitive accelerometer sensor and a continuous time $\Sigma\Delta$ modulator have been presented to demonstrate the effectiveness of the procedure. LS-SVM feasibility models have been constructed for all the component blocks. A set of performance metrics, viz., sensitivity, specificity and accuracy have been computed. These values are found to be close to their ideal values. With the determined specifications of the component blocks, the target systems are implemented at the transistor level and are simulated with SPICE. The SPICE simulation results satisfy the functional specifications of the system, validating the overall procedure.

In summary the main contributions of the present thesis are as follows:

1. A methodology is developed for generation of good high-level performance estimation models for analog component blocks using least squares support vector machine (LS-SVM). The models have high accuracy and good generalization ability. The model construction time is low.

2. A methodology is developed for generation of an optimal component-level topology for linear analog systems starting from a transfer function model of the system. The generated topology is ensured to perform satisfactorily under circuit-level non-ideal conditions. Through this methodology, the designer is able to specify the design goal and desired specifications at a higher level of abstraction in terms of transfer functions, while the design tool is able to generate an optimal component-level topology directly from the transfer functions in a highly automated manner.

3. A methodology is developed for high-level specification translation. Through this methodology, it is possible to obtain a set of practically correct circuit level specifications of the component blocks of a system through a fast exploration process in a single pass.

6.2 Directions for Further Research

There are a number of areas of further research works. Some of the topics closely related to the present research work includes:

1. Static sampling method has been used for data generation in chapter 3 of the dissertation. In dynamic sampling, the sampling is controlled by a so called 'learning machine', which determines the new feature vectors to be sampled based on existing instances. This process is sometimes called adaptive sampling or active learning. An active learning scheme intelligently samples the design space so that fewer design instances are needed compared to the static sampling scheme in order to obtain a model of same accuracy.

2. The LS-SVM technique involves solution of linear matrix equations. For models with large number of training data, the complexity of the solution procedure using direct method often becomes high and the accuracy is also not good. Efficient iterative algorithms, such as Krylov subspace and Conjugate Gradient (CG) method needs to be used in such cases. The dominant specification parameters have been identified using designer's knowledge. This can be implemented using data mining techniques.

3. The topology exploration using state space models used in chapter 4 of the dissertation do not consider the circuit-level non idealities of the component blocks. These have been considered through behavioral simulation process. The circuit-level non-idealities may be incorporated within the exploration process so that a topology is generated which is optimized for the non-idealities. This would avoid the time consuming behavioral simulation-based process.

4. The top-down topology generation methodology discussed in chapter 4 of the dissertation is currently limited to linear analog systems. This needs to be extended to non linear analog systems.

5. The design centering problem discussed in chapter 5 of the dissertation is based on simple heuristic technique. A more formal treatment of the design centering is required. This is an important research topic in a robust analog design automation process.

6. The specification translation methodology as discussed in chapter 5 of the dissertation determined only nominal values. A more practical output would be a range f values for each specification parameters so that within the ranges the desired specifications of the system are satisfied within acceptable error. This will give the circuit designers some amount of relaxation while implementing the component blocks at the circuit level of abstraction.

Appendix A

Appendix: Least Squares Support Vector Machine

Support Vector Machines (SVM) were first proposed in the year 1995 to solve machine learning problems [86]. Traditional neural network approaches have suffered difficulties with generalisation, producing models that can overfit the data. This is a consequence of the optimisation algorithms used for parameter selection and the statistical measures used to select the 'best' model. SVM's are based on the structural risk minimization (SRM) principle, which has been shown to be superior [87], to traditional empirical risk minimisation (ERM) principle, employed by conventional neural networks. SRM minimises an upper bound on the expected risk, as opposed to ERM that minimises the error on the training data. It is this difference which equips SVM with a greater ability to generalize, which is the goal in statistical learning.

SVMs were originally developed to solve the classification problem, but recently they have been extended to the domain of regression problems [60]. In the literature the terminology for SVMs is slightly confusing. The term SVM is typically used to describe classification with support vector methods and support vector regression is used to describe regression with support vector methods. In this dissertation the term SVM will refer to both classification and regression methods, and the terms Support Vector Classification (SVC) and Support Vector Regression (SVR) will be used for support vector machine based classification and regression respectively. A modified version of SVM techniques, referred to as least squares SVM (LS-SVM) has been proposed by Suykens et.al [58]. LS-SVM technique simplifies traditional

SVM technique to some extent. In our work, we use LS-SVM technique. In the following sections, we discuss in detail least squares support vector regression and classification respectively.

