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 to the
Indian Institute of Technology, Kharagpur
For award of the degree
of

Doctor of Philosophy (Ph.D.)

by

Soumya Pandit

School of Information Technology
Indian Institute of Technology, Kharagpur

May 2009

To my parents for their constant support throughout my life.

To Srabanti, the love of my life.

Certificate of Approval

*Certified that the thesis entitled "**An Optimization-based Methodology for High-Level Design of Analog Systems**" submitted by **Soumya Pandit** to the Indian Institute of Technology, Kharagpur for the award of the degree of Doctor of Philosophy, has been accepted by the external examiners and that the student has successfully defended the thesis in the viva-voce examination held today.*

Prof. Chittaranjan Mandal
(Supervisor)

Prof. Amit Patra
(Co-Supervisor)

Prof. Indranil Sengupta
(Chairman of DSC)

Prof. G.Panda
(External Examiner)

Prof. Shamik Shural
(Member of DSC)

Prof. Siddhartha Sen
(Member of DSC)

Declaration

I, the undersigned, hereby certify the following:

1. *the work contained in this thesis is original and has been done by me under the guidance of my supervisor.*
2. *the work has not been submitted to any other Institute for any degree or diploma.*
3. *I have followed the guidelines provided by the Institute in preparing the thesis.*
4. *I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.*
5. *whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.*

Dated: 18th May 2009
Kharagpur,

Soumya Pandit
Research Scholar,
School of Information Technology
Indian Institute of Technology
Kharagpur -721 302, INDIA

Certificate

*This is to certify that the thesis entitled “**An Optimization-based Methodology for High-Level Design of Analog Systems**” submitted by **Soumya Pandit** to the Indian Institute of Technology, Kharagpur for the award of the degree of Doctor of Philosophy, is a record of bona fide research work carried out by him under our supervision and guidance and is worthy of consideration for the award of the degree of Doctor of Philosophy of the Institute*

Chittaranjan Mandal
Associate Professor,
School of Information Technology &
Computer Science & Engineering
Indian Institute of Technology
Kharagpur -721 302, INDIA

Amit Patra
Professor,
Department of Electrical Engineering
Indian Institute of Technology
Kharagpur -721 302, INDIA

Acknowledgments

Writing this part of the thesis is probably the hardest. Though the list of people to thank is long, making this list is not the hard part. The hard part is finding the words that convey the sincerity and magnitude of my gratitude. People who were unknown to me a few years back, have now become a large part of my life, while people I already knew have remained pillars of constant support and encouragement through these long years. It is with all this help that I stand where I am.

My supervisor, Prof. Chittaranjan Mandal, has been primarily responsible for the freedom and timely guidance which I enjoyed during my doctoral studies. I shall be sincerely indebted to him for this. Prof. Amit Patra, my co-supervisor, has given me a lot of invaluable feedback on my research and I would like to thank him for that. I shall remain grateful to him for giving me chance to work in his consultancy project and for his partial financial support of the present research work. Prof. Shamik Sural has been a wonderful teacher and I am grateful to him for being a member of my doctoral committee. Prof. Indranil Sengupta, Head of SIT and one of my doctoral committee members has been very generous with his time on my seminars and I am indeed very thankful to him. Prof. Siddartha Sen, doctoral committee member has been exceptionally helpful in my stay at IIT Kharagpur, during which his door has always been open for me. I really enjoyed the working in his consultancy project, which is the primary source of funding of the present research work. I express my heartiest thanks to Sumit Kr. Bhattacharya, an M.Tech student of Electrical Engineering department for his significant technical contribution to this work. Thanks are due to Advanced VLSI Design Laboratory (AVDL), IIT Kharagpur and Department of Information Technology, Govt. of India for partial financial support to this project.

The AVDL members and facility staffs deserve a lot of thanks for helping me: Ashudeb, KCR, Sougata, Swamiji, Santosh, Subhashis, Anirban, Samiran, Uttam-da, Joydeep and others have all contributed in no small amount to this project. The cooperation of all the SIT scholars: Somnath, Debashis, Misraji, Rajkumar, Ashalatha, Mandal-da, and Doctor-da is much appreciated. I express my thanks to

all the SIT staff members: Mithun, Soumitri, Somadi, Samarda, Dasbabu, Malayda, Vinod and Raybabu.

The largest part of my campus life was dominated by my friends whom I worked with and learned from. I will begin with Mandal-da (Sushanta Mandal), who was such a source of knowledge and companionship that one cannot really hope for any better. Santosh, Sougata, Somnath and KCR have been wonderful friends on this journey.

I would not have been sitting here writing this acknowledgment of my PhD dissertation, were it not for my parents (*Baba and Ma*). Thank you for all the sacrifices you have made for me, thank you for your faith and confidence in me, and thank you for always supporting me in every endeavor. Your unending passion for work are truly inspiring to me. I owe great thanks to my brothers (*Borda and Chhorda*) for all things that they gave me or taught me. Without their support I would never have made any success. Words are inadequate to express my gratitude to my wife, Srabanti for being by my side through this roller-coaster ride, for having faith in me and for being my strength when everything seemed to be working against me, for celebrating my successes as if they were hers and for helping me emerge from my failures. I especially thank her for being so patient and taking a lot of pain in going through the entire thesis and providing me useful suggestions for technical improvement of the thesis. She has been my greatest source of strength and inspiration right from my undergraduate courses. I owe this degree most to her. Special thanks to my in-laws for being so supportive and understanding. I am grateful to them for their constant words of encouragement. Thanks are also due to all of my other family members, whose name I could not specify in this short space. Above all, to the One for always guiding me from within!

Soumya Pandit

Abstract

Analog high-level design is rapidly becoming a main topic of interest in an analog design automation process because of its promise to deliver short design closure at much lesser costs. This has motivated 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 along with a set of geometry constraints applied over them define the sample space. A Halton sequence generator is used as a sampling algorithm. Performance data are generated by simulating each sampled circuit configuration through SPICE. Least squares support vector machine (LS-SVM) is used as a regression function. Optimal values of the model hyper parameters are determined through a grid search-based technique and a genetic algorithm (GA)-based technique. The generalization ability of the models is estimated through a hold-out method and a k-fold cross validation method. The constructed performance models have been used to implement a GA-based topology sizing process. The advantages of the present methodology are that the constructed models are accurate with respect to real circuit-level simulation results, fast to evaluate and have a good generalization ability. In addition, the model construction time is low and the construction process does not require any detailed knowledge of circuit design. The entire methodology has been demonstrated with a set of experimental results.

This thesis then presents a top-down methodology for the generation of an optimal functional and component-level topology for linear analog systems, starting from a transfer function model of the system. The given transfer functions are converted to state space model which acts as the basis for generation of topologies of the system. The topology exploration process is modeled as a state space matrix exploration process. Similarity transformation matrix is used as a topology transformation operator. The newly generated topologies have the same behavioral properties, but different performance properties. A simulated annealing-based optimization procedure determines an optimal state space model based on the performances of the models. The optimized state space model is realized by functional component blocks to generate an optimal functional topology, which is subsequently realized by appropriate analog component blocks to generate an optimal component-level topology of the system. As a case study, the thesis presents a methodology for generation of an operational transconductance amplifier (OTA)-capacitor (C) based optimal topology for a 3^{rd} order and a 4^{th} order continuous-time $\Sigma\Delta$ modulator. It is found from the experimental results that the generated topologies are better in performances when compared to that of the commonly used topologies and satisfy the desired dynamic range specifications even under circuit-level non-ideal conditions. The advantage of the methodology is that the designer is able to specify the design goal and the desired specifications of a system at a higher level of abstraction in terms of transfer functions, while the design tool is able to generate an optimal component-level topology of the system directly from the transfer functions in a highly automated manner.

The thesis finally presents a design space exploration procedure for determining the specification parameters of the component blocks of the generated topology such that the desired specifications of the system are satisfied with optimized performances. A meet-in-the-middle approach is followed for constructing the feasible design space. It 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 interval analysis techniques and the latter, via a bottom-up procedure through actual circuit simulations. Least squares support vector machine-based classification principle is used to accurately identify the actual geometry of the feasible design space. Genetic algorithm is used for exploring the feasible design space. The reduced design space speeds up the exploration process. The cost function is computed through behavioral simulation of the entire system and by evaluating the high-level performance models of the component blocks. The final solution point is kept away from the feasible design space boundary, in order to increase the tolerance of the component specifications. The benefit of the methodology is that it is able to obtain practically correct circuit-level specifications of the component blocks of the system through a fast exploration process in a single pass.

The effectiveness of the procedure has been demonstrated by a set of experimental results.

These methodologies form the core of a semi-automated tool for analog high-level design. The methodologies have been implemented under Matlab-Simulink environment. For demonstration of the methodologies, two case studies were chosen: an interface electronics system for MEMS capacitive accelerometer sensor and a 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 based on the derived specifications and are simulated with SPICE. The SPICE simulation results satisfy the desired specifications of the system and match closely with the predicted results. These validate the entire procedure. The entire high-level design process for the interface electronics system is completed in less than ten minutes time and for the $\Sigma\Delta$ modulator system, the design process is completed within an hour. Thus the present methodologies make the high-level design process fast and accurate. In addition, the methodologies can be followed by users with less design experience.

Contents

1 Introduction **1**

1.1 Literature Survey and Motivation . . . 3

1.1.1 Analog Performance Modeling . . . 4

1.1.2 Generation/Selection of an Optimal System Topology . . . 5

1.1.3 High-Level Specification Translation . . . 7

1.2 Overview and Contributions of the Thesis . . . 8

1.2.1 Problem Definition . . . 8

1.2.2 Generation of High-Level Performance Models . . . 8

1.2.3 Top-Down Methodology for Generation of an Optimal Topology for Linear Analog Systems . . . 9

1.2.4 High-Level Specification Translation . . . 11

1.2.5 Contributions . . . 12

1.3 Organization of the rest of the Thesis . . . 12

2 Optimization-based Analog High-Level Design Methodology **15**

2.1 Generic Methodology . . . 16

2.1.1 Simulation-based Approach . . . 16

2.1.2 Equation-based Approach . . . 18

2.2 High-Level Model Generation . . . 19

2.2.1 Behavioral Model Generation . . . 19

2.2.1.1 Analytical Techniques . . . 20

2.2.1.2 Fitting or Regression Methods . . . 22

2.2.1.3 Symbolic Model Generation Methods . . . 23

2.2.1.4 Model Order Reduction Methods . . . 24

2.2.2 Performance Estimation Model Generation . . . 24

2.2.2.1 Bottom-Up Approach . . . 24

2.2.2.2 Top-Down Approach 25
2.2.3 Feasibility Model Generation 26
2.3 Optimization Methods 26
2.3.1 Deterministic Methods 27
2.3.2 Stochastic Methods 27
2.3.3 Multi-Objective Optimization Method 28
2.4 Topology Generation/Selection Methods 30
2.4.1 Selection before or after sizing 30
2.4.2 Selection during sizing 31
2.4.3 Top-Down Generation 31
2.5 Summary 32

3 Generation of High-Level Performance Estimation Models 33
3.1 High-Level Performance Estimation Models 33
3.2 Regression-based Model Generation 35
3.2.1 Sample Space and Design of Experiments 35
3.2.2 Training Data Generation and Scaling 37
3.2.3 Regression Using LS-SVM 40
3.2.3.1 Selection of Hyper parameters 41
3.2.4 Quality Measures 45
3.3 Comparison with Existing Methodologies 45
3.4 Topology Sizing Methodology using GA 47
3.5 Experimental Results 47
3.5.1 Experiment 1 49
3.5.2 Experiment 2 53
3.5.3 Experiment 3 53
3.6 Conclusion 60

4 Top-Down Generation of an Optimal Topology 63
4.1 Top-Down Topology Generation Methodology 64
4.1.1 Outline of the Methodology 64
4.1.2 Transfer Function and State Space Representation 66
4.1.3 Topology Generation 67
4.1.3.1 Functional Topology Generation 68
4.1.3.2 Component-Level Topology Generation 69

4.1.4 Topology Transformation 71
4.1.5 Performance Estimation 76
4.1.5.1 Sensitivity minimization 76
4.1.5.2 Hardware Complexity 79
4.1.5.3 Relative Power Consumption 80
4.1.6 Topology Exploration and Selection 81
4.1.6.1 Feasible *Gm* Value Constraints 81
4.1.6.2 Non-overload Constraints 82
4.1.7 Complete Flow of the Topology Generation Process for CT $\Sigma\Delta$ Modulator System 82
4.1.7.1 Related Work on $\Sigma\Delta$ Modulator Synthesis 83
4.1.7.2 Methodology 85
4.2 Comparison with Existing Methodologies 85
4.3 Experimental Results 87
4.3.1 Experiment 1 87
4.3.2 Experiment 2 92
4.3.3 Experiment 3 95
4.4 Conclusion 100

5 High-Level Specification Translation 103
5.1 Problem Formulation 103
5.2 Feasible Design Space Construction 105
5.2.1 Application Bounded Space 105
5.2.2 Circuit Realizable Space 106
5.2.3 Feasible Design Space 107
5.3 Feasible Design Space Identification 108
5.3.1 Accuracy Measurement 108
5.4 Design Space Exploration 109
5.4.1 Cost Function Formulation 111
5.4.1.1 Primary Objectives 111
5.4.1.2 Secondary Objectives 111
5.4.2 Exploration Algorithm 112
5.5 Comparison with Existing Methodologies 112
5.6 Experimental Results 113
5.6.1 Experiment 1 113

5.6.2 Experiment 2 117
5.6.3 Experiment 3 119
5.7 Conclusion 122

6 Conclusion and Directions for Further Research 125
6.1 Summary and Conclusions 125
6.1.1 Contributions 127
6.1.2 Tools developed 128
6.2 Directions for Further Research 129

A Least Squares Support Vector Machine 131
A.1 Least-Squares Support Vector Regression 132
A.2 Least-Squares Support Vector Classification 134

B Global Optimization Techniques 137
B.1 Genetic Algorithm 137
B.2 Simulated Annealing 140

List of Figures

1.1 Flow of a typical analog high-level design procedure 2

2.1 Optimization-based design flow 17
2.2 Simulink based behavioral model 21

3.1 2D projection of a four dimensional sample space. 36
3.2 Non-linear relation between the sample space and the input, output space. 37
3.3 An outline of the procedure for generation of training data. 38
3.4 Outline of GA-based hyperparameter selection procedure 44
3.5 Topology sizing methodology using GA optimizer with LS-SVM model 48
3.6 PMOS OTA circuit 49
3.7 Scatter Plot of the constructed models. 52
3.8 Considered system for Experiment 3. 54
3.9 Noise as function of Gm_1 and Gm_2 58
3.10 Power as function of Gm_1 and Gm_2 58
3.11 Input parasitics as function of Gm_1 59

4.1 Top-down generation of an optimal topology for linear analog systems 65
4.2 Block diagram of a CT $\Sigma\Delta$ modulator 67
4.3 Functional topology of a 4^{th} order CT $\Sigma\Delta$ modulator. 69
4.4 Functional topology for a general n^{th} order CT $\Sigma\Delta$ modulator 70
4.5 A generic 3^{rd} order CT modulator with *GmC* loop-filter. ($Ga_{ij} \equiv Gma_{ij}$) 71
4.6 *SNR* plot for two topologies generated through topology transformation, establishing the invariant property. 75

4.7 Complete flow of the topology generation process for $\Sigma\Delta$ modulator system. 84
4.8 The generated 3^{rd} order modulator topology. 88
4.9 Characterization of the generated 3^{rd} order modulator topology. . . . 90
4.10 Chosen standard topologies of a 3^{rd} order modulator for comparison. 91
4.11 SNR/DR comparison between the generated, CIFF and DF topology. 92
4.12 Monte Carlo analysis plot for sensitivity comparison between the generated (optimized), CIFF and DF topology 93
4.13 The generated 4^{th} order modulator topology and the corresponding *SNR* plot. 97
4.14 The CIFF 4^{th} order topology. 98
4.15 Bar diagram of peak *SNR* deviation for 5% coefficient variation for the generated (optimized) and the CIFF 4^{th} order modulator topology. 99

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. 107
5.2 Feasible design space and its subspaces. 109
5.3 DSE mechanism. 110
5.4 Considered system for experimentation. 114
5.5 SPICE output amplitude with 3 mV input 117
5.6 3^{rd} order OTA-C based $\Sigma\Delta$ modulator topology 119
5.7 Comparator circuit 120
5.8 Behavioral and SPICE simulated SNR plot of the modulator. 121

List of Tables

3.1 Transistor Sizes and Feasibility Constraints for OTA 49
3.2 Grid search technique using hold out method 50
3.3 Grid search technique using 5-fold cross validation method 50
3.4 GA technique using hold out method 51
3.5 GA technique using 5-fold cross validation 51
3.6 Comparison between GA and Grid search technique for LS-SVM construction 51
3.7 Construction of Nonlinear Performance Model 53
3.8 Comparison between our methodology and EsteMate 53
3.9 Functional Specs and Design constraints 55
3.10 Transistor Sizes and Feasibility Constraints for Preamplifier 56
3.11 Accuracy of Preamplifier block 56
3.12 Synthesized Topology Parameters 60
3.13 Comparison of Predicted performances and SPICE value 60