A.1 Least-Squares Support Vector Regression

Consider a given set of training samples $\{x_k, y_k\}_{k=1,2,...,n}$ where x_k is the input value and y_k is the corresponding target value for the k^{th} sample. With a SVR, the relationship between the input vector and the target vector is given as

$$\hat{\bar{y}}(x) = w^T \phi(\bar{x}) + b \tag{A.1}$$

where ϕ is the mapping of the vector $\bar{x}$ to some (probably high-dimensional) feature space, b is the bias and w is the weight vector of the same dimension as the feature space. The mapping $\phi(\bar{x})$ is generally nonlinear which makes it possible to approximate nonlinear functions. The approximation error for the k^{th} sample is defined as

$$e_k = y_k - \hat{y}_k(x_k) \tag{A.2}$$

For a given data, the weights which give smallest summed quadratic error of the training samples are determined. Because this can easily lead to overfitting, ridge regression (a form of regression) is used to smoothen the approximation. The minimization of the error together with the regression is given as

$$min \;\; \mathcal{J}(w, e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=0}^{n} e_k^2 \tag{A.3}$$

with equality constraint

$$y_k = w^T \phi(x_k) + b + e_k, \;\; k = 1, 2, ..., n \tag{A.4}$$

where γ is the regularization parameter. The first term of the cost function (A.3) is a so called L_2 norm on the regression weights. Using this norm, the weight values are penalized quadratically. The second term takes into account the regression error for all the samples. The relative weight of this term compared to the first term is defined by the regularization parameter, which has to be optimized by the model

developer.

Traditional SVM approach defines the regression error differently by neglecting all regression errors smaller than $\pm\epsilon$ (the ϵ insensitive loss function). It is this difference in error definition that makes the LS-SVM optimization problem computationally much easier than the original SVM problem. Furthermore, the value of parameter ϵ does not have to be optimized for LS-SVM, which is the case for SVMs.

Similar to SVM, the LS-SVM also considers this optimization problem (A.3) to be a constrained optimization problem and uses a Lagrange function to solve it. Instead of minimizing the primary objective (A.3), a dual objective, the so-called Lagrangian, is formed of which the saddle point is the optimum. The Lagrangian for this problem is given as

$$\mathcal{L}(w, b, e, \alpha) = \mathcal{J}(w, e) - \sum_{k=0}^{n} \alpha_k \left(w^T \phi(x_k) + b + e_k - y_k \right) \tag{A.5}$$

where α_ks' are called the Lagrangian multipliers. The saddle point is found out by setting the derivatives equal to zero:

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \rightarrow w = \sum_{k=0}^{n} \alpha_k \phi(x_k) \tag{A.6}$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \rightarrow w = \sum_{k=0}^{n} \alpha_k = 0 \tag{A.7}$$

$$\frac{\partial \mathcal{L}}{\partial e_k} = 0 \rightarrow \alpha_k = \gamma e_k \tag{A.8}$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_k} = 0 \rightarrow w^T \phi(x_k) + b + e_k - y_k = 0 \tag{A.9}$$

By eliminating e_k and w through substitution, the final model is expressed as a weighted linear combination of the inner product between the training points and a

new test object. The output is given as

$$\begin{aligned}\hat{\bar{y}}(\bar{x}) &= \langle w, \phi(\bar{x})\rangle && \text{(A.10)}\\ &= \left\langle \sum_{k=1}^{n} \alpha_k \phi(x_k), \phi(x) \right\rangle + b && \text{(A.11)}\\ &= \sum_{k=1}^{n} \alpha_k \langle \phi(x_k), \phi(x)\rangle + b && \text{(A.12)}\\ &= \sum_{k=1}^{n} \alpha_k K(x_k, x) + b && \text{(A.13)}\end{aligned}$$

where $K(x_k, x)$ is the kernel function. The elegance of using the kernel function lies in the fact that fact that one can deal with feature spaces of arbitrary dimensionality without having to compute the map $\phi(\bar{x})$ explicitly. Any function that satisfies Mercers condition can be used as the kernel function. The Gaussian kernel function defined as

$$K(x_k, x) = exp\left(-||x_k - x|\,|^2/\sigma^2\right) \qquad \text{(A.14)}$$

is commonly used, where σ^2 denotes the kernel bandwidth.

The two hyperparameters, namely the regularization parameter γ and the kernel bandwidth σ^2 have to be tuned by the model developers. These can be optimized by the use of Vapnik-Chervonenkis bound, k-fold cross validation technique or Bayesian learning. The optimal parameters are found through grid search technique. The result of this grid-search is an error surface spanned by the hyper parameters. A robust model is obtained by selecting those parameters that gives the lowest error in a smooth area based on 10-fold cross validation of the training set. The mean square error is used as the error function.

The entire LS-SVR technique is available in a Matlab toolbox *lssvmlab* [61] developed by the authors of [58]. This has been extensively utilized in this work.

A.2 Least-Squares Support Vector Classification

The classification problem is restricted to consideration of the two-class problem without loss of generality. In this problem the goal is to separate the two classes by a function which is induced from available examples. The goal is to produce a classifier that will work well on unseen examples, i.e. it generalizes well.

We consider each of n data points $x_k \in \Re^p, k = 1, 2, ..., n$ to be associated with a label $y_k \in \{1, 0\}$ which classifies the data into one of the two sets. The SVM classifier according to Vapnik's original formulation satisfies the condition

$$y_k \left[w^T \varphi(x_k) + b\right] \geq 1 \quad k = 1, 2...., n \tag{A.15}$$

The nonlinear function $\varphi(.) : \Re^p \rightarrow \Re^{p_h}$ maps the input space to a high (and possibly infinite) dimensional feature space. In primal weight space, the classifier then takes the form

$$y(x) = sign \left[w^T \varphi(x) + b\right] \tag{A.16}$$

but, on the other hand, is never evaluated in this form. Vapnik's classifier formulation was modified in [58] into the an LS-SVC formulation given by (A.3) subject to the equality constraints

$$y_k \left[w^T \varphi(x_k) + b\right] = 1 - e_k \quad k = 1, 2, ..n \tag{A.17}$$

This formulation consists of equality instead of inequality constraints and takes into account a squared error with regularization term similar to ridge regression. The optimization problem is solved through Lagrange multiplier technique. The Lagrangian is given by

$$\mathcal{L}(w, b, e, \alpha) = \mathcal{J}(w, b, e) - \sum_{k=1}^{n} \alpha_k \left\{y_k \left[w^T \varphi(x_k) + b\right] - 1 + e_k\right\} \tag{A.18}$$

where $\alpha_k \in \Re$ are the Lagrange multipliers that can be positive or negative in the LS-SVC formulation. Following similar techniques as employed in LS-SVR construction, the final LS-SVC is given by

$$y(x) = sign \left[\sum_{k=1}^{n} \alpha_k y_k K(x, x_k) + b\right] \tag{A.19}$$

where $K(x, x_k)$ is the kernel function.

Appendix B

Global Optimization Techniques

Computer aided analog design problems are generally expressed as computing global optimal solutions of an optimization problem [3]. Classical nonlinear programming techniques fail to solve such problems because these problems usually contain multiple local optima. Therefore, global search methods are invoked in order to deal with such problems.

In this dissertation constrained global optimization problems are considered. Without loss of generality, only minimization problems are studied since maximization problems can be transformed to minimization problems by inverting the sign of their objective functions. The mathematical definition for the considered problems is given below: In this dissertation we use metaheuristic techniques to solve constraint global optimization problems. Metaheuristics contain all heuristics methods that show evidence of achieving good quality solutions for the problem of interest within an acceptable time. Usually, metaheuristics offer no guarantee of obtaining the global solutions. Metaheuristics are classified into two broad classes; point- to-point methods and population-based methods. In the former methods, the search invokes only one solution at the end of each iteration from which the search will start in the next iteration. On the other hand, the population-based methods invoke a set of many solutions at the end of each iteration. Below, we highlight the principles of genetic algorithm as an example of population-based methods, and simulated annealing as an example of point-to-point methods.

B.1 Genetic Algorithm

A Genetic Algorithm (GA) is a search based optimization method that draws inspiration from natural selection and survival of the fittest in the biological world. GA falls into the more wider category of search methods known as Evolutionary algorithms (EAs). GA starts with an initial population whose elements are called *chromosomes*. The chromosome consists of a fixed number of variables which are called *genes*. In order to evaluates and rank chromosomes in a population, a *fitness function* based on the objective function is defined. A set of three operators are specified to construct the complete structure of a GA procedure. These are *selection/reproduction*, *crossover* and *mutation* operators. The selection operator selects an intermediate population from the current one in order to be used by the other operators; crossover and mutation. In this selection process, chromosomes with higher fitness function values have a greater chance to be chosen than those with lower fitness function values. Crossover defines how the selected chromosomes (parents) are recombined to create new structures (offspring) for possible inclusion in the population. Mutation is a random modification of a randomly selected chromosome. Its function is to guarantee the possibility of exploring the space of solutions for any initial population and to permit the escape from a zone of local minimum. GA operators; selection, crossover and mutation have been extensively studied. Many effective setting of these operators have been proposed to fit a wide variety of problems. More details about GA elements are discussed below before stating a standard GA procedure.

1. *Fitness Function:* Fitness function F is a designed function that measures the goodness of a solution. It is designed in a way that better solutions have a higher fitness function value than worse solutions. The fitness function plays a major role in the selection process.

2. *Coding:* Coding in GA is the form in which chromosomes and genes are expressed. There are mainly two types of coding; binary and real. Binary GA requires the solutions to be coded as finite-length binary strings of 1's and 0's. This is naturally suited to combinatorial optimization problems with discrete search spaces. In real-parameter GA, the solutions are represented as direct real numbers. Binary GA presents a number of difficulties like Hamming cliffs and inability to achieve any arbitrary precision when applied to problems with

continuous search spaces. To avoid these limitation, real-parameter GAs are developed.