4.1 Non-idealities of the OTA-C component blocks 74
4.2 Non-idealities considered 89
4.3 Comparision of behavior under ideal and non-ideal conditions 89
4.4 Comparison in terms of yield for coefficient variation 92
4.5 Comparison between the topologies for relative power and complexity 95
4.6 Comparision of behavior under ideal and non-ideal conditions: 4^{th} order modulator 96
4.7 Comparison in terms of yield for coefficient variation: 4^{th} order modulator 96
4.8 Comparison between the topologies for relative power and complexity 98

5.1 Application bounded constraints: Experiment 1 115

5.2 Circuit Realizable Constraints: PA block 115
5.3 SVM Performances: Experiment 1 116
5.4 Translated Specifications: Experiment 1 116
5.5 GA results: averaged over 10 runs, 3.0 GHz 512 MB RAM PIV PC . 116
5.6 End results: Experiment 1 and 2 117
5.7 Application bounded constraints: Experiment 2 118
5.8 SVM Performances: Experiment 2 118
5.9 Translated Specifications: Experiment 2 118
5.10 Feasibility constraints for OTA and Comparator: Experiment 3 . . . 120
5.11 Translated Specifications: Experiment 3 121
5.12 End results: Experiment 3 122

Abbreviations and Symbols

ADC	Analog-to-Digital Converter
AMS	Analog Mixed Signal
ANN	Artificial Neural Network
C	Capacitor
CIFF	Cascaded Integrator Feed Forward
CMOS	Complimentary Metal Oxide Semiconductor
DF	Distributed Feedback
DR	Dynamic Range
DSE	Design Space Exploration
GA	Genetic Algorithm
IC	Integrated Circuit
LS-SVM	Least Squares Support Vector Machine
MEMS	Micro Electromechanical Systems
NTF	Noise Transfer Ratio
OTA	Operational Transconductance Amplifier
PLL	Phase Locked Loop
RBF	Radial Basis Function
RF	Radio Frequency
SA	Simulated Annealing
SFG	Signal Flow Graph
SNR	Signal-to-Noise Ratio
SPICE	Simulator Program with Integrated Circuit Emphasis
SRM	Structural Risk Minimization
STF	Signal Transfer Ratio
SVM	Support Vector Machine
$\mathcal{B}$	Parameterized behavioral models

$\bar{X}$	Set/vector of design variables/parameters
$\bar{\rho}$	Set/vector of performance parameters
$\mathcal{P}$	Performance model
$\Sigma\Delta$	Sigma Delta
$\mathcal{D}_\alpha$	Circuit-level design space
$\mathcal{D}_g$	Design space
$\mathcal{D}_a$	Application-bounded specification space
$\mathcal{D}_c$	Circuit-realizable specification space
γ	Regularization parameter
σ^2	RBF kernel parameter
ARE	Average relative error
R	Correlation Coefficient
Gm	OTA transconductance value
C_L	Load Capacitance
$L(s)$	Transfer function in s domain
$(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$	State space matrix tuple
$\mathcal{T}$	Component-level topology
$\mathcal{T}_F$	Functional topology
S_{L12}	$L12$ norm of the sensitivity function
$\mathbf{W}$	Observability Gramian matrix
$\mathbf{K}$	Controllability Gramian matrix
$\mathbf{T}$	Topology transformation operator

Chapter 1

Introduction

The growing complexity of CMOS integrated systems being designed today, together with the increasing fragility of analog component blocks brought about by shrinking geometries and reduced power consumption, pose tremendous 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 architectural 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., that are required to realize the specified behavior of the system [3].

An analog high-level design process is defined as the translation of analog system-level specifications into a proper topology 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 a functional description of the system and the 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. This phase of

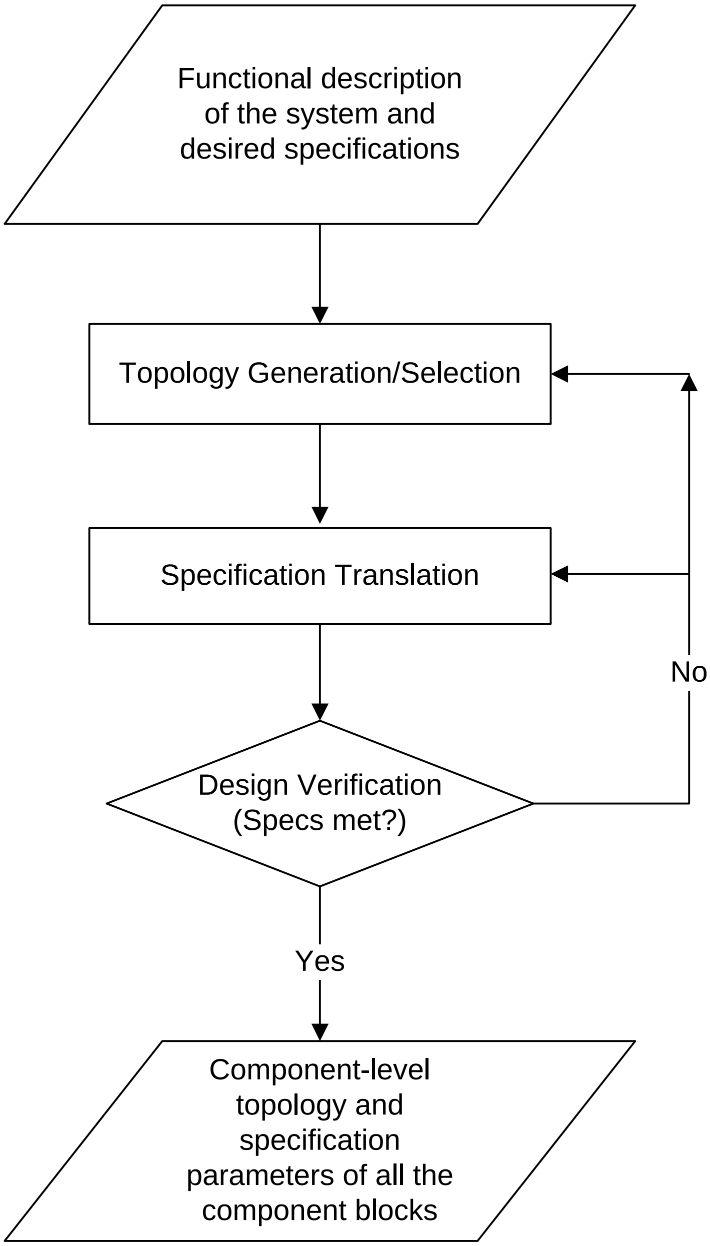

Functional description of the system and desired specifications
Topology Generation/Selection
Specification Translation
Design Verification (Specs met?)
No
Yes
Component-level topology and specification parameters of all the component blocks

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

the analog high-level design process is called topology generation/selection. 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 of the selected high-level 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 [2, 3]. 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 develops methodologies for implementing three different tasks of an optimization-based analog high-level design process. These are (i) high-level performance estimation, (ii) generation of an optimal component-level topology of a system and (iii) 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 defines the specific problems that have been addressed in the present work, gives an overview of the present research 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 following topics (i) high-level performance estimation, (ii) topology generation and (iii) specification translation. This provides us the motivation behind 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 (e.g., transistor sizes, biasing) of an analog circuit as inputs and determines its performance parameters (e.g., power consumption, thermal noise) 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 the 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 equation-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 a 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 of 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 representation is used. An automatic procedure for generation of posynomial models using fitting technique is described in [11, 12]. This technique overcomes several limitations of the handcrafted posynomial modeling techniques. The models are built from a set of data obtained through SPICE simulations. Therefore, full accuracy of SPICE simulation is 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. In [13] a method has been presented to automatically generate compact symbolic performance models of analog circuits with no prior specification of an equation template. A neural network based tool for automated power and area estimation is described in [14]. Circuit simulation results are used to train a neural network model, which is subsequently used as an estimator. Fairly recently, support vector machine (SVM) has been used for modeling of performance parameters for RF and analog circuits [15, 16, 17]. Analog performance models constructed with regression technique are generally fast to evaluate and the accuracy with respect to circuit-level simulation results is also good. An important advantage of a non-parametric regression technique (e.g., SVM technique) over a parametric regression technique (e.g., curve-fitting technique) is that it does not require any model template. However, a major limitation of the non-parametric regression technique is that, the generalization ability of the constructed models is often not good. In addition, the model construction time is generally 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 good generalization ability and low construction time using non-parametric regression 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 [18]. 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 [19], the topology selection tool selects only those topologies from the library 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 [20], 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 [5, 21], 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 the selection of an optimal topology for $\Sigma\Delta$ modulator in [22, 23, 24].

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 [25]. At the architectural level, this has been used in [26] 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 all the available topologies are selected 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, 27, 28]. The topology generation methodology starts from a signal flow graph (SFG) description of the functional topology [1] of a system. In [7], a branch-and-bound algorithm first generates alternative component-level topologies by mapping SFG components to circuit-level component blocks, stored in a library. For each resultant topology, a genetic algorithm-based technique is used for constraint transformation such that an optimal component-level topology is finally selected. In [27], a tabu search method and heuristic conversion rules are used to find different opamp-based topologies for linear analog systems. Through a parametric optimization procedure, an optimal component-level topology is finally generated. In [28], 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 low level description and is oriented to a specific functional topology. These methodologies therefore, do not consider the task of generation of an optimal functional topology of a system. However, for several complex analog component blocks/systems, e.g., $\Sigma\Delta$ modulator, higher order state variable filters, there exists several functional topologies and an optimal functional topology needs to be selected prior to the component-level topology generation process. In [29], component-level topologies for analog filters have been generated from a transfer function model. This is a relatively higher level of description. The functional topologies of the filters are generated from the transfer function. The component-level topologies are generated by mapping the functional component blocks to circuit-level implementation style specific realizations. However, this method does not include any

---

[1] A functional topology of a system is an interconnection of a set of functional component blocks, e.g., adder, integrator, realizing the desired functionality of the system.

performance optimization procedure within it. Therefore, the aspect of generation of an optimal component-level topology is not addressed.

In this work, we have explored the possibility of developing a methodology for generating an optimal functional and component-level topology of a system from a high-level description of the system (e.g., transfer function model).

1.1.3 High-Level Specification Translation

A high-level specification translation procedure is implemented through a design space exploration procedure. The crucial components of a specification translation procedure are the feasibility models of the component blocks of the system topology and an exploration algorithm. The feasibility models are required for limiting the design space exploration procedure to generate realizable values of the component block specifications. [30] presents a technique for construction of the feasibility models using binary search techniques - radial binary search and vertical binary search. [31] presents two methods for the calculation of the feasible performance values of analog circuits. The first method [32] computes the Pareto-optimal trade-offs of competing performances with full simulator accuracy. The Pareto front is a part of the boundary of the feasible performance region. The second method [33] 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 [34]. 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 [19]. A directed interval-based search space profiling technique and a genetic optimization-based constraint transformation technique are described in [35]. The constructed feasibility models are used in conjunction with an optimization algorithm to implement a specification translation process [19, 31, 35, 36, 37]. The standard optimization algorithms like simulated annealing, genetic algorithm etc., are used to implement the design space exploration procedure.

In this work, we have explored the possibility of developing a methodology for constructing the feasible design space by incorporating application system constraints and mutual influence among 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 this thesis and then gives a brief overview of the methodologies adopted to address them. Finally, the major contributions of this thesis are summarized.

1.2.1 Problem Definition

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

2. Development of a methodology for top-down generation of an optimal functional and 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 sub-sections give an overview of the approaches for solving these problems.

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 non-parametric regression technique. The transistor sizes of the circuit-level implementations of the component blocks along with a set of geometry constraints applied over them define the sample space. A Halton sequence generator is used for extracting samples from the sample space. Performance data are generated by simulating each sampled circuit configuration through SPICE. 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 a non-parametric regression function. Optimal values of the model hyper parameters are determined using two techniques - grid search-based technique and genetic algorithm (GA)-based technique. The generalization ability of the constructed models is estimated using a hold out method and a 5-fold cross validation method. Average relative error and correlation coefficients are calculated for measuring the quality of the constructed models. The constructed performance estimation models are used in a genetic algorithm-based high-level topology sizing process.

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 as functions of its specification parameters. The quality measures of the models are computed. The models are found to be quite accurate.
2. A comparison between the models constructed using the grid search-based training technique and the GA-based training technique w.r.t. the generalization ability and the training time is made. It is found that the training time is considerably less for the GA-based training technique compared to the grid search-based training technique, with almost the same generalization ability.
3. To demonstrate the topology sizing process, the interface electronics for a MEMS capacitive accelerometer has been chosen as an example. The topology parameters are determined such that a given set of specifications is satisfied with optimized performances. The predicted performances are compared with the 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 methodology for top-down generation of an optimal functional and component-level topology for linear analog systems, starting from a transfer function model of the system. The given transfer function model is converted to a state space matrix model. This acts as the basis for generation of topologies of the system. The topology exploration process is modeled as a state space matrix exploration process. A similarity transformation matrix is used for generation of new state space models from a given one. These new topologies have identical behavioral properties, but different performance properties. A 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 functional component blocks to generate an optimal functional topology, which is subsequently realized by appropriate analog component blocks to generate an optimal component-level topology of the system. The generated topology is then behaviorally simulated to check whether all the desired specifications are satisfied even in the presence of circuit-level non-idealities. If the test fails, the complete process is repeated and a new optimized topology is generated.

As a case study, the thesis presents a methodology for generation of an operational transconductance amplifier (OTA)-capacitor (C) based optimal topology for continuous-time $\Sigma\Delta$ modulator. The loop filter transfer functions and the desired dynamic range specifications 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 have been carried out for each of the examples are as follows:

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

2. The generated topology is behaviorally simulated to check whether it satisfies the desired dynamic range under non-ideal conditions and non-overload conditions. It is found that 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.

4. The relative power consumption is computed for the generated topology and the two standard topologies. It is found that the generated topology has lower relative power consumption compared to the standard topologies.

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 a 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 interval analysis techniques and the latter, via a bottom-up procedure through actual circuit simulation. Least squares support vector machine (LS-SVM)-based classification 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 the 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. GA-based design space exploration procedure determines the specification parameters of the component blocks.

3. 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 desired 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 the generation of good high-level performance estimation models for analog component blocks using least squares support vector machine (LS-SVM). The constructed models are accurate with respect to real circuit-level simulation results, fast to evaluate and have a good generalization ability. In addition, the model construction time is low and the construction process does not require any detailed circuit design knowledge.

2. A methodology is developed for generation of an optimal functional and 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 the 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 construction and accurate identification of a feasible design space and also for an exploration technique 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.

The present methodologies make the high-level design process fast and accurate. In addition, these can even be followed by novice users.

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 an optimization-based generic methodology for topology sizing and specification translation task. It also discusses in brief the 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 principles of the various methods for the generation/selection of an optimal component-level topology are also discussed.

Chapter 3 : This chapter describes a methodology for the construction of a high-level performance model using least squares SVM technique. It also includes a GA based topology sizing procedure, where the constructed performance models are used. Experimental results are 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 is presented. Experimental results are provided to illustrate the effectiveness of the methodology.

Chapter 5 : This chapter describes a methodology for high-level specification translation. Feasible design space is identified accurately using least squares SVM technique. GA-based optimization technique is used for design space exploration. Experimental results are 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 are provided therein.

Chapter 2

Optimization-based Analog High-Level Design Methodology

The flow diagram of a general procedure for an analog high-level design process was 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, and 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. The techniques that are used in the present research work are identified within the chapter. This chapter therefore, provides the necessary background required for understanding the contributions of the present 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 optimization procedures. Section 4 presents an overview of the various techniques for generation/selection of system topologies. Finally, Section 5 presents a summary of the chapter.