3. *Selection:* Genetic Algorithm is modeled on Darwin's evolution theory of the survival of the fittest. Thus, in any generation of solutions, the best ones survive with higher probability and create offsprings. There exists a number of selection operators for reproduction in GA literature but the essential idea in all of them is that solutions are picked from the current population and their multiple copies are inserted in the mating pool in a probabilistic manner. The various methods of selecting chromosomes from the pool of parent solutions are : proportionate selection, tournament selection, rank selection etc. In our work, we use proportionate selection operator. This is the most commonly used selection method and is usually implemented with a roulette-wheel simulation. Every solution is assigned a fitness value F_i, and has a roulette-wheel slot sized in proportion to its fitness. In order to create a new population, the roulette-wheel is spun n times, each time selecting an instance of the solution chosen by the roulette wheel pointer. Thus, the probability p_i of selecting the i^{th} solution is given by

$$p_i = \frac{F_i}{\sum_{i=1}^{n} F_i} \tag{B.1}$$

4. *Crossover:* Crossover operator aims to interchange the information and genes between chromosomes. Therefore, crossover operator combines two or more parents to reproduce new children. One of these children may hopefully collect all good features that exist in his parents. Crossover operator is applied with probability p_c. In the crossover operator, uniform crossover technique is used in our work. Two arbitrary chromosomes (parents) are randomly selected from the population and their genes are rearranged at several crossover points, which are determined randomly in order to generate two new chromosomes (children).

5. *Mutation:* The mutation operator is used with a low probability p_m to alter the solutions locally to hopefully create better solutions. The need for mutation is to maintain a good diversity of the population. Although this operator performs a random change in the solution chosen for mutation, the low mutation probability ensures that the process creates only a few such solutions in the search space and the evolution does not become random.

6. *Elite-Preserving Operator:* In order to ensure that the statistics of the population-best solutions does not degrade with generations, the elite-preserving operator is often used in GAs. Typically, the best $\alpha\%$ of the population from the current population is directly copied to the next generation. The rest of the new population is created by the usual genetic operations applied on the entire current population. Thus, the best solutions of the current population not only get passed from one generation to another, but they also participate with other members of the population in creating other population members.

With these background on GA operators, we present a complete GA procedure utilizing these operators.

1. Choose a coding to represent problem parameters, a selection operator, a crossover operator and a mutation operator. Choose a population size n, crossover probability p_c, and mutation probability p_m. Initialise a random population of chromosomes of size l. Choose a maximum allowable generation number t_{max}. Set $t = 0$.

2. Evaluate each chromosome in the population.

3. If $t > t_{max}$ or other termination criteria is satisfied, Terminate.

4. Perform reproduction on the population.

5. Perform crossover on random pairs of chromosomes.

6. Perform mutation on every chromosome.

7. Evaluate chromosomes in the new population. Set $t = t + 1$ and go to step 3.

The algorithm is straightforward with repeated application of three operators (Steps 4 to 7) to a population of points.

B.2 Simulated Annealing

The simulated annealing procedure simulates the annealing process to achieve the minimum function value in a minimization problem. The SA algorithm successively generates a trial point in a neighbourhood of the current solution and determines

whether or not the current solution is replaced by the trial point based on a probability depending on the difference between their function values. Convergence to an optimal solution is theoretically guaranteed only after an infinite number of iterations controlled by the procedure so-called cooling schedule. The main control parameter in the cooling schedule is the temperature parameter T. The main role of T is to let the probability of accepting a new move be close to 1 in the earlier stage of the search and to let it be almost zero in the final stage of the search. A proper cooling schedule is needed in the finite-time implementation of SA to simulate the asymptotic convergence behavior of the SA.

According to the Boltzmann probability distribution, a system at thermal equilibrium at a temperature T has its energy distributed probabilistically according to $P(E) = exp(-\Delta E/kT)$, where k is the Boltzmann constant. This expression suggests that a system at a high temperature has an almost uniform probability of being at any energy state, but at a low temperature it has a small probability of being at a high energy state. Therefore, by controlling the temperature T and assuming that the search process follows the Boltzmann probability distribution, the convergence of an algorithm is controlled.

The SA algorithm is stated as follows:

1. Choose an initial point x^1, a termination criteria ϵ. Set T a sufficiently high value, number of iterations performed at a particular temperature be n, and set t = 0.

2. Calculate a neighbouring point x^2. Usually, a random point in the neighbourhood is created.

3. If $\Delta E = E(x^2) - E(x^1) < 0$, set $t = t + 1$;
   Else create a random number r in the range $(0, 1)$. If $r \leq exp(-\Delta E/T)$ set $t = t + 1$;
   Else go to step 2.

4. If $|x^2 - x^1| < \epsilon$ and T is small, terminate
   Else if $(t \ mod \ n) = 0$ then lower T according to a cooling schedule.
   Go to step 2;

5. Else go to step 2.

One of the most powerful features of SA is its ability of easily escaping from being trapped in local minima by accepting up-hill moves through a probabilistic procedure especially in the earlier stages of the search. On the other hand, the main drawbacks that have been noticed on SA are its suffering from slow convergence and its wandering around the optimal solution if high accuracy is needed.

Bibliography

[1] S. Balkir, G. Dündar, and A.S. Öğrenci. *Analog VLSI Design Automation.* CRC Press, 2003.

[2] G.V. Plas, G. Gielen, and W. Sansen. *A Computer-Aided Design and Synthesis Environment for Analog Integrated Circuits.* Kluwer Academic Publishers, 2003.

[3] Georges.G.E. Gielen and Rob.A. Rutenbar. Computer-Aided Design of Analog and Mixed-Signal Integrated Circuits. *Proceedings of the IEEE*, Vol.88:pp.1825–1852, December 2000.