2.1 Generic Methodology

In an optimization-based high-level design methodology, the design problem is translated into a function minimization problem which is solved through numerical optimization techniques. These techniques implicitly explore 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 the 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 a function. The cost function being minimized during the optimization process contains two types 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, namely simulation-based approach and equation-based approach are distinguished. The basic principles 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 be-

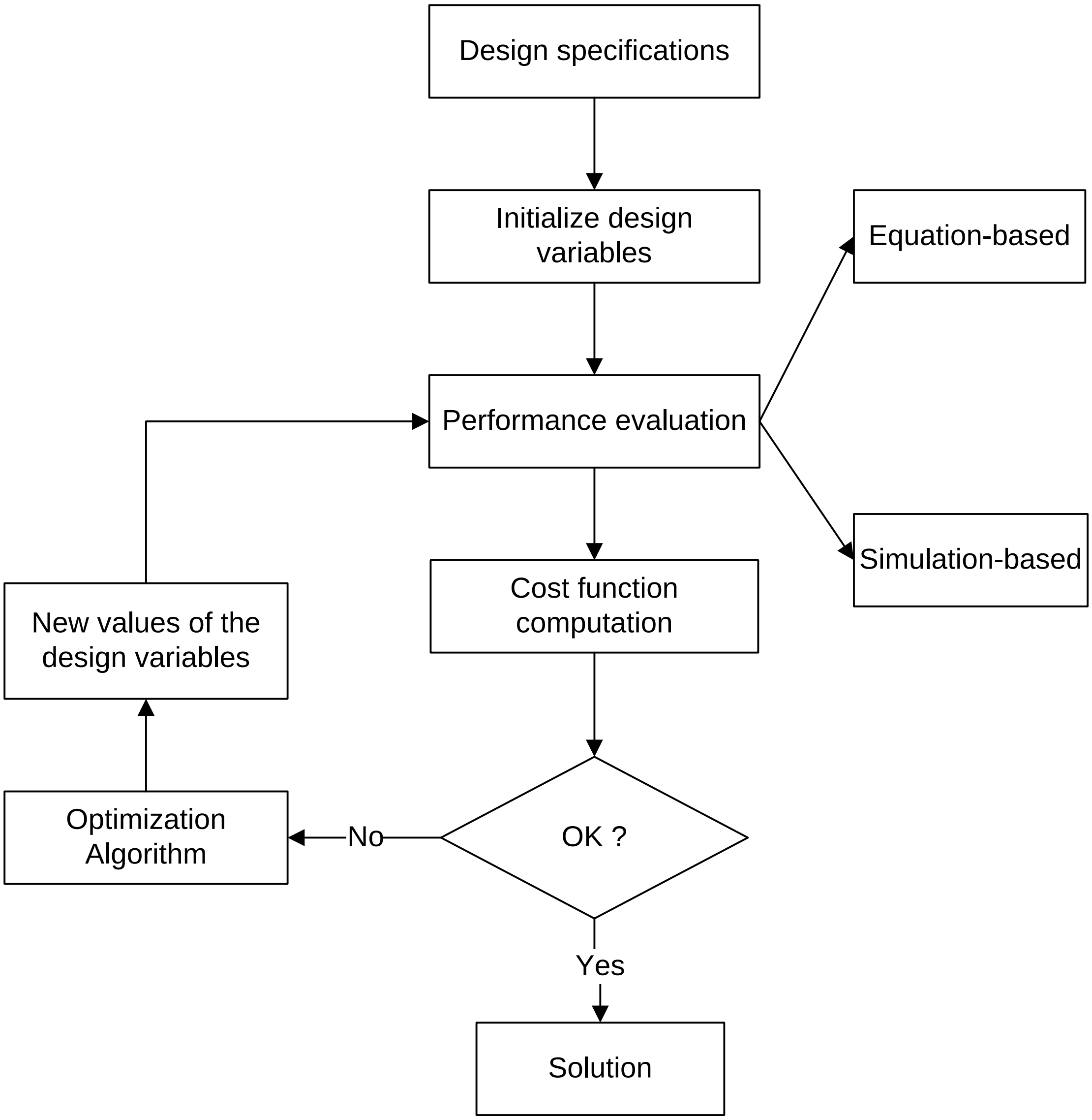


Figure 2.1: Optimization-based design flow

havioral models are developed for all the component blocks of the system topology. The specification parameters of the component blocks serve as the model parameters. The system topology is described as an interconnection of these behavioral models. Amongst the model parameters, those that would be used as design variables along with the respective bounds are identified by the user. 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 to be applied, simulation commands and the required data processing of the simulation results to obtain the desired system performances.

A serious problem of this approach is that the time required to evaluate the performances of a design is high. 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 the user can program a new high-level design problem in a minimum amount of time (provided, 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 [21]. 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 works for $\Sigma\Delta$ modulator design are reported in [22, 23].

The simulation-based approach has been used in implementing the design space exploration procedure, described in Chapter 5 of the dissertation.

2.1.2 Equation-based Approach

This approach uses analytical equations for performance evaluation. These equations directly relate the specification parameters of the component blocks to the desired specifications of the system and are generally derived either from the designer's knowledge or through symbolic analysis approach [3]. An important advantage of this approach is that the process of 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 a difficult, time consuming and 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 to the system topology and cannot be reused for other topologies.

Symbolic equations-based high-level design procedure have been reported in [7, 27]. This technique has also been applied to the high-level design of $\Sigma\Delta$ modulator in the SD-OPT tool [38].

The equation-based approach has been used in the implementation of the design space exploration procedure, described in Chapter 3 and 5 of the dissertation.

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 influences its behavior. Then

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

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

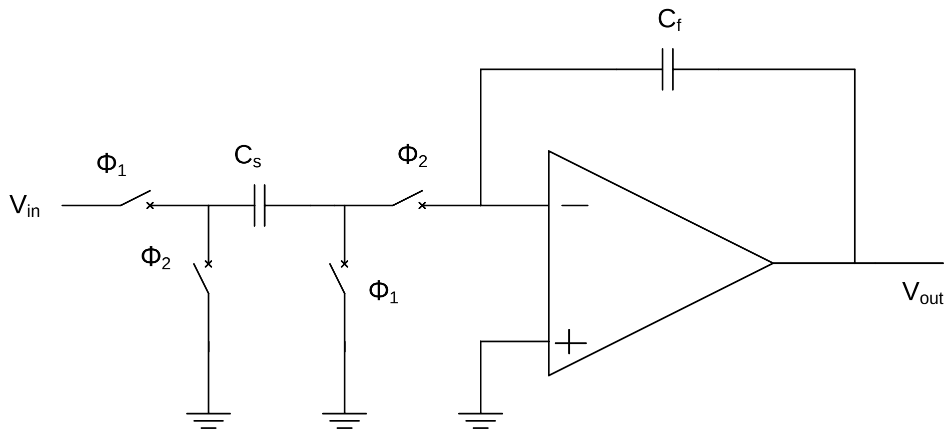
Good behavioral models are essential components in a behavioral simulation-based high-level design process. The models need to be good in two senses [39]. First, they must accurately represent all practical circuit behavior. For example, a behavioral model for a voltage amplifier must capture all of 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 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 below the basic principle of each of these techniques in brief.

2.2.1.1 Analytical Techniques

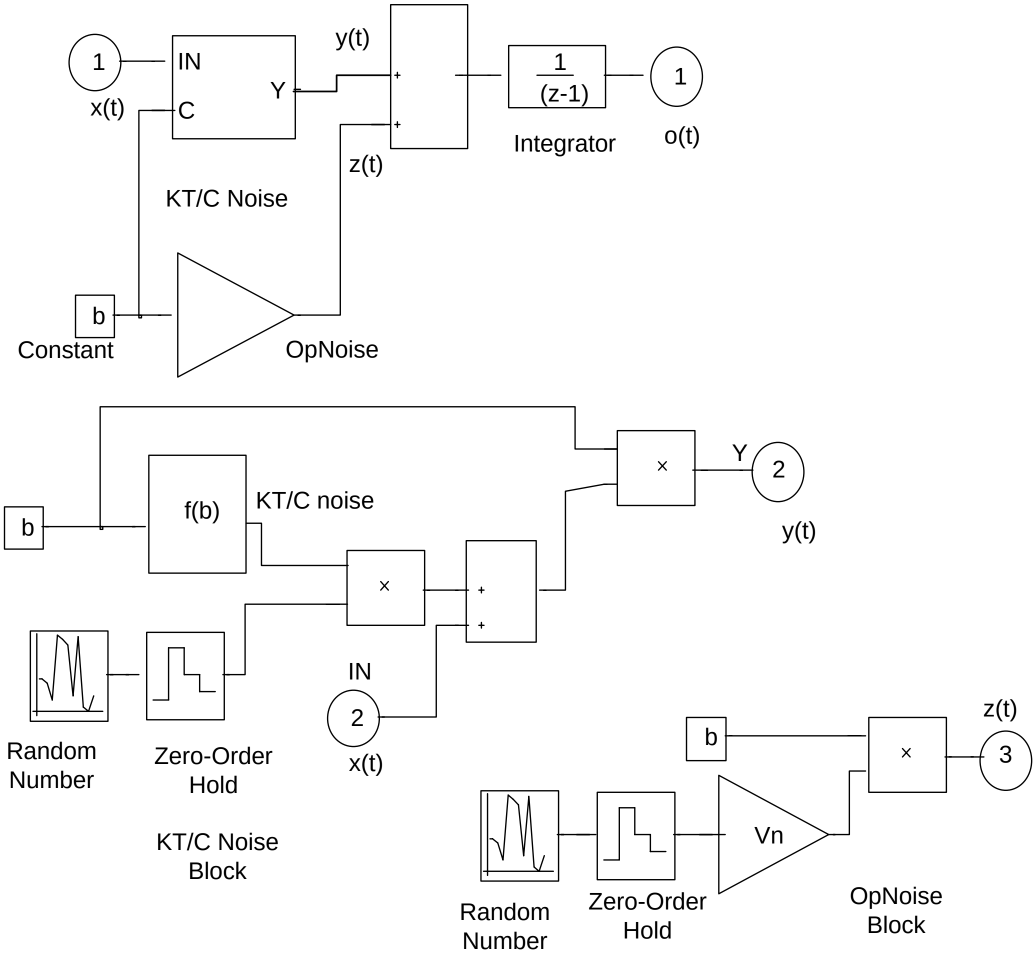
The complete behavior of a component block is split up into two parts– fundamental/ideal behavior and non-idealities. For any analog component block, the ideal behavior is generally a simple mathematical operation such as scaling, integration, multiplication, etc. The non-idealities are then modeled 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 hypotheses which are typically applicable to a particular system only. 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.

The technique 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

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



(a) Single-ended SC integrator.



(b) Model of a noisy integrator.

Figure 2.2: Simulink based behavioral model

$C_s/C_f = b$ 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 input voltage $x(t)$ leading to

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

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

$$z(t) = bV_n RN(t) \qquad (2.4)$$

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 [23, 40, 41]. Behavioral modeling and simulation of pipelined ADC and PLL following this technique are discussed in [42] and [43] respectively.

The behavioral models used in this dissertation for behavioral simulation purpose are constructed using this technique and are implemented with Simulink models.

2.2.1.2 Fitting or Regression Methods

In fitting or regression methods a parameterized model, e.g., a rational transfer function or 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. The drawbacks of these methods are that a good model template needs to be selected before the construction process and that the quality of the estimated results depends heavily upon the chosen template. The template 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 with the response of the real circuit. The choice of an appropriate neural network structure is also not an easy task.

A fitting approach for generation of posynomial models for analog circuits has been described in [11]. Neural network based approach has been used in [14, 44].

It is to be noted that this technique is not restricted to behavioral model generation purpose only. In Chapter 3 of this thesis, this method has been used for construction of high-level performance estimation models.

2.2.1.3 Symbolic Model Generation Methods

Symbolic analysis of an analog circuit is a 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) the circuit elements, represented by symbols. This technique is thus complementary 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 [45]: determinant-based method, signal flow graph method and tree enumeration method. Determinant-based methods solve a set of linear equations implied by the symbolic analysis procedure [46]. 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 describes a network as a graph. Whatever be the approach followed, the expressions generated can then be further 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 [47], OPASYN [48], AMGIE [19].

Although this method has not been used in this dissertation, it is presented here to make the discussion complete.

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 [49, 50].

This method has not been used in the present work.

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 some design parameters of the block are given 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 [45] or through regression method [49], 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' [8]. 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.

A fairly complete survey of the various works employing bottom-up approaches for performance model generation is provided in [49]. 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 [14, 44]. Support vector machine based performance model generation technique is described in [16].

In Chapter 3 of the dissertation, this approach has been followed for constructing the high-level performance models. Least squares support vector machine has been used as the regressor.

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 on 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, which is not an easy task.

A top-down approach for estimating power of a high-speed CMOS ADC is described in [51]. This approach has been used for estimating performances of analog filters in [27]. In [26], 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 a 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) [31]. 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 [34]. 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 [19]. In [33], the 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 [30]. 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 [52]. A Pareto front computation technique using normal boundary intersection method is described in [32].

In Chapter 5 of the dissertation, feasibility models have been constructed using least squares support vector machine technique.

2.3 Optimization Methods

In a parametric optimization procedure, the topology of the circuit/system and the component blocks is 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 problems are formulated as 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. In addition, we discuss in short the basic prinicple behind multi-objective optimization technique, which is another class of technique, widely used in analog circuit synthesis and optimization.

2.3.1 Deterministic Methods

In these methods, the updating process of the design variables requires information about the cost function and its derivative. The changes in the design variables that make the cost function decrease are accepted in the process. Commonly used deterministic methods are simplex methods, gradient-based methods, etc. A limitation of this approach is that the optimization process in many cases is quickly trapped in a local optimum of the cost function. Another problem is the rapid increase of the execution time with the increase in the number of design variables and design space. These techniques are used primarily for the fine tuning of sub-optimal sizings.

2.3.2 Stochastic Methods

In these methods, the design variables are varied randomly. The derivatives of the cost function are not required. Greedy stochastic algorithms only accept a new set of variables if it reduces the cost function value. The main advantage of the stochastic methods over the deterministic ones is the capability to escape from local optimum and hence a higher probability to reach a global optimum. Simulated annealing (SA) is a 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 (GA), 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 is 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 [47], ASTRX/OBLX [53], ORCA [21], SD-Opt [38] and so on. On the other hand, some of the key analog CAD tools using genetic algorithm are ANTIGONE [28], Watson [54] etc.

In this dissertation, the stochastic methods – SA and GA have been used as the optimization methods. The detailed description of these algorithms have been presented in Appendix B of the dissertation.

2.3.3 Multi-Objective Optimization Method

In virtually every engineering problem a trade-off exists between two or more competing objectives, i.e., improving one forces the other(s) to worsen. In analog design optimization problems, multi-objective optimization problems often need to be solved. This is because the analog performance parameters are often tightly coupled and competitive in nature.

Traditional single-objective optimization algorithms provide only one solution (sometimes a set of candidate solutions) to such problems which minimizes/maximizes an overall objective function obtained by mixing individual targets through apprpriate weightage factors. The use of single-objective optimization technique for solving trade-off problems leads to inferior results for several reasons.

- The results of such an optimization process are values of design variables for which global cost function is minimized. However, no information is available on how far this design point is from the optimal value for each of the (possibly conflicting) individual objectives.

- With this technique, it is not possible to know the specific design variable which is driving the optimizer towards the solution obtained.

- There is no formal procedure for the choice of individual weights through which the individual cost functions are combined.

A multi-objective optimization problem is formally defined as follows:

$$\begin{aligned}
\text{Minimize} \quad & f_m(\mathbf{x}), & m = 1, 2, ..., M; \\
\text{subject to} \quad & g_j(\mathbf{x}) \geq 0, & j = 1, 2, ..., J; \\
& h_k(\mathbf{x}) = 0, & k = 1, 2, ..., K; \\
& x_i^L \leq x_i \leq x_i^U, & i = 1, 2, ..., n.
\end{aligned} \tag{2.6}$$

A solution $\mathbf{x}$ is a vector of n design variables: $\mathbf{x} = [x_1, x_2, ..., x_n]^T$

In multi-objective optimization problems, few important concepts are commonly used. These are:

1. In an n dimensional space, the inequality operator is redefined in the following manner
$$\mathbf{a} \leq \mathbf{b} \quad \Leftrightarrow \quad a_i \leq b_i \quad \forall \quad i \in \{1, 2, ..., n\} \tag{2.7}$$

2. A design variable vector $\mathbf{x}$ is said to be Pareto-dominant over another design variable vector $\mathbf{y}$ iff the following holds:
$$\mathbf{x} \text{ dominates } \mathbf{y} \quad \Leftrightarrow \quad f(\mathbf{x}) \leq f(\mathbf{y}) \quad \forall \quad i \in \{1, 2, ..., k\} \tag{2.8}$$

3. A vector $\mathbf{x}$ is a Pareto-optimal design vector with respect to a set A of design variables iff $\mathbf{x}$ is not dominated by any of the design vectors in the set A.
$$\mathbf{x} \text{ is Pareto-optimal w.r.t A} \quad \Leftrightarrow \quad \nexists \mathbf{a} \in A : f(\mathbf{a}) \leq f(\mathbf{x}) \tag{2.9}$$

4. The set Σ of Pareto-optimal design vectors for the given set A is called the Pareto-optimal front.

The aim of a multi-objective optimization algorithm is to find samples of the Pareto front with respect to the complete set of feasible design vectors. The genetic algorithm is often used to solve multi-objective optimization problems.

Although this type of problem has not been addressed in this dissertation, it is presented here to make the discussion complete.

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. These two tasks are however, different. 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, on the other hand, 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 sub-sections we give a brief overview of the basic principle of these methods. This is based upon the survey paper by Martens and Gielen [18].

2.4.1 Selection before or after sizing

There are two approaches to this method. In one approach, the topology is first selected by a designer based upon experience or a knowledge-assistant tool. The topology parameters are then determined such that system performances are optimized. There are several techniques behind the process of selection of a topology. Some of the commonly used techniques are rule-based selection, use of feasibility function of each topology, etc. The details of these have been discussed in [18]. In the second approach of this method, 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.

This methodology has been used for selection of an optimal topology for systems like PLL in [55], RF systems in [5, 21], $\Sigma\Delta$ modulators in [38].

In Chapter 3 of this thesis, a genetic algorithm-based methodology has been developed for the purpose of topology sizing and optimization. This can be used to implement a sizing based topology selection process.

2.4.2 Selection during sizing

In this method, the tasks of selection and sizing are performed simultaneously. A template of the topology is defined 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 an 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 employing algorithms like branch and bound algorithm etc. 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 [25] and $\Sigma\Delta$ modulators in [26, 41].

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. There are several reasons behind the requirement of a topology generation methodology. With a topology generation process, it is possible to explore the largest design space including possibly more optimal solutions. Furthermore, only standard component blocks are covered by templates, whereas new applications may require combinations of several component blocks. Finally, the topology generation process adds the possibility of examining new approaches for existing problems.

The top-down methodology starts from a functional description of a 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 suitable internal representation like signal flow graph [27, 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.

This methodology has been used for generation of opamp based topologies for linear analog systems in [27], general analog systems in [7], ADC in [56]. In [28], a genetic algorithm based technique has been described for simultaneous topological and parametric optimization process.

In Chapter 4 of the dissertation, a top-down methodology has been developed for generation of an optimal component-level topology for linear analog systems starting from a transfer function description of the system.

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 the present work 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 been discussed. The deterministic as well as the stochastic methods of implementing an optimization procedure have 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 for the present research work.

Chapter 3

Generation of High-Level Performance Estimation Models

In Section 2.2.2 of the dissertation, we have 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. The constructed models are subsequently used to implement a genetic algorithm (GA)-based topology sizing process.

The chapter is organized as follows. Section 1 defines the high-level performance estimation models and their essential characteristics. In Section 2, the present methodology to generate performance models is described in detail. Section 3 presents a comparison between the present methodology and the other existing methodologies. The GA-based topology sizing process 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 an analog component block when some high-level design parameters of the block are given as inputs [8]. The important requirements for a good high-level performance 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 a 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 [49]. In addition, a topology sizing process employing 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. The model accuracy is measured as the deviation of the model predicted value 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 a 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 test-bench circuit and design variable selection and computational time needed to use an algorithm to train a model. As a rough estimate, the construction cost is measured as

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

where the terms are self explanatory. There exists a trade-off between these requirements since 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.

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 the 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 prime indicating the reduction is omitted in the rest of this chapter. Both the inputs and the output of the performance model $\mathcal{P}$ 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 is 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 are combined into a single non-linear vector inequality, which is interpreted element wise as:

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

These constraints as functions of $\bar{\alpha}$ define a space, which we call as a 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. Within the sample space, the circuit performance behavior becomes weakly non-linear [57]. Therefore, simple sampling strategies are used to construct models with good generalization ability.

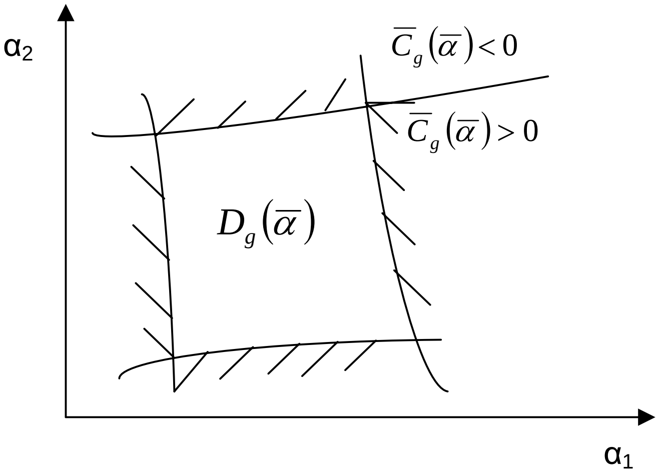


Figure 3.1: 2D projection of a four dimensional sample space.

The transistor sizes for generating training data corresponding to $\bar{X}$ and $\bar{\rho}$ are 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 from the sample space for training data generation. In the present work, these points are generated using a Halton sequence generator [58]. 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.

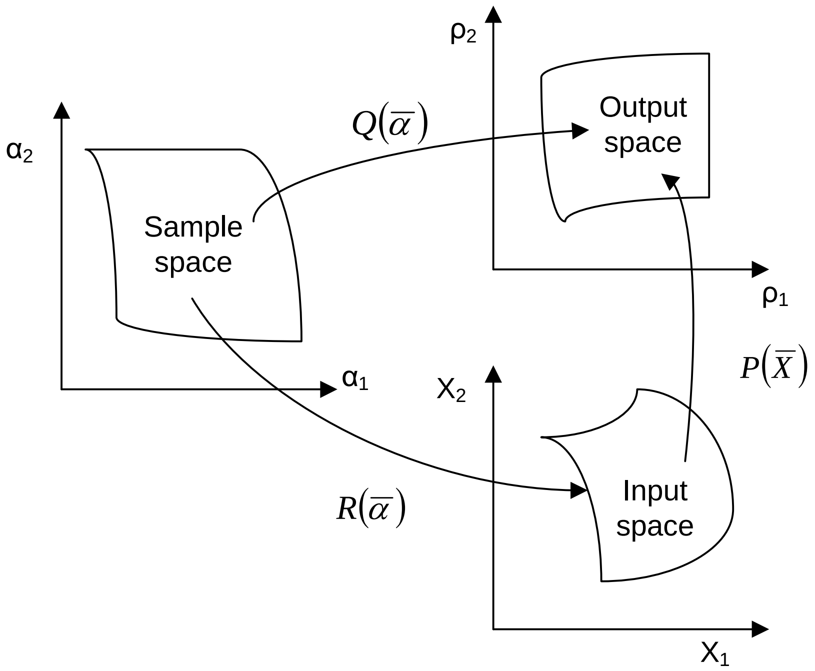


Figure 3.2: Non-linear relation between the sample space and the input, output space.

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 a high-level performance model $\mathcal{P}$ are functions of transistor-level parameters $\bar{\alpha}$. The inputs and the 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 the electrical parameters are complex non-linear functions, considering the deep submicron effects of MOS transistors. In this work, these are achieved element-wise through a 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 the 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 that would provide sufficient data to facilitate auto-

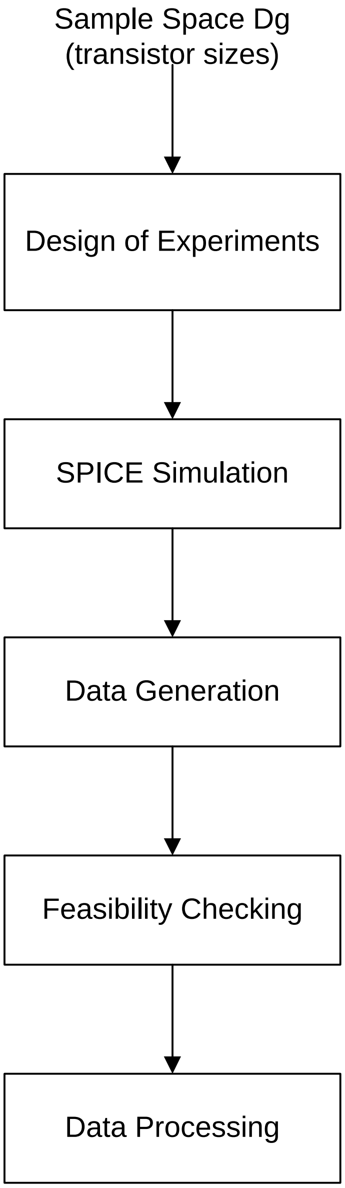


Figure 3.3: An outline of the procedure for generation of training data.

matic 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 is then considered to ensure that only feasible data are taken 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 [30, 57, 59]:

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. The data of the input and/or output parameters are scaled. The 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 $[0, 1]$ [60]:

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

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

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-SVM

The detailed theory of least squares support vector machine (LS-SVM) based function estimation process has been discussed in Appendix A.1 of this dissertation. In principle, an LS-SVM always fits a linear relation ($\bar{\rho} = w\bar{X} + b$) 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 [61]:

$$\mathcal{J} = \frac{1}{2}w^T w + \gamma \sum_{k=1}^{N_{tr}} e_k^2 \tag{3.11}$$

subject to

$$\bar{\rho}_k = w^T \phi(\bar{X}_k) + b + e_k \quad k = 1, 2, ..., N_{tr} \tag{3.12}$$

where N_{tr} denotes the total number of training data sets and the suffix k denotes the index of the training set, i.e., k^{th} training data, ϕ is the mapping of the vector $\bar{X}_k$ to some high-dimensional feature space, b is the bias and w is the weight vector of the same dimension as the feature space. This formulation involves the trade-off between a cost function term (first term in (3.11)) and a sum of squared errors (second term in (3.11)) 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 the regularization parameter. LS-SVM considers the optimization problem to be a constrained optimization problem and uses dual Lagrangian-based formulation

$$\mathcal{L} = \mathcal{J}(w, e) - \sum_{k=1}^{N_{tr}} \alpha_k \left(w^T \phi(\bar{X}_k) + b + e_k - \rho_k \right) \tag{3.13}$$

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

$$\bar{\rho} = \sum_{k=1}^{N_{tr}} \alpha_k K(\bar{X}_k, \bar{X}) + b \tag{3.14}$$

where $\alpha_k = \gamma e_k$ and $K(\bar{X}_k, \bar{X})$ is known 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 [61]. 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

$$K(\bar{X}_k, \bar{X}) = exp\left(-\left|\left|\bar{X}_k - \bar{X}\right|\right|^2 / \sigma^2\right) \tag{3.15}$$

where σ^2 is a parameter of the kernel and controls the width of the kernel function. The reasons for this choice are: First, an RBF kernel non-linearly maps samples into a higher dimensional space unlike the linear kernel. Furthermore, the linear kernel is a special case of the RBF as argued in [62]. In addition, the sigmoid kernel behaves like an 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 the kernel function in constructing circuit-level performance models by many researchers [16, 34].

3.2.3.1 Selection of Hyper parameters

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 non-linear mapping from the input space to some high-dimensional feature space. These higher level parameters are usually referred to 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 generalization error is a function that measures the generalization ability of the constructed models, i.e., the ability to predict correctly the performance of an unknown sample. 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. The test data are completely unknown to the estimator. The errors it makes are accumulated 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 the data points that end up in the training set and on those 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: In this method, 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 once for testing. 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 upon how the data gets divided. Every data point gets to be in the test set exactly once, and gets to be in the 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 re-run from scratch k times, which means that 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 by $\log_2 \gamma \in \{lb_\gamma, ub_\gamma\}$ 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 within the grid space, 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 the SVM 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 the 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 chromosomes consist of two parts, $\log_2 \gamma$ and $\log_2 \sigma^2$. Binary encoding scheme is used to represent the chromosomes. During the evolutionary process of GA, a model is trained with the current hyper parameter values. The hold-out method as well as the k-fold cross validation method are used for estimating the generalization error. The fitness of the chromosomes depends on the average relative error (ARE) calculated over the test samples. The fitness function is defined as

$$\text{fitness} = \frac{1}{ARE(\gamma, \sigma^2)} \tag{3.16}$$

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

$$ARE = \frac{1}{N_{te}\rho'} \sum_{1}^{N_{te}} (\rho' - \rho) \tag{3.17}$$

Here N_{te}, ρ and ρ' are the number of test data, the SVM estimator output and the corresponding SPICE simulated value, respectively. The fitness of each chromosome is taken to be the average of five repetitions. This reduces the stochastic variability of the model training process in GA-based LS-SVM. Roulette wheel selection technique is used for the selection operation [1]. Besides, in order to keep the best chromosome in every generation, the idea of elitism is adopted. Uniform crossover technique is

---

[1] The genetic algorithm has been discussed in detail in Appendix B.1 of this dissertation

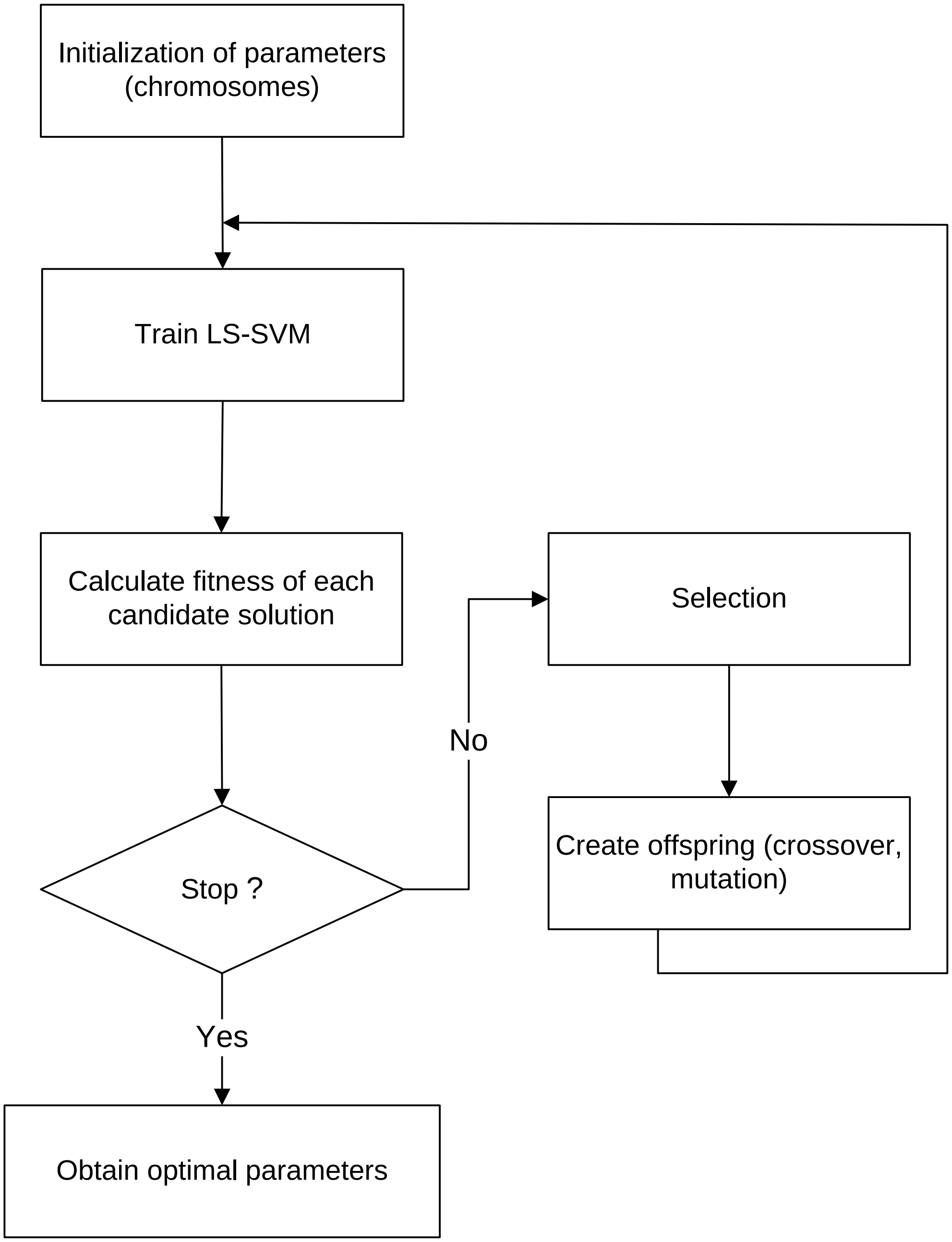

Initialization of parameters (chromosomes)
Train LS-SVM
Calculate fitness of each candidate solution
Selection
No
Create offspring (crossover, mutation)
Stop ?
Yes
Obtain optimal parameters

Figure 3.4: Outline of GA-based hyperparameter selection procedure

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 a pre-defined threshold over certain fixed generations, the whole process is terminated and the corresponding hyper parameter pair is taken as the output.

3.2.4 Quality Measures

Statistical functions are generally used to assess the quality of the generated estimator. The ARE function defined in (3.17) is one such measure. Another commonly used measure is the correlation coefficient. This is defined as follows:

$$\text{Correlation Coefficient}(R) = \frac{N_{te}\sum \rho\rho' - \sum \rho \sum \rho'}{\sqrt{\left[N_{te}\sum \rho^2 - \left(\sum \rho\right)^2\right]\left[N_{te}\sum \rho'^2 - \left(\sum \rho'\right)^2\right]}} \tag{3.18}$$

The correlation coefficient is a measure of how closely the LS-SVM outputs fit 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 are used to generate the performance models. The models are thus accurate compared to real circuit-level simulation results. (ii) There is no need for any *a priori* knowledge about the unknown dependency between the inputs and the outputs of the models to be constructed. (iii) The generalization ability of the models is high. (iv) The model construction time is low and the construction process does not require any detailed circuit design knowledge.

The EsteMate methodology [14] using artificial neural network (ANN) and the SVM-based methodology discussed in [15, 16] are closely related with the present methodology. The methodology that we have developed, however has a number of advantages over them. These are:

1. In the EsteMate methodology, the specification parameters of a component block constitute the sample space for training data generation. The specification parameters are electrical parameters and there exists strong non-linear correlations amongst 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 along with a set of geometry constraints applied over them define the sample space. Within this sample space, the circuit performance behavior becomes weakly non-linear. Thus simple sampling strategies are used in our methodology to construct models with good generalization ability.

2. In EsteMate, for each sample, a complete circuit sizing task using a 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 that in the EsteMate methodology [14].

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 from 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 [63], 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 [16], uses heuristic knowledge to determine the model hyper parameters. The present methodology uses optimization techniques to determine optimal values for them. GA-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 [15].