[4] S. Donnay. *Analog High-Level Design Automation in Mixed-Signal ASICs.* PhD thesis, KULeuven, 1998.

[5] Georges.G.E. Gielen. Modeling and Analysis Techniques for System-Level Architectural Design of Telecom Front-Ends. *IEEE. Trans. Microwave Theory and Techniques*, Vol.50:pp.360–368, January 2002.

[6] A. Nunez and R Vemuri. An Analog Performance Estimation for Impriving the Effectiveness of CMOS Analog Systems Circuit Synthesis. In *Proc. of DATE*, pages 406–411, 1999.

[7] A. Doboli, N. Dhanwada, A. Nunez-Aldana, and R. Vemuri. A Two-Layer Library-Based Approach to Synthesis of Analog Systems from VHDL-AMS Specifications. *ACM Trans. DAOES*, Vol.9:pp.238–271, April 2004.

[8] E. Lauwers and Georges Gielen. Power Estimation Methods for Analog Circuits for Architectural Exploration of Integrated Systems. *IEEE. Trans. VLSI Systems*, Vol.10:pp.155–162, April 2002.

[9] M.M Hershenson, S.P. Boyd, and T.H. Lee. Optimal Design of a CMOS Op-amp via Geometric Programming. *IEEE Trans. CADICS*, Vol.20:pp.1–21, January 2001.

[10] P. Mandal and V. Visvanathan. CMOS Op-amp Sizing Using a Geometric Programming Formulation. *IEEE Trans. CADICS*, Vol.20:pp.22–38, January 2001.

[11] W. Daems, G. Gielen, and W. Sansen. Simulation-Based Generation of Posynomial Performance Models for the Sizing of Analog Integrated Circuits. *IEEE Trans. CADICS*, Vol.22:pp.517–534, May 2003.

[12] X. Li, P. Gopalkrishnan, Y. Xu, and L.T. Pileggi. Robust Analog/RF Circuit Design With Projection-Based Performance Modeling. *IEEE Trans. CADICS*, Vol.26:pp.2–15, January 2007.

[13] G.V. Plas, J. Vandenbussche, G. Gielen, and W. Sansen. EsteMate: A tool for Automated Power and Area estimation in Analog Top-down Design and Synthesis. In *Proc. of CICC*, pages 139–142, May 1997.

[14] X. Ren and Kazmierski. T. Performance Modeling and Optimization of RF Circuits using Support Vector Machines. In *MIXDES*, pages 317–321, 2007.

[15] T. Kiely and G. Gielen. Performance Modeling of Analog Integrated Circuits using Least-Squares Support Vector Machines. In *Proc. of DATE*, 2004.

[16] M. Ding and R. Vemuri. A Combined Feasibility and Performance Macromodel for Analog Circuits. In *Proc. of DAC*, pages 63–68, June 2005.

[17] E. Martens and G. Gielen. Classification of analog synthesis tools based on their architecture selection mechanisms. *Integration*, 41:238–252, 2008.

[18] G.V. Plas, G. Debyser, F. Leyn, K. Lampaert, J. Vandenbussche, G.G.E. Gielen, W. Sansen, P. Veselinovic, and D. Leenarts. AMGIE-A Synthesis Environment for CMOS Analog Integrated Circuits. *IEEE Trans. CADICS*, Vol.20:pp.1037–1058, 2001.

[19] F. Medeiro, A. Verdu, and A. Vazquez. *Top-down Design of High-Performance Delta Sigma Modulators*. Kluwer Academic Publishers, 1999.

[20] J. Crols, S. Donnay, M. Steyaert, and G. Gielen. A High-Level Design and Optimization Tool for Analog and RF Receiver Front-Ends. In *Proc. of ICCAD*, pages 550–553, 1995.

[21] K. Francken and Georges.G.E. Gielen. A High-Level Simulation and Synthesis Environment for Sigma-Delta Modulators. *IEEE Trans. CADICS*, Vol.22:pp.1049–1061, August 2003.

[22] Jesús Ruiz-Amaya *et.al.,*. High-Level Synthesis of Switched-Capacitor, Switched-Current and Continuous -Time Sigma Delta Modulator Using SIMULINK-Based Time-Domain Behavioral Models. *IEEE Trans. CAS -1*, Vol.52:pp.1795 –1810, September 2005.

[23] P.C. Maulik, L.R. carley, and R.A. Rutenbar. Integer Programming Based Topology Selection of Cell-Level Analog Circuits. *IEEE Trans. CADICS*, 14:401–412, April 1995.

[24] H. Tang and A. Doboli. High-Level Synthesis of Delta Sigma Modulator Topologies Optimized for Complexity, Sensitivity, and Power Consumption. *IEEE Trans. CADICS*, Vol.25:pp.597–607, March 2006.

[25] A. Doboli and R. Vemuri. Exploration-Based High-Level Synthesis of Linear Analog Systems Operating at Low/Medium Frequencies. *IEEE Trans. CADICS*, Vol.22:pp.1556–1567, November 2003.