3.4 Topology Sizing Methodology using GA

The topology sizing process is defined as the task of determining the topology parameters (specification parameters of the constituent component blocks) of a high-level topology such that the desired specifications of the system are satisfied with optimized performances. 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 multidimensional space, in which the topology parameters are the dimensions. The valid design space for 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. Performance estimation models for estimating the performances of a topology of the entire system are constructed by combining the LS-SVM models of the individual component blocks through analytical formulae. The performance estimation models take each combination of topology parameters and produce 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.

It is to be noted that the topology sizing process does not deal with the feasibility of the topology parameters. This process is meant for comparing the performances between a set of topologies. Hence, this process is used to implement a topology selection process following the ‘selection before or after sizing’ technique (c.f. 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 [64].

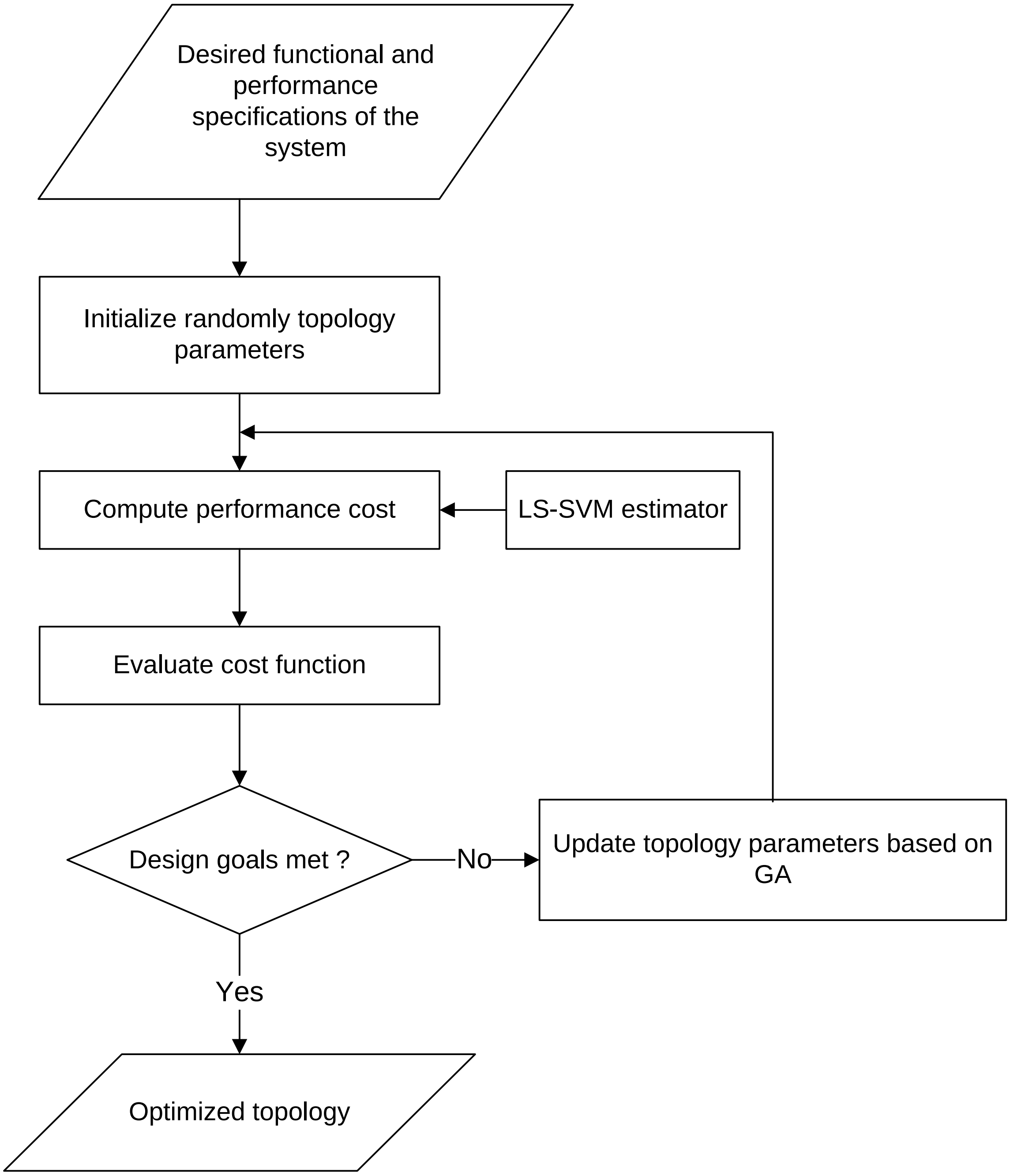

Desired functional and performance specifications of the system
Initialize randomly topology parameters
Compute performance cost
LS-SVM estimator
Evaluate cost function
Design goals met ?
No
Update topology parameters based on GA
Yes
Optimized topology

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

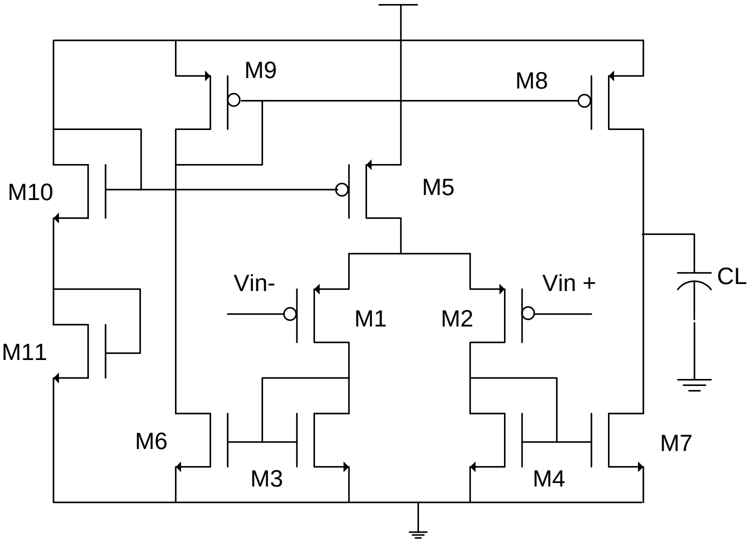


Figure 3.6: PMOS OTA circuit

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}	$\leq 2mV$
Performance Constraints	Slew rate	$[0.1V/\mu s, 20V/\mu s]$
	Bandwidth	$\geq 2MHz$
	DC Gain	$\geq$ 70 dB
	Phase margin	$[45^0, 60^0]$

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)

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

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 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-SVM technique. The generalization errors are estimated through the hold-out method and the 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 hyper parameters (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 avarage relative errors for the test samples are low (i.e., the generalization ability of the models is 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 hyper parameter values and the quality measures are reported in Table 3.4 and

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	12.06
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	11.56

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	44.83
ρ_3	5.32	540.15	1.81	1.90	0.999	0.990	46.61

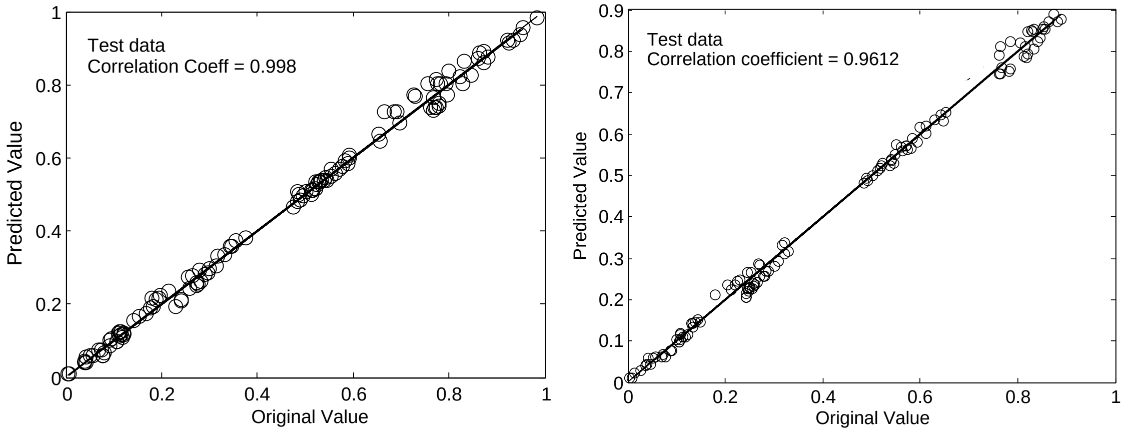
Table 3.6: Comparison between GA and Grid search technique for LS-SVM construction

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	12.06
	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	11.56
	Grid Search	5.43	387.55	2.02	3.14	0.999	0.937	118.13

3.5. From the results the above observations are also noted.

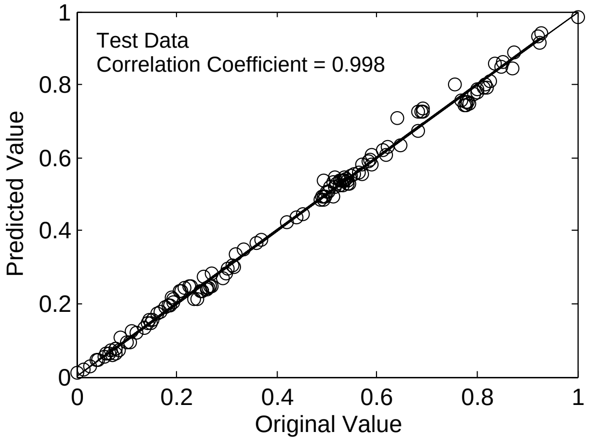
A comparison between the grid-search technique and the GA-based technique with respect to accuracy (ARE), correlation coefficient (R) and required training time is made in Table 3.6. All the experiments are performed on a PC with PIV 3.00 GHz processor and 512 MB RAM. We observe from the comparison that 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(a), Fig. 3.7(b) and Fig. 3.7(c) respectively. These scatter plots illustrate the correlation between the SPICE



(a) Scatter plot of estimated and original values for the noise model with normalized test data

(b) Scatter plot of estimated and original values for the power model with normalized test data.



(c) Scatter plot of estimated and original values for the impedance model with normalized test data.

Figure 3.7: Scatter Plot of the constructed models.

simulated and the LS-SVM estimated test data. The correlation coefficients are very close to unity. Perfect accuracy would result in the data points forming a straight line along the diagonal axis.

The LS-SVM network can be trained to learn any arbitrary nonlinear input-output relationships from corresponding data [61]. Therefore, nonlinear analog performance models can also be constructed with reasonable accuracy. For experimental demonstration, we choose to model slew rate as functions of DC gain and bandwidth. The design constraints as mentioned in Table.3.1 are considered. The GA has been used for selecting the model hyperparameters. The results are summarized in Table 3.7. We observe that the accuracy is quite good, similar to that for linear models.

Table 3.7: Construction of Nonlinear Performance Model

σ^2	γ	$ARE(\%)$		R	
		Training	Test	Training	Test
4.65	195.17	2.32	3.18	0.997	0.994

Table 3.8: Comparison between our methodology and 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 [14]	2564	641	2.88	6.53	10 hour	21 min

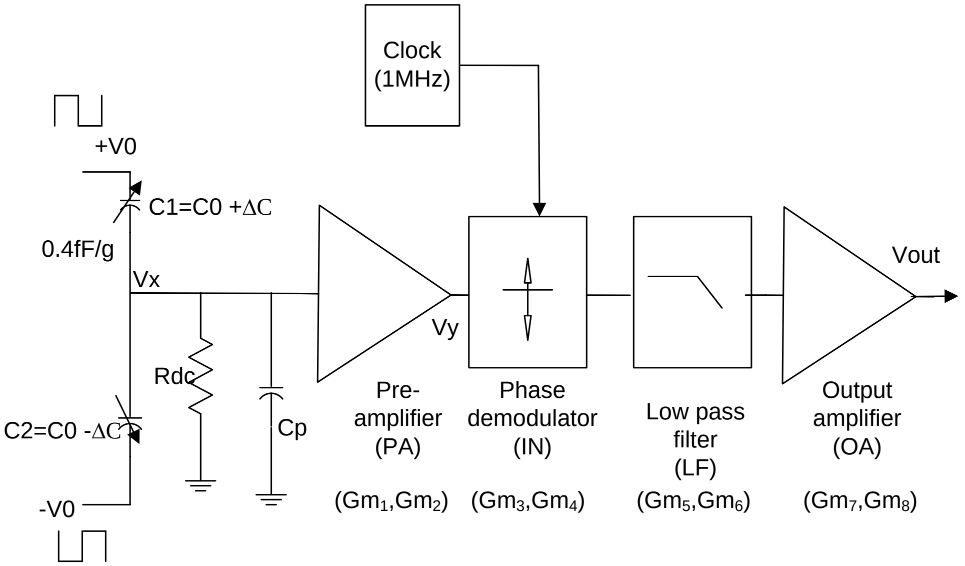
3.5.2 Experiment 2

The objective of this experimentation is to quantitatively compare between our methodology and the EsteMate [14]. The power consumption model is reconstructed using the EsteMate technique. The specification parameter space is sampled randomly. A set of 5000 samples is 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 to the specification parameters [65]. 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.

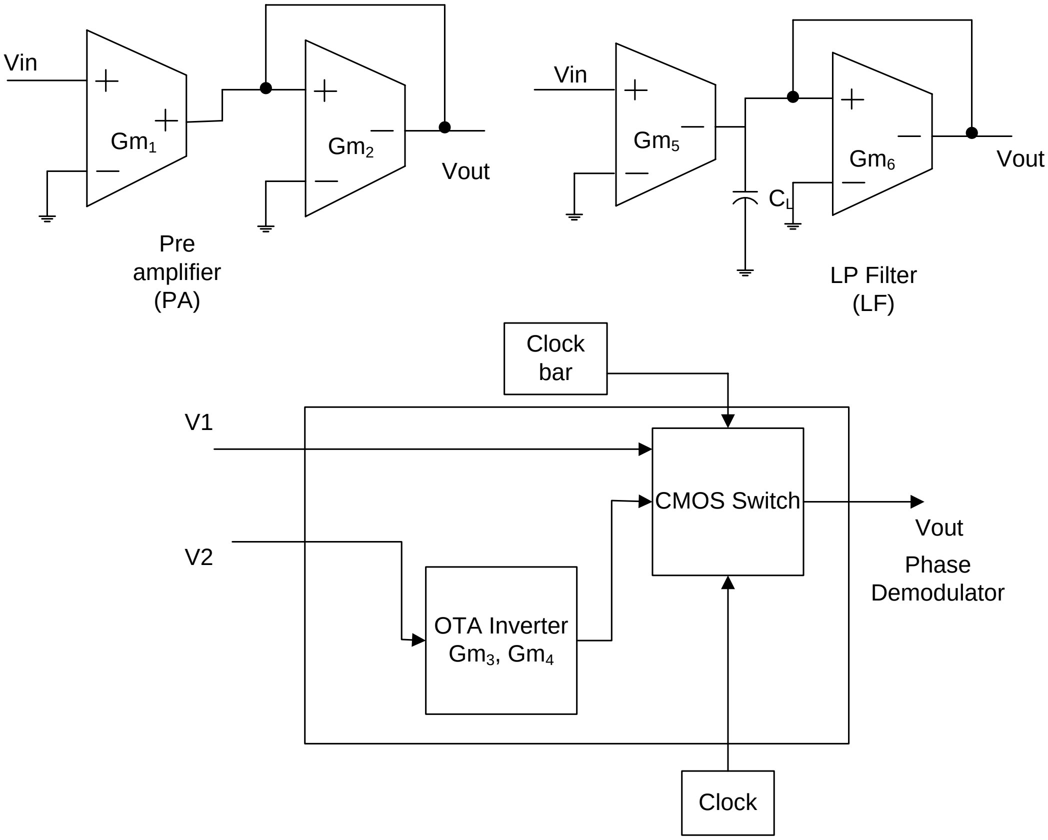
A comparison between the two methodologies is reported in Table 3.8. From the results, we find that the data generation time is much less in our method compared to the EsteMate method. In addition, we find that the generalization ability of our method is better than the EsteMate method. The experimental observations verify the theoretical arguments given in Section 3.3.

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. 3.8(a). In this configuration, a half-bridge consisting



(a) Voltage sensing configuration of the interface electronics for MEMS capacitive sensor



(b) OTA-C realizations of amplifier and filter.

Figure 3.8: Considered system for Experiment 3.

Table 3.9: 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	$\leq$ 40 kHz

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

$$V_{out} = V_0 \frac{2\Delta C}{2C_0 + C_p} A_v \tag{3.19}$$

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

The desired functional specifications to be satisfied are (i) output voltage sensitivity (i.e., the total gain, since the input sensitivity is known) and (ii) cut-off frequency of the filter. The performance parameters to be optimized are (i) input-referred thermal noise, (ii) total power consumption and (iii) parasitic capacitance at the sensor node V_x. The functional specifications and design constraints for the system are based on [66] and are listed in Table 3.9. 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. Figure 3.8(b) shows the implementations of the amplifier and the filter blocks using OTAs and capacitor.

High-level performance models for the synthesizable component blocks corresponding 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

Table 3.10: 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 3.11: 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	3.45	123.93	0.58	0.9999	0.62	0.9999

as on the functional specification, i.e. the output voltage sensitivity and the cut-off frequency are the transconductance values of all the OTAs involved. 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. Thus the Gm values of the OTAs are considered as high-level design parameters. The geometry constraints and the feasibility constraints for the PA block of the topology are tabulated in Table 3.10. Similar types of constraints are considered for the other component blocks also. The input-output parameters of the models to be constructed are extracted through techniques discussed earlier. The sensor node parasitic capacitance is measured utilizing the half-bridge circuit shown in Fig. 3.8(a), 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 (3.19).

Table 3.11 shows the hyper parameter 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 and input parasitics of the preamplifier block with the high-level design parameters (Gm_1, Gm_2) are shown in Fig. 3.9, Fig. 3.10 and Fig. 3.11 respectively. From Fig. 3.9, we see that for the noise model, Gm_1 has the major contribution in comparison to Gm_2 and as Gm_1 increases, the input referred thermal noise decreases. From Fig. 3.10, we see that for the power model, as Gm_1 and Gm_2 increases, the power consumption increases. From Fig. 3.11, we see that for the parasitics model, as Gm_1 increases with constant bias current, the parasitic capacitance increases. This is due to increase of input transistor width, which in turn increases the gate area.

The performance models corresponding to the noise and the power consumption for the PA block are reused for the other component blocks. This is because all the component blocks have topological similarities and each of them is constructed from OTA circuits, as demonstrated in Fig. 3.8(b). This a clear advantage of this methodology.

The performances of the individual component blocks are combined analytically to estimate the performances of the total system. This makes the task of generation of performance model of the complete system fairly simple. However, this comes at the cost of reduced model accuracy. This trade-off is an important characteristic of analog high-level modeling process, which has to be handled very carefully. In addtion, the task of analytical combination needs to be done manually.

The input referred noise and power consumption of the total system is given by

$$\begin{aligned} V_{nT}^2 &= V_{n1}^2(Gm_1, Gm_2) + \frac{V_{n2}^2(Gm_3, Gm_4)}{A_1^2} + \\ &\quad \frac{V_{n3}^2(Gm_5, Gm_6)}{A_1^2} + \frac{V_{n4}^2(Gm_7, Gm_8)}{A_1^2} \qquad (3.20) \\ P_T &= P_1(Gm_1, Gm_2) + P_2(Gm_1, Gm_2) + \\ &\quad P_3(Gm_1, Gm_2) + P_4(Gm_1, Gm_2) \qquad (3.21) \end{aligned}$$

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

With these, the optimization problem for the topology sizing task is formulated as

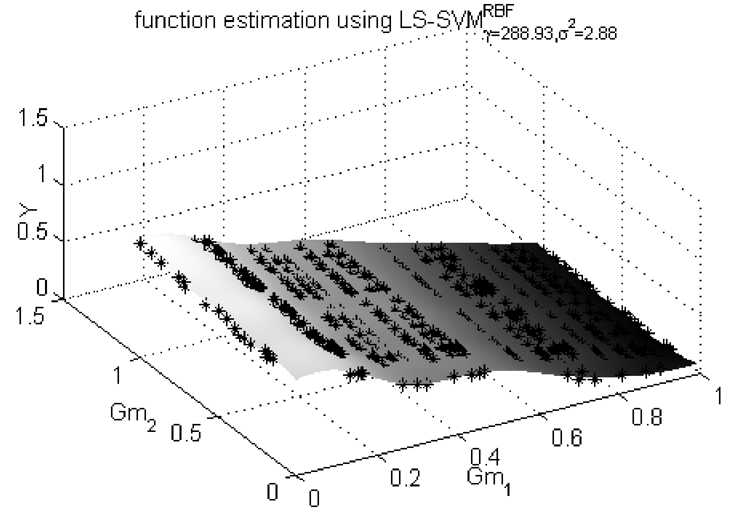


Figure 3.9: Noise as function of Gm_1 and Gm_2

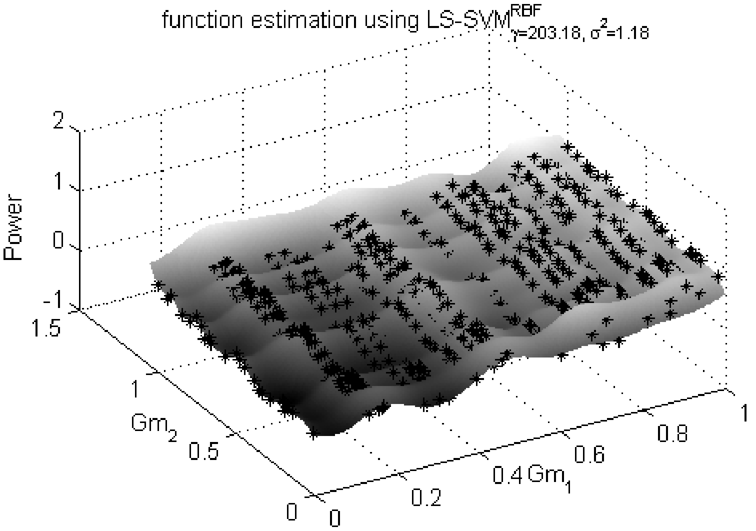


Figure 3.10: Power as function of Gm_1 and Gm_2

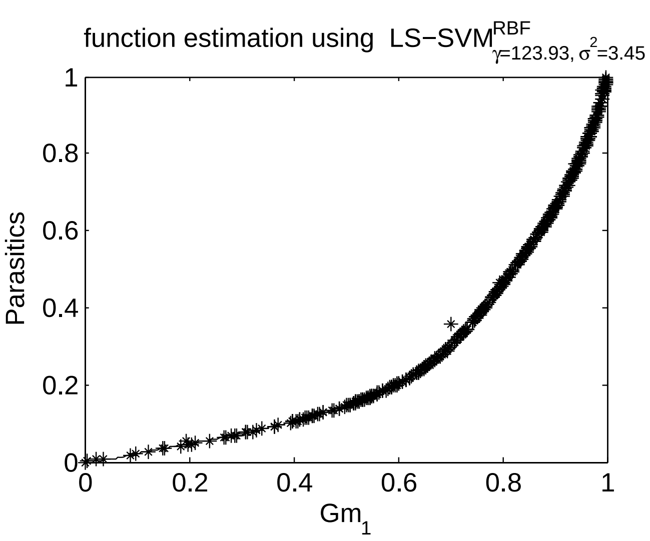


Figure 3.11: Input parasitics as function of Gm_1

$$\begin{aligned} \text{Minimize} \quad & \omega_1 V_{nT} + \omega_2 P_T + \omega_3 P_a \\ \text{such that} \quad & (V_{out})_{target} - V_{in}\left[\frac{Gm_1}{Gm_2}\frac{Gm_3}{Gm_4}\frac{Gm_5}{Gm_6}\frac{Gm_7}{Gm_8}\right] \leq \epsilon_1 \\ & f_c - \frac{Gm_6}{2\pi C_L} \leq \epsilon_2 \\ & Gm_{imin} \leq Gm_i \leq Gm_{imax} \\ & C_{Lmin} \leq C_L \leq C_{Lmax} \end{aligned} \tag{3.22}$$

where ω_i are the associated weights.

The target output voltage sensitivity of the system (i.e. the total gain of the system) is taken as $145 mV/g$ 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 3.12 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

Table 3.12: 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 3.13: Comparison of Predicted performances and SPICE value

Performances	Pred	SPICE	Error %
Noise ($nV/\sqrt{Hz}$)	19.65	20.32	3.40
Power (μ W)	572.78	592	3.36
Parasitics (fF)	92.05	94.12	2.24
Sensitivity	145.16	138	4.93
Cut-off (kHz)	35.28	38	7.70

predicted performances and simulated values is presented in Table 3.13. We observe that the relative error between predicted performances and simulated performances in each case is acceptable. However, for the output sensitivity and the cut-off frequency, the error is 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 along with a set of geometry constraints applied over them define the sample space. A Halton sequence generator is used as the sampling algorithm. The training data are generated through simple circuit simulation using SPICE. Least square support vector machine is used as the regression function. The generalization error has 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 a GA-based technique. 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. From the experimental results, it has been found that when the cross validation method is used for estimating the generalization error during the hyper parameter determination process, the constructed models have better generalization ability in comparison to that when the hold-out method is used for estimation. However, the cross validation method is more time consuming than the hold-out method. It has also been observed that the GA-based training technique is faster compared to the grid search-based training technique with almost the same accuracy. The constructed performance models have been used to implement a GA-based topology sizing process. To demonstrate this process, the interface electronics for a MEMS capacitive accelerometer has been chosen as an example. The topology parameters are determined such that the desired specifications are satisfied with optimized performances. The predicted results have been compared with SPICE simulation results. The two sets of results match closely.

The advantages of the present methodology are that the constructed models are accurate with respect to real circuit-level simulation results, fast to evaluate and have a good generalization ability. In addition, the model construction time is low and the construction process does not require any detailed circuit design knowledge. The methodology employs artificial intelligence-based technique for regression model generation and therefore, does not depend upon the nature of the functional relationship to be modeled. It has been experimentally demonstrated that the procedure is able to develop performance models for complex non-linear analog characteristics with good accuracy.

Although the procedure is largely automated, the designer still needs to provide several manual inputs. Firstly, the dominant specification parameters need to be supplied by the designers. The selection process is dependent upon designer's knowledge. Secondly, the tasks of sample data generation and regression model generation are performed discontinuously. Thirdly, the performance models of the individual component blocks are combined analytically by the designers to construct the performance model of a complete system.

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 this 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. Similarity transformation matrix is used as topology transformation operator. A simulated annealing-based optimization procedure determines an optimal topology of the system based upon the performances of the topologies. The entire methodology is illustrated with the help of a 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 the 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 the presence of circuit-level non-idealities.

The chapter is organized as follows. The complete methodology is described in detail in Section 1. A comparison between the present and the other existing methodologies is given in Section 2. Experimental results are provided in Section 3. Finally, conclusion is drawn in Section 4.

4.1 Top-Down Topology Generation Methodology

This section describes the top-down topology generation methodology. An outline of the methodology is given first. Each step of the methodology is described in detail by considering continuous-time ΣΔ modulator system as a case study. However, for pedagogical reasons, the topology generation step is discussed prior to the similarity transformation-based topology transformation step. Finally, the complete flow of the methodology for ΣΔ modulator system is outlined. The entire methodology tackles the task of topology generation through a simulated annealing-based procedure.

4.1.1 Outline of the Methodology

An outline of the methodology is shown in Fig. 4.1. The inputs to the topology generation process are a transfer function model of the linear analog system to be designed and the desired specifications. The transfer function specifies the input-output behavior of a system in the frequency domain. It is transformed into a time domain equivalent, the state space model. A state space model is a mathematical representation of a functional topology of the system. The tasks of topology exploration and selection are performed at the state space model level. A similarity transformation matrix is used as a topology transformation operator. From a given state space model, an infinite number of other state space models is generated using similarity transformation matrix operation. This corresponds to generation of an infinite number of other topologies from an initial topology. These newly generated topologies have same behavioral properties, but different performance properties. An optimal state space model is determined through an iterative 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 of the system is generated from it in two steps. In the first step, the model is realized by several functional component blocks, e.g., adder, integrator, etc., using analog computation techniques leading to an optimal functional topology of the system. In the second step, the functional component blocks are realized following different circuit-level implementation styles like switch capacitor, active RC, etc., leading to an optimal component-level topology of the system. This two-step process gives freedom to users in selecting appropriate implementation styles. The generated component-level topology is then behaviorally simulated to check whether it satis-

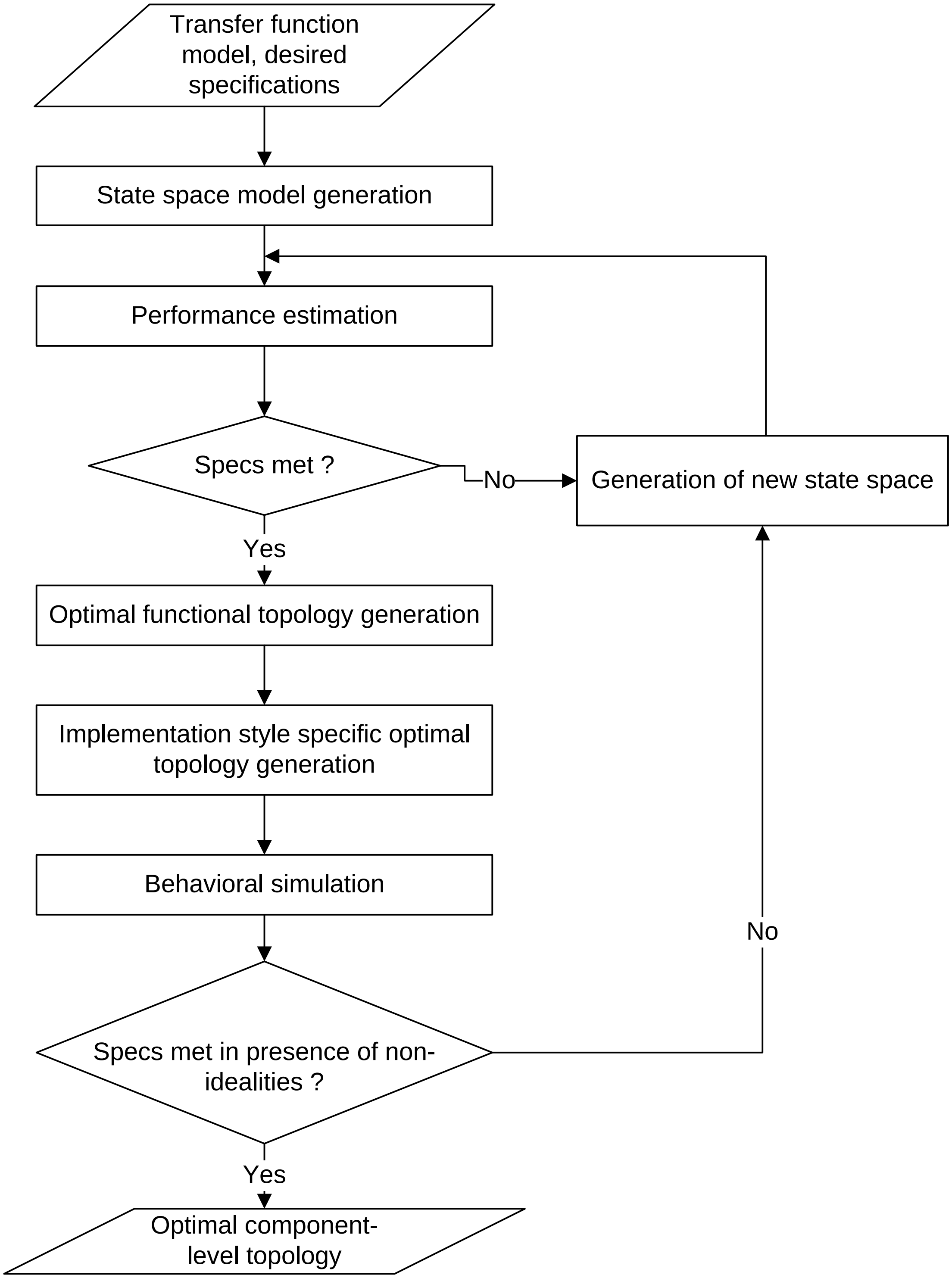

Transfer function model, desired specifications
State space model generation
Performance estimation
Specs met ?
No
Generation of new state space
Yes
Optimal functional topology generation
Implementation style specific optimal topology generation
Behavioral simulation
No
Specs met in presence of non-idealities ?
Yes
Optimal component-level topology

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

fies all of the desired specifications in presence of circuit-level non-idealities. If it fails, the topology exploration and selection steps are repeated and a new optimized state space model is generated. The final output of the generation process is an optimal component-level topology of the system which satisfies all of the desired specifications under circuit-level non-ideal conditions.

4.1.2 Transfer Function and State Space Representation

The general behavioral level representation for a continuous time linear system is expressed as an s-domain transfer function, defined as [67]

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

A transfer function however, does not provide any information concerning the physical structure of the system. In the time domain, this is equivalently described by a state space model. The state space model of a system consists of a set of differential equations that describes the internal and terminal behavior of the system [67]. A state space model for continuous time linear system is defined as [67] [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 the 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 transfer functions and the state space equations are the two equivalent ways of modeling a continuous time system. The relationship between these two representations is given as [67]

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

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 the characteristic equation

---

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

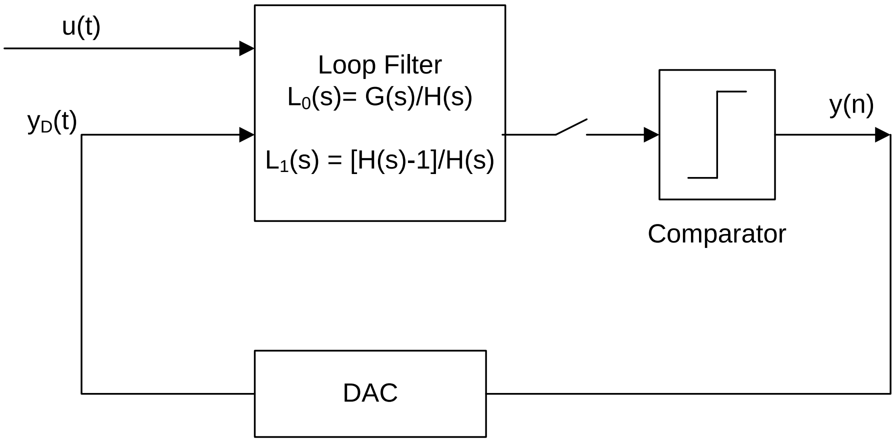


Figure 4.2: Block diagram of a CT ΣΔ modulator

plays a major role in determing the behavioral properties. From (4.4), this is defined as

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

The roots of the characteristic equation are often referred to as the eigen values of the matrix **A**. These correspond to the poles of the system.