[26] E. Martens and Georges Gielen. Top-Down Heterogeneous Synthesis of Analog and Mixed-Signal Systems. In *Proc. of DATE*, pages 275–280, March 2006.

[27] B.A.A. Antao and A.J. Broderson. ARCHGEN:Automated Synthesis of Analog Systems. *IEEE Trans. VLSI Systems*, Vol.3:pp.231–244, June 1995.

[28] R. Harjani and J Shao. Feasibility and Performance Region Modeling of Analog and Digital Circuits. *Analog Integrated Circuits and Signal Processing*, Vol.10:pp.23–43, August 1996.

[29] G. Stehr, H. Graeb, and K. Antreich. Analog Performance Space Exploration by Normal-Boundary Intersection and by Fourier-Motzkin Elimination. *IEEE Trans. CADICS*, Vol.26:pp.1733–1748, Oct. 2007.

[30] G. Stehr, H. Graeb, and K. Antreich. Performance Trade-Off Analysis of Analog Circuits By Normal-Boundary Intersection. In *Proc. of DAC*, pages 958–963, 2003.

[31] G. Stehr, H. Graeb, and K. Antreich. Analog Performance Space Exploration by Fourier-Motzkin Elimination with Application to Hierarchical Sizing. In *Proc. of ICCAD*, pages 847–854, November 2004.

[32] F.De. Bernardinis, M.I. Jordan, and A. Sangiovanni Vincentelli. Support Vector Machines for Analog Circuit Performance Representation. In *Proc. of DAC*, June 2003.

[33] N. Dhanwada, A. Doboli, A. Nunez-Aldana, and R. Vemuri. Hierarchical constraint transformation based on genetic optimization for analog system synthesis. *Integration*, Vol.39:pp.267–290, June 2006.

[34] J. Zou, D. Mueller, H. Graeb, and U. Schlichtmann. A CPLL Hierarchical Optimization Methodology Considering Jitter, Power and Locking Time. In *Proc. of DAC*, pages 19–24, July 2006.

[35] F. Bernardinis, P. Nuzzo, and A. Sangiovanni Vincentelli. Mixed Signal Design Space Exploration through Analog Platforms. In *Proc. of DAC*, pages 875–880, CA, USA, June 2005.

[36] F. Medeiro, B. Perez-Verdu, A. Rodriguez-Vazquez, and J.L. Huertas. A Vertically Integrated Tool for Automated Design of Sigma-Delta Modulators. *IEEE J. of Solid State Circuits.*, Vol.30:pp.762–772, July 1995.

[37] P. Vanassche, G. Gielen, and W. Sansen. *Systematic Modeling And Analysis of Telecom Front-Ends And Their Building Blocks*. Springer, 2005.

[38] P. Malcovati, S. Brigati, F. Francesconi, F. Maloberti, P. Cusinato, and A. Baschirotto. Behvaioral Modeling of Switched-Capacitor Sigma Delta Modulators. *IEEE Trans. CAS-1*, Vol.50:pp.352–364, March 2003.

[39] Y. Wei, A. Doboli, and H. Tang. Systematic Methodology for Designing Reconfigurable Sigma-Delta Modulator Topologies for Multimode Communication Systems. *IEEE Transactions on CADICS*, Vol.26:pp480–496, March 2007.

[40] J. Ruiz-Amaya, J.M. de la Rosa, M. Delgado-Restituto, and A. Rodriguez-Vazquez. Behavioral Modeling, Simulation and High-Level Synthesis of Pipeline A/D Converters. In *Proc. of ISCAS*, pages 5609– 5612, May 2002.

[41] G. Manganaro, S.U. Kwak, S. Cho, and A. Pulinchery. A Behavioral Modeling Approach to the Design of a Low Jitter Clock Source. *IEEE Transactions on CAS -II*, Vol.50:pp.801–814, November 2003.

[42] G. Wolfe and R. Vemuri. Extraction and Use of Neural Network Models in Automated Synthesis of Operational Amplifiers. *IEEE Trans. CADICS*, Vol.22:pp.198–212, February 2003.

[43] G. Gielen and W. Sansen. *Symbolic Analysis for Automated Design of Analog Integrated Circuits*. Kluwer Academic Publishers, 1991.

[44] C.-J. Richard Shi and X.D. Tan. Canonical Symbolic Analysis of Large Analog Circuits with D eterminant Decision Diagrams. *IEEE Trans. CADICS*, Vol.19:pp.1–18, January 2000.

[45] G. Gielen, H. Walscharts, and W. Sansen. OPTIMAN: Analog Circuit Design Optimization based on Symbolic Simulation and Simulated Annealing. *IEEE Jr. Solid State Circuits*, Vol.25:pp.707–713, June 1990.

[46] H.Y. Koh, C.H. Sequin, and P.R. Gray. OPASYN:A Compiler for CMOS Operational Amplifiers. *IEEE Trans. CADICS*, Vol.8:pp.113–125, February 1990.

[47] Rob.A. Rutenbar, Georges.G.E. Gielen, and J.Roychowdhury. Hierarchical Modeling, Optimization, and Synthesis for System-Level Analog and RF Designs. *Proceedings of the IEEE*, Vol.95:pp.1825–1852, March 2007.

[48] J. Roychowdhury. Algorithmic Macromodeling Methods for Mixed-Signal Systems. In *Proc. of VLSIID*, pages 63–68, 2004.

[49] E. Lauwers and G. Gielen. A Power Estimation Model for High-Speed CMOS A/D Converters. In *Proc. of DATE*, pages 410–405, 1999.

[50] T. Eeckelaert, T. McConaghy, and G. Gielen. Efficient Multiobjective Synthesis of Analog Circuits using Hierarchical Pareto-Optimal Performance Hypersurfaces. In *Proc. of DATE*, pages 1070–1075, 2005.

[51] E. Ochotta, R. Rutenbar, and R. Carley. Synthesis of high performance analog circuits in ASTRX/OBLX. *IEEE Transactions on CADICS.*, Vol.15:pp.273–294, March 1996.

[52] B.De. Smedt and G.G.E. Gielen. WATSON:Design Space Boundary Exploration and Model Generation for Analog and RF IC Design. *IEEE Transactions on CADICS.*, Vol.22:pp.213–224, February 2003.

[53] I. Vassiliou, H. Chang, A. Demir, E. Charbon, P. Miliozzi, and A. Sangiovanni-Vincentelli. A Video Driver System Designed Using a Top-Down, Constraint-Driven Methodology. In *Proc. of ICCAD*, pages 463–468, November 1996.

[54] N.C. Horta and J.E. Franca. Algorithm-Driven Synthesis of Data Conversion Architectures. *IEEE Transactions on CADICS.*, Vol.16:pp.1116–11135, March 1997.

[55] G.P. Box, W.g. Hunter, and J.S. Hunter. *Statistics for Experimenters: An Introduction to Design, Analysis and Model Building.* Wiley, New York, 1978.

[56] G. Stehr, H. Graeb, and K. Antreich. Feasibility Regions and their Significance to the Hierarchical Optimization of Analog and Mixed-Signal Systems. *Int.Ser.Numer.Math*, Vol.146:pp.167–184, 2003.

[57] H. Graeb, S. Zizala, J. Eckmueller, and K. Antreich. The Sizing Rules Method for Analog Integrated Circuit Design. In *IEEE/ACM ICCAD*, pages 343–349, 2001.

[58] J.A.K Suykens, T.V Gestel, J.D Brabanter, B.D Moor, and V. Joos Vandewalle. *Least Squares Support Vector Machines.* World Scientific, 2002.

[59] S.S. Keerthi and C.J. Lin. Asymptotic behaviors of support vector machines with Gaussin kernel. In *Prof. of Nueral Computation*, pages 1667–1689, 2003.

[60] V Vapnik. *Statistical Learning Theory.* Springer, New York, 1998.

[61] LS-SVM Toolbox. http://www.esat.kuleuven.ac.be/sista/lssvmlab, February 2003.

[62] P.E Allen and D.R Holberg. *CMOS Analog Circuit Design.* Oxford University Press, 2004.

[63] J. Wu, G.K. Fedder, and L.R. Carley. A Low-Noise Low-Offset Capacitive Sensing Amplifier for a 50-$\mu g/\sqrt{Hz}$ Monolithic CMOS MEMS Accelerometer. *IEEE J. of Solid-State Circuits*, Vol.39:pp.722–730, May 2004.

[64] A.S. Jackson. *Analog Computation*. McGraw Hill, 1960.

[65] A. Marques, V. Peluso, M. Steyaert, and W. Sansen. Optimal Parameters for Delta-Sigma Modulator Topologies. *IEEE Trans. CAS-II*, Vol.45:pp.1232–1241, September 1998.

[66] R. Schreir. The Delta-Sigma Toolbox 6.0. http://www.mathworks.com/matlabcentral/fileexchange, December 2004.

[67] T. Kuo, K. Chen, and J. Chen. Automatic Coefficients Design for High-Order Sigma-Delta Modulators. *IEEE Trans. CAS-II*, Vol.46:pp.6–15, January 1999.

[68] O. Bajdechi, G. Gielen, and J. Huijsing. Systematic Design Exploration of Delta-Sigma ADCs. *IEEE Trans. CAS-1*, Vol.51:pp.86–95, January 2004.

[69] S. Pengbo, Y. Wei, and A. Doboli. Flexibility-oriented Design Methodology for Reconfigurable Delta Sigma Modulators. In *Proc. of DATE*, pages 415–420, 2007.

[70] S. Orkun, E. Yetik, S. Talay, and G. Dundar. A Coefficient Optimization and Architecture Selection Tool for SD modulators Considering Components Non-idealities. In *Proc. of GLSVLSI*, pages 423–428, 2007.

[71] S. Norsworthy, R. Schreier, and G. Temes. *Delta-Sigma Data Converters - Theory, Design and Simulation*. IEEE Press, 1997.