For illustrating a linear analog system, we consider a continuous time ΣΔ modulator system. The block diagram shown in Fig. 4.2 describes in general all single-bit modulators [68]. It is splitted into a linear block (the loop filter) and a non-linear 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. Many popular ΣΔ modulator topologies, such as cascade integrators with feedback and feedforward, cascade of resonators, etc., [68] are well represented by state space models. Cascaded ΣΔ modulators can also be modeled in the same way by treating each section in the above way [69]. However, in practical cases, there are some designs which cannot be modeled through state space equations [70]. Automated synthesis of these topologies however, do not come under the scope of the present methodology.

4.1.3 Topology Generation

The topology $\mathcal{T}$ (more specifically, the component-level topology) of a system is a block diagram description of the system consisting of several component blocks. The topology generation process is carried out in two steps. In the first step, a functional topology $\mathcal{T}_F$ of the system is generated from the given transfer functions via the state space matrix model and in the second step the component-level topology $\mathcal{T}$ is realized from $\mathcal{T}_F$. In the following sub-sections, we discuss each of them in detail.

4.1.3.1 Functional Topology Generation

The functional topology $\mathcal{T}_F$ is an interconnection of functional component blocks, e.g., adder, integrator, etc., realizing the desired functionality of the system and is independent of any circuit-level implementation style. The functional topology for a linear analog system is mathematically represented by a state space model $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$. The functional topology generation procedure is based upon the following two principles of analog computation theory [71]:

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 is required to implement a state x_i and another set of $(n+p)$ input-weighted adders is required to implement an output y_i.

As a practical example, let us consider the following state space model of a $\Sigma\Delta$ modulator system

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

The functional topology corresponding to this state space model is shown in Fig. 4.3 which is a widely used practical $\Sigma\Delta$ modulator topology (chain of integrator with distributed feedback and distributed feedforward inputs).

The functional topology of an n^{th} order $\Sigma\Delta$ modulator with two inputs and single output, constructed on the basis of these two principles is shown in Fig. 4.4. In the figure, the integrators are represented by $1/s$ blocks, $x_1, x_2, ..., x_n$ are the state

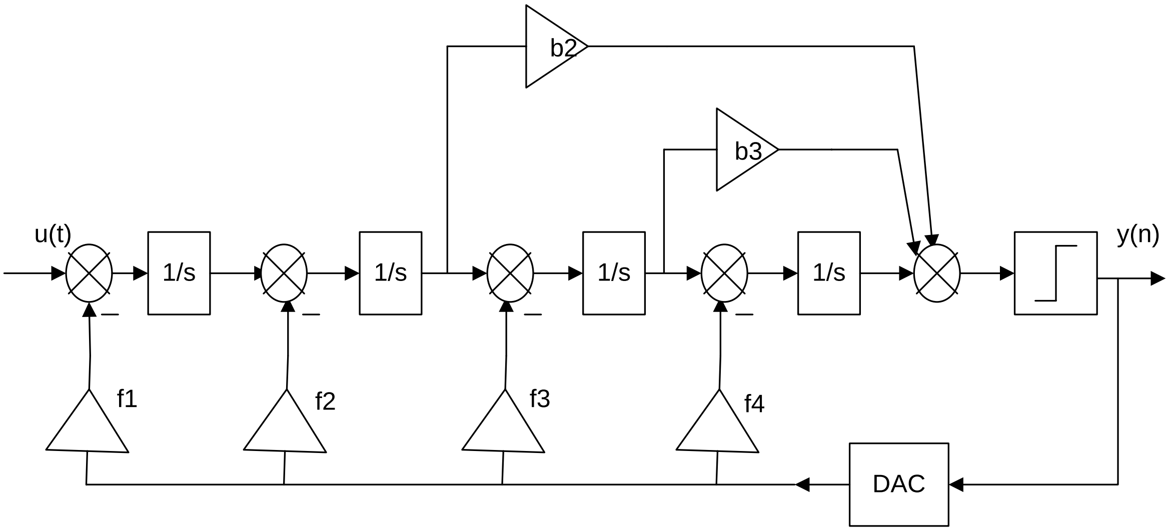


Figure 4.3: Functional topology of a 4^{th} order CT ΣΔ modulator.

variables, $a_{11}, ..., a_{nn}, b_{11}, ..., b_{n1}, b_{12}, ..., b_{n2}, c_1, ..., c_n, d_1, d_2$ are the signal-path coefficients of the modulator and correspond 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.

4.1.3.2 Component-Level Topology Generation

The component-level topology $\mathcal{T}$ of the system is synthesized by replacing each of the component blocks of the functional topology $\mathcal{T}_F$ with circuit-level implementation style specific realization. The commonly used implementation styles are operational transconductance amplifier(OTA)-capacitor, active RC and switch capacitor. At this step, topologies can be generated in competing technologies and evaluated during design space exploration. This provides flexibility to the users in selecting an implementation style that is most appropriate for realizing the system requirements.

In the present work, the OTA-C implementation style [72] has been chosen. In this style, the integrators are implemented by adding capacitors to the outputs of the OTAs. Adders are simple nodes, where current addition takes place. The scalars are implemented by simple OTAs, converting the input voltage signals to output current signals [72]. The mapping of the matrix elements to the transconductance (Gm) values of the OTAs are given by the following relationships [73]:

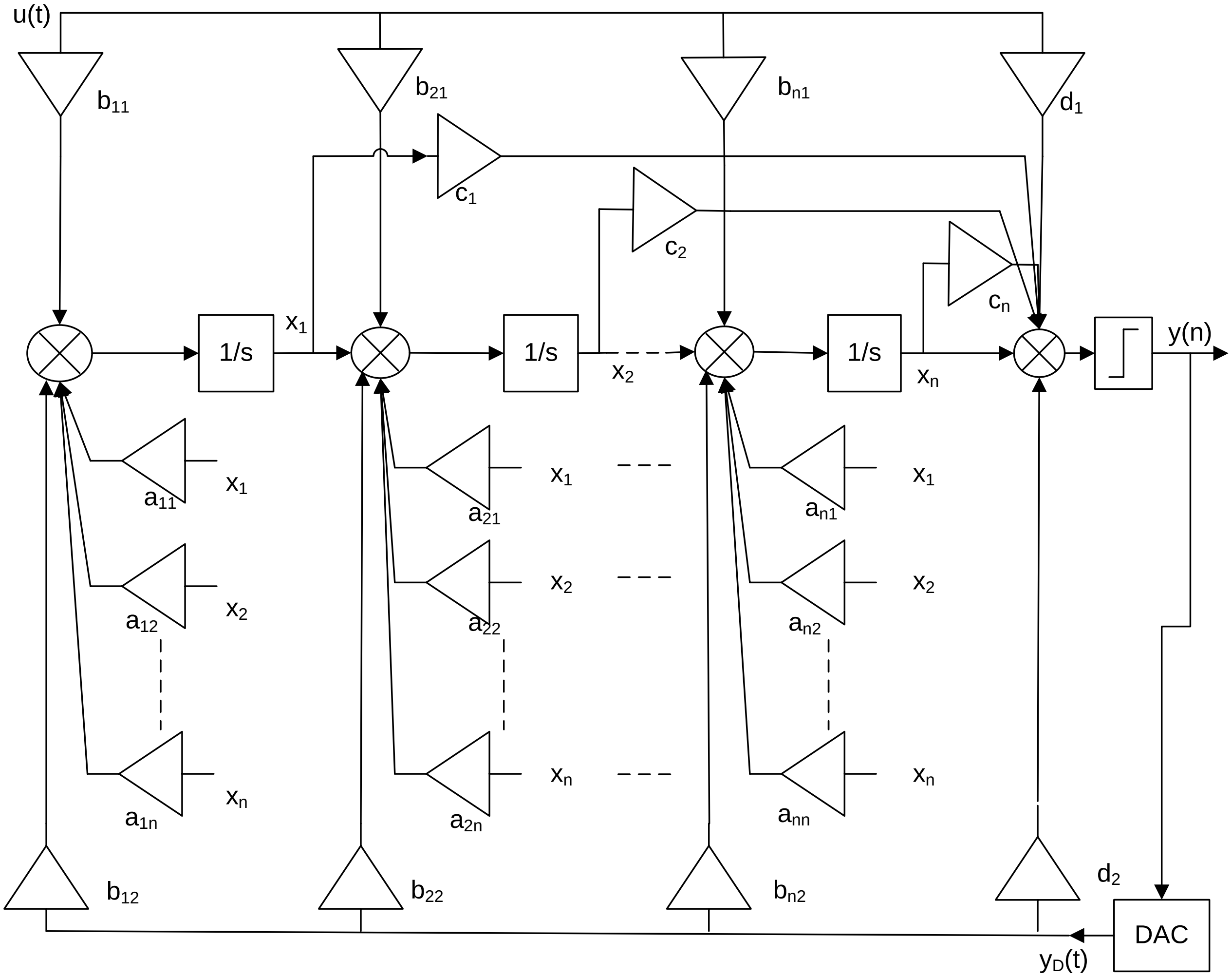


Figure 4.4: Functional topology for a general n^{th} order CT $\Sigma\Delta$ modulator

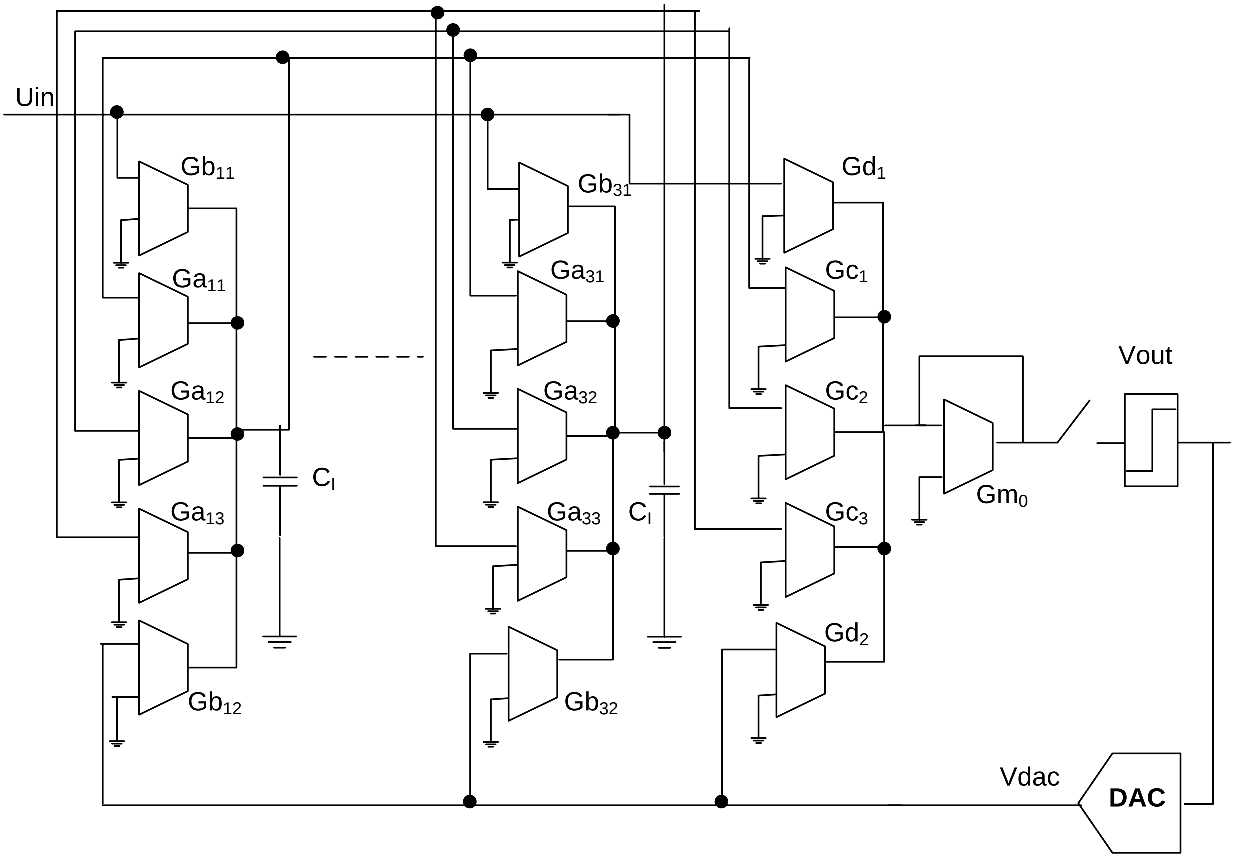


Figure 4.5: A generic 3^{rd} order CT modulator with GmC loop-filter. $(Ga_{ij} \equiv Gma_{ij})$

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

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, $Gma_{ij}, Gmb_{ij}, Gmc_i, Gmd_i$ are the Gm values of the OTAs required to implement the matrix elements.

In terms of OTA and C, a generic 3^{rd} order CT modulator single-loop topology is shown in Fig. 4.5.

4.1.4 Topology Transformation

The topology transformation operation maps a topology $\mathcal{T}$ into a new topology $\bar{\mathcal{T}}$. In our methodology, this process is implemented at the state space matrix level.

The vector function $\mathbf{x}(.)$ describes the evolution of the basic internal variables of a state space functional topology, viz., the integrator outputs. The state vector of a system is defined as a minimal set of variables, such that a knowledge of these variables at any time t_0, and the information on the input signal subsequently applied, are sufficient to determine the state of the system at any time $t > t_0$. 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 [74]. Using this transformation, the transformed dynamic equations are written as

$$\begin{aligned} \frac{d\bar{\mathbf{x}}(t)}{dt} &= \bar{\mathbf{A}}(t)\bar{\mathbf{x}}(t) + \bar{\mathbf{B}}(t)\mathbf{u}(t) && (4.9) \\ \bar{\mathbf{y}}(t) &= \bar{\mathbf{C}}(t)\bar{\mathbf{x}}(t) + \bar{\mathbf{D}}(t)\mathbf{u}(t) && (4.10) \end{aligned}$$

Thus it is possible to generate a new state space model $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathbf{D}})$ from an old model $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ through similarity transformation. The relationship between the original and the new state space matrices is

$$\begin{aligned} \bar{\mathbf{A}} &= \mathbf{T}^{-1}\mathbf{A}\mathbf{T} && (4.11) \\ \bar{\mathbf{B}} &= \mathbf{T}^{-1}\mathbf{B} && (4.12) \\ \bar{\mathbf{C}} &= \mathbf{C}\mathbf{T} && (4.13) \\ \bar{\mathbf{D}} &= \mathbf{D} && (4.14) \end{aligned}$$

The similarity transformation matrix

$$\mathbf{T} \in \Re^{n\times n}, \;\; \det(\mathbf{T}) \neq 0 : \mathcal{T} \rightarrow \bar{\mathcal{T}} \tag{4.15}$$

which transforms a topology $\mathcal{T}$ into a new topology $\bar{\mathcal{T}}$ is used as the topology transformation operator [2]. Since there exists an infinite number of non-singular matrices, an infinite number of state space models can be generated from a given one. Accordingly an infinite number of topologies can be generated. [3]

---

[2] More correctly, the similarity transformation matrix transforms a functional topology $\mathcal{T}_F$ to a new functional topology $\bar{\mathcal{T}}_F$

[3] However, it is to be noted that it is not always possible to obtain any desired topology, even

As an example of the topology transformation process, consider a 3^{rd} order $\Sigma\Delta$ modulator topology described by the following state space model:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0.3125 & 0 & -0.02313 \\ 0 & 0.0625 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 5.643 & -7.525 \\ 0.08819 & -1.325 \\ 0 & -0.2339 \end{bmatrix} \tag{4.16}$$

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

Let an arbitrary, non-singular similarity transformation matrix be

$$\mathbf{T} = \begin{bmatrix} -0.9635 & -3.8779 & -2.9476 \\ -2.4620 & 2.1874 & -0.6847 \\ 1.8864 & -1.1728 & -1.6800 \end{bmatrix} \tag{4.17}$$

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

$$\bar{\mathbf{A}} = \begin{bmatrix} -0.0405 & 0.0528 & -0.007813 \\ -0.05323 & 0.007813 & -0.02291 \\ 0.08328 & -0.02754 & 0.03269 \end{bmatrix} \quad \bar{\mathbf{B}} = \begin{bmatrix} -0.8802 & 1.228 \\ -1.034 & 1.027 \\ -0.2666 & 0.801 \end{bmatrix} \tag{4.18}$$

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

Under the topology transformation operation, the values of the state space matrix elements change. This leads to different performances (e.g., power consumption, input referred thermal noise) of the corresponding topologies. However, the behavioral properties (e.g., SNR/DR) of the topologies remain the same as governed by the following invariant property of similarity transformation.

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

---

of the same order as the original, by a similarity transformation. This is a purely algebraic fact and the discussion is beyond the scope of this dissertation.