[72] M. Ortmanns, F. Gerfers, and Y. Manoli. A Case Study on a 2-1-1 Cascaded Continuous-Time Sigma-Delta Modulator. *IEEE trans. CAS-I*, Vol.52:pp.1515–1525, Auguest 2005.

[73] R.D. Batten, A. Eshraghi, and T.S. Fiez. Calibration of Parallel Delta Sigma ADCs. *IEEE trans. CAS-II*, Vol.49:pp.390–399, June 2002.

[74] E. Sánchez-Sinencio and J. Martinez-Silva. CMOS Transconductance amplifiers, architectures and active filters: a tutorial. *IEE Proceedings, Circuits Devices-Syst.*, Vol.147:pp.3–12, February 2000.

[75] S. Koziel and S. Szczepanski. Dynamic Range Comparison of Voltage-Mode and Current-Mode State-Space $G_m - C$ Biquad Filters in Reciprocal Structures. *IEEE trans. CAS-I*, Vol.50:pp.1245–1255, October 2003.

[76] Gilbert. Strang. *Linear Algebra and its Applications.* Harcourt Brace Jovanovich, Publishers, San Diego, 1988.

[77] M. Gevers and G. Li. *Parametrizations in Control, Estimation and Filtering Problems.* Springer-Verlag, 1993.

[78] Lothar. Thiele. On the Sensitivity of Linear State-Space Systems. *IEEE Transactions on Circuits and Systems*, Vol.33:pp.502–510, May 1986.

[79] F Gerfers, M Ortmanns, and Y Manoli. A 1.5V 12-bit Power-Efficient Continuous-Time Third-Order Sigma-Delta Modulator. *IEEE J. of Solid-State Circuits*, Vol.38:pp.1343–1352, August 2003.

[80] S.R. Kadivar, D. Schmitt-Landsiedel, and H. Klar. A new algorithm for the design of stable higher order single loop sigma delta analog-to-digital converters. In *Proc. of ICCAD*, pages 554–561, 1995.

[81] H. Aboushady, L.de. Lamarre, N. Beilleau, and Louerat. M.M. A 5mW, 100kHz Bandwidth, Current-Mode Continuous-Time Sigma-Delta Modulator with 84dB Dynamic Range. In *Proc. of ESSCIRC*, 2002.

[82] Helmut E. Graeb. *Analog Design Centering and Sizing.* Springer, 2007.

[83] E. Hansen and G.W. Walster. *Global Optimization Using Interval Analysis.* Marcel Dekker, 2004.

[84] Interval Analysis Toolbox. www.ti3.tu-harbarg.de/ rump/intlab.

[85] J.A. Cherry and W.M. Snelgrove. *Continuous-Time Delta-Sigma Modulators For High-Speed A/D Conversion.* Kluwer Academic Publishers, 2002.

[86] V Vapnik. *The nature of statistical learning theory.* Springer, New York, 1995.

[87] S.R. Gunn, M. Brown, and K.M. Bossley. Network performance assessment for neuro fuzzy data modeling. In *Intelligent Data Analysis, LNCS*, pages 313–323, 1997.

Publications out of this work

1. Soumya Pandit, S.K.Kar, C.R.Mandal and A.Patra. High Level Synthesis of Higher Order Continuous Time State Variable Filters with Minimum Sensitivity and Hardware Count, In *Proc. of IEEE DATE 2006*, pages 1203-1204,

2. Soumya Pandit, C.R.Mandal and A.Patra. A Formal Approach for High Level Synthesis of Linear Analog Systems. In *Proc. of ACM GLSVLSI 2006,* pages 345-348

3. Soumya Pandit, S.K.Bhattacharya, C.R.Mandal and A.Patra. A Hybrid Search procedure for System Level Analog Design Space Exploration Used in High Level Synthesis of Analog Systems. In *Proc. of International Conference on Computers and Devices for Communication' 2006* pages 114-117

4. Soumya Pandit, C.R.Mandal and A.Patra. High Level Synthesis of Linear Analog Systems, In *Proc. of EAIT 2005, Elsevier* pages 389-392

5. Soumya Pandit, S.K.Bhattacharya, C.R.Mandal and A.Patra. A Fast Exploration Procedure for Analog High-Level Specification Translation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.* (submitted after second stage revision)

6. Soumya Pandit, C.R.Mandal and A.Patra. Systematic Procedure for High-Level Design of Capacitive Sensor System Front-Ends *IET Journal Circuits, Devices and Systems* (under revision)

BIO-DATA

Name:	**Soumya Pandit**
Sex:	Male
Date of Birth:	09.07.1976
Father's Name:	Sujit Kumar Pandit
Mother's Name:	Mukul Pandit
Nationality:	Indian
Permanent Address:	168, Rajdanga School Road P.O. - East Kolkata Township Kolkata - 700107 India

Educational Qualifications:

Sl No.	University/Institute	Degree	Year
1	University of Calcutta	B.Sc (Physics Hons)	1998
2	University of Calcutta	M.Sc (Electronic Science)	2000
3	University of Calcutta	M.Tech (Radio Physics & Electronics)	2002
4	Indian Institute of Technology Kharagpur	Ph.D.	Continuing