Table 4.1: Non-idealities of the OTA-C component blocks

Blocks	Non-idealities
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

Proof The characteristic equation is written as

$$\left|s\mathbf{I}-\bar{\mathbf{A}}\right| = \left|s\mathbf{I}-\mathbf{T}^{-1}\mathbf{A}\mathbf{T}\right| = \left|s\mathbf{T}^{-1}\mathbf{T}-\mathbf{T}^{-1}\mathbf{A}\mathbf{T}\right| \tag{4.19}$$

Since the determinant of a product matrix is equal to the product of the determinants of the matrices, (4.19) becomes

$$\left|s\mathbf{I}-\bar{\mathbf{A}}\right| = \left|\mathbf{T}^{-1}\right|\left|s\mathbf{I}-\mathbf{A}\right|\left|\mathbf{T}\right| = \left|s\mathbf{I}-\mathbf{A}\right| \tag{4.20}$$

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

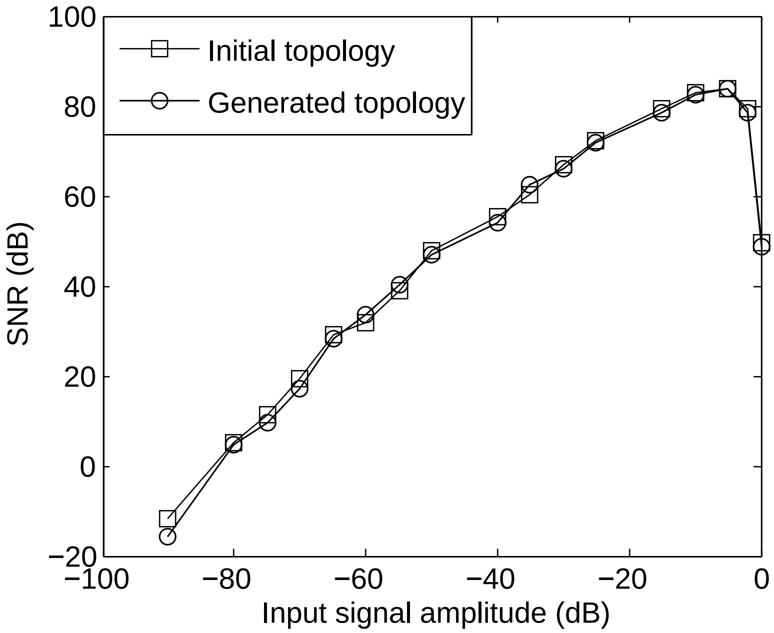
$$\begin{aligned} \bar{\mathbf{L}}(s) &= \bar{\mathbf{C}}(s\mathbf{I}-\bar{\mathbf{A}})\bar{\mathbf{B}}+\bar{\mathbf{D}} && (4.21)\\ &= \mathbf{C}\mathbf{T}(s\mathbf{I}-\mathbf{T}^{-1}\mathbf{A}\mathbf{T})\mathbf{T}^{-1}\mathbf{B}+\mathbf{D} && (4.22) \end{aligned}$$

which is simplified to

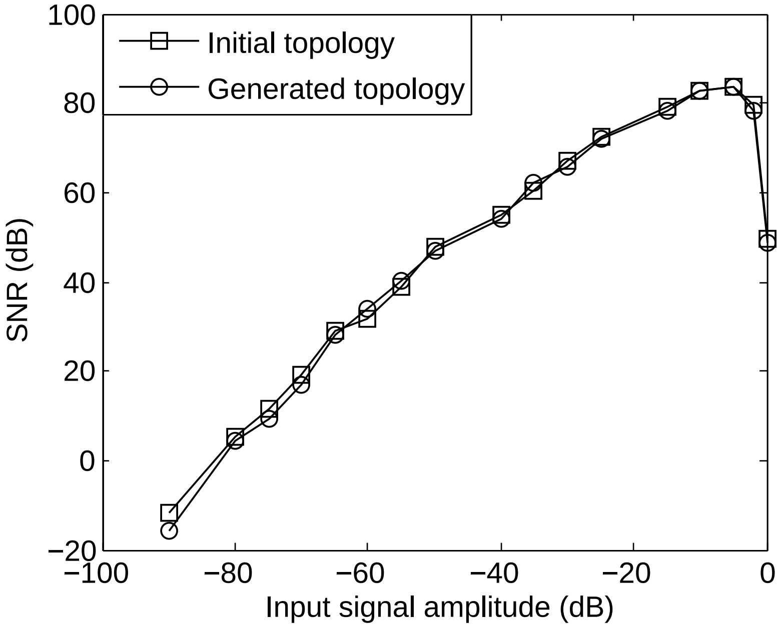
$$\bar{\mathbf{L}}(s) = \mathbf{C}(s\mathbf{I}-\mathbf{A}^{-1})\mathbf{B}+\mathbf{D} = \mathbf{L}(s) \tag{4.23}$$

Since the transfer function remains invariant under the transformation, the SNR/DR remains invariant under it.

Under non-ideal conditions the invariance property is verified through behavioral simulation as follows. We consider the 3^{rd} order topologies given by (4.16) and (4.18). These are behaviorally simulated. For behavioral simulation purpose, the OTAs are replaced by the appropriate behavioral models. The behavioral models as described in [40, 41] are considered for behavioral simulation purpose. The incorporated non-



(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 topology transformation, establishing the invariant property.

idealities are listed in Table 4.1. The *SNR*s of the two 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 ranges of both the topologies under non-ideal conditions are almost equal with a difference less than 1%. We repeat the same experiment for a 4^{th} order modulator. In this case also, we observe that the *DR* values are nearly the 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 even under non-ideal conditions.

4.1.5 Performance Estimation

Since the topologies (as state space models) generated through the topology transformation operation have different performance properties (e.g., power consumption), performance estimation models are required to be constructed for evaluating the performances of the topologies. An optimal topology (state space model) needs to be selected based on the performances of the topologies. During a high-level topology generation process, detailed knowledge about the circuit-level implementation of the component blocks of a topology is not known. Therefore, 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.[4] 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) relative power consumption. The process of estimation of these are discussed below.

4.1.5.1 Sensitivity minimization

The sensitivity of the output of a system to topology parameter variations is an important performance metric for analog systems. The *Gm* values of the OTAs required to implement the OTA-C topology of Fig. 4.5 serve as the topology parameters. These are related to the state space matrix elements through (4.7). The *Gm* values of the OTAs in the fabricated chip can never match the desired transcon-

---

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

ductance values exactly due to limitations of the fabrication process. Moreover due to different environmental effects the Gm values of the OTA may change from time to time. So the variation of the output of a system must be minimal due to topology parameter variation. For a topology $\mathcal{T}_F(\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D})$ with transfer function $\mathbf{L}(s)$, we define the sensitivity of $\mathbf{L}(s)$ with respect to the elements of the 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 $\mathbf{L}(s)$ to the coefficients of $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ as [75, 76]

$$\begin{aligned} S_{\mathbf{A}}(s) &\triangleq \frac{\partial \mathbf{L}}{\partial \mathbf{A}} = \mathbf{G}(s)\mathbf{F}^T(s) && (4.24)\\ S_{\mathbf{B}}(s) &\triangleq \frac{\partial \mathbf{L}}{\partial \mathbf{B}} = \mathbf{G}(s) && (4.25)\\ S_{\mathbf{C}}(s) &\triangleq \frac{\partial \mathbf{L}}{\partial \mathbf{C}^T} = \mathbf{F}(s) && (4.26)\\ S_{\mathbf{D}}(s) &\triangleq \frac{\partial \mathbf{L}}{\partial \mathbf{D}} = 1 && (4.27) \end{aligned}$$

where $\mathbf{F}(s)$ and $\mathbf{G}(s)$ are two intermediate transfer functions defined as

$$\begin{aligned} \mathbf{F}(s) &= (s\mathbf{I}-\mathbf{A})^{-1}\mathbf{B} && (4.28)\\ \mathbf{G}(s) &= \mathbf{C}\,(s\mathbf{I}-\mathbf{A})^{-1} && (4.29) \end{aligned}$$

We note that the above sensitivity functions are matrix functions of the complex variable 's'. A further definition is required to enable the measurement of 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 [75] as follows

$$S_{L12} = \left|\left|\frac{\partial \mathbf{L}}{\partial \mathbf{A}}\right|\right|_1^2 + \left|\left|\frac{\partial \mathbf{L}}{\partial \mathbf{B}}\right|\right|_2^2 + \left|\left|\frac{\partial \mathbf{L}}{\partial \mathbf{C}^T}\right|\right|_2^2 \tag{4.30}$$

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

$$||\mathbf{f}(j\omega)|| = \left(\frac{1}{2\pi}\int_0^{2\pi} ||\mathbf{f}(j\omega)||_F^p \, d\omega\right)^{\frac{1}{p}} \tag{4.31}$$

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

$$||\mathbf{f}(j\omega)||_F = \left\{tr\left[\mathbf{f}^T(-j\omega)\mathbf{f}(j\omega)\right]\right\}^{\frac{1}{2}} \tag{4.32}$$

In this equation $tr(\mathbf{Y})$ means the trace of the matrix $\mathbf{Y}$. It is to be noted that in (4.30), 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.

The direct evaluation of the term $||\frac{\partial\mathbf{L}}{\partial\mathbf{A}}||_1^2$ 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 [76]. In addition, it was shown [76] that the solution that minimizes the upper bound also happens to minimize the measure S_{L12} itself. Using Cauchy-Schwartz inequality the first term in the right hand side of (4.30) is written as

$$\left|\left|\frac{\partial\mathbf{L}}{\partial\mathbf{A}}\right|\right|_1^2 \leq \left|\left|\frac{\partial\mathbf{L}}{\partial\mathbf{B}}\right|\right|_2^2 \left|\left|\frac{\partial\mathbf{L}}{\partial\mathbf{C}^T}\right|\right|_2^2 \tag{4.33}$$

With these the defining equation for the sensitivity becomes

$$\begin{aligned}\tilde{S}_{L12} &= \left|\left|\frac{\partial\mathbf{L}}{\partial\mathbf{B}}\right|\right|_2^2 \left|\left|\frac{\partial\mathbf{L}}{\partial\mathbf{C}^T}\right|\right|_2^2 + \left|\left|\frac{\partial\mathbf{L}}{\partial\mathbf{B}}\right|\right|_2^2 + \left|\left|\frac{\partial\mathbf{L}}{\partial\mathbf{C}^T}\right|\right|_2^2 && (4.34)\\ &= ||\mathbf{G}(s)||_2^2\,||\mathbf{F}(s)||_2^2 + ||\mathbf{G}(s)||_2^2 + ||\mathbf{F}(s)||_2^2 && (4.35)\end{aligned}$$

where $\tilde{S}_{L12}$ is the upper bound of S_{L12}. It 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 the controllability Gramian matrix $\mathbf{K}$ are defined as [67]

$$\begin{aligned}\mathbf{W} &= \frac{1}{2\pi}\int_{-\infty}^{+\infty} \mathbf{G}^*\mathbf{G}\,d\omega && (4.36)\\ \mathbf{K} &= \frac{1}{2\pi}\int_{-\infty}^{+\infty} \mathbf{F}\mathbf{F}^*\,d\omega && (4.37)\end{aligned}$$

The asterisk (*) denotes the adjoint operator, i.e., if $\mathbf{Y}$ be 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 [67]:

$$\mathbf{A}^T\mathbf{W} + \mathbf{W}\mathbf{A} = -\mathbf{C}^T\mathbf{C} \tag{4.38}$$

$$\mathbf{A}\mathbf{K} + \mathbf{K}\mathbf{A}^T = -\mathbf{B}\mathbf{B}^T \tag{4.39}$$

Using these Lyapunov equations, the relationships between the intermediate transfer functions and the Gramian matrices are given as: [75]

$$||\mathbf{G}(s)||_2^2 = tr(\mathbf{W}) \tag{4.40}$$

$$||\mathbf{F}(s)||_2^2 = tr(\mathbf{K}) \tag{4.41}$$

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

$$\tilde{S}_{L12} = tr(\mathbf{W})tr(\mathbf{K}) + tr(\mathbf{W}) + tr(\mathbf{K}) \tag{4.42}$$

Under similarity transformation operation, 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

$$\tilde{S}_{L12}(\mathbf{T}) = tr\left(\mathbf{T}^T\mathbf{W}\mathbf{T}\right) tr\left(\mathbf{T}^{-1}\mathbf{K}\mathbf{T}^{-T}\right) + tr\left(\mathbf{T}^T\mathbf{W}\mathbf{T}\right) + tr\left(\mathbf{T}^{-1}\mathbf{K}\mathbf{T}^{-T}\right) \tag{4.43}$$

Thus the L_1/L_2 norm sensitivity for the various topologies are estimated by evaluating the expression in (4.43). This is easy to evaluate using the Matlab control system toolbox.

4.1.5.2 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 topology which means reduction in the number of OTAs. The following cost function is used as a measure of the hardware complexity

$$X = \sum_{k=1}^{(n+1)(n+2)} h_k \tag{4.44}$$

where, 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

$$h_k = \begin{cases} 0 & \text{if } m_k < \epsilon \\ 1 & \text{otherwise} \end{cases} \tag{4.45}$$

where ϵ represents the lower bound of the acceptable range of the matrix elements.

4.1.5.3 Relative Power Consumption

Without considering the circuit-level implementation details of the component-blocks it is very difficult to construct an estimator which accurately estimates the real power consumption of a system. Thus for topology exploration purpose it is not necessary to estimate the exact power consumption [8]. The estimator needs to measure correctly the change in power consumption of a topology with change in the design parameter values of the topology [8]. Such an estimator is often referred to as a relative power estimator. Depending upon the chosen application system, the relative power estimator has to be constructed and fed in the process.

For the $\Sigma\Delta$ modulator system, a commonly used assumption for constructing a power estimator is that the largest part of the power consumption of the modulator is determined by the OTAs [22, 38]. This accounts for the static power consumption. This is given as [77]

$$P \approx n_I I_b V_{DD} M \tag{4.46}$$

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 an exact estimation of the power consumption, these parameters should be known along with n_I. However, for tracking purpose, the power consumption of a single OTA can be considered to be proportional to Gm^2. With this, the relative power estimator is given by

$$P \approx \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^2 C_I^2 + \sum_{i=1}^{n}\sum_{j=1}^{2} b_{ij}^2 C_I^2 + \sum_{i=1}^{n} c_i^2 Gm_0^2 + \sum_{i=1}^{n} d_i^2 Gm_0^2 \tag{4.47}$$

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

4.1.6 Topology Exploration and Selection

The topology exploration and selection processes are implemented at the state space matrix level. These processes determine an optimal state space model based on the estimated performances of the models. These two processes are combined through a simulated annealing (SA)-based optimization process. The topology space $\mathbf{V}(\mathcal{F})$ is explored by the SA [5] algorithm which selects an optimal topology as a solution point. The exploration points are generated through the topology transformation operator $\mathbf{T}$. The elements of this matrix are restricted within a definite range of real numbers. The performance estimators discussed in Section 4.1.5 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. 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 decrease in cost function value.

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

$$\begin{aligned} \text{Minimize} \quad & \Phi(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathbf{D}}) \\ \text{such that} \quad & g_1(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathbf{D}}) \leq 0,\ g_2(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathbf{D}}) \leq 0 \\ \text{and} \quad & det(\mathbf{T}) \neq 0 \end{aligned} \tag{4.48}$$

The overall cost function Φ is given as the sum of the three individual cost functions (4.43),(4.44) and (4.47). $g_i(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathbf{D}})$ is a feasibility constraint. Depending upon the chosen application system, different types of feasibility constraints are to be formulated.

For the $\Sigma\Delta$ modulator case study, the following feasibility constraints have been formulated.

4.1.6.1 Feasible Gm Value Constraints

Since it is difficult to realize a too high Gm value or a too low Gm value, constraints are added within the exploration procedure to ensure that for the generated topology,

---

[5] The details of the SA algorithm have been discussed in Appendix B.2 of this dissertation.

the required OTA values lie within a certain limit. The feasibility constraints for the state space coefficients are therefore explicitly written as

$$Gm_{min} \leq (|a_{ij}|\, C_I, |b_{ij}|\, C_I, |c_i|\, Gm_0, |d_i|\, Gm_0) \leq Gm_{max} \tag{4.49}$$

4.1.6.2 Non-overload Constraints

For a stable $\Sigma\Delta$ 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 constraints under normal operating conditions require that the matrix elements must satisfy certain constraints. These are formulated from (4.2) and (4.3) following the approach of [78] as follows.

$$\left|(j\omega\mathbf{I} - \bar{\mathbf{A}})^{-1}\bar{\mathbf{B}} \left[\begin{array}{cc} U_{max} & V_{ref} \end{array} \right]^T \right| \leq PL_1 \tag{4.50}$$

$$\left|L_1\bar{\mathbf{C}} + \bar{\mathbf{D}} \left[\begin{array}{cc} U_{max} & V_{ref} \end{array} \right]^T \right| \leq PL_2 \tag{4.51}$$

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 the modulator order and $P > 1$. The reason to have a scaled version of L_1 and L_2 is to account for an over-estimation of worst case analysis [26]. A lower P is more likely to avoid overloading, but also more likely to over-constrain the coefficient variables. The exact value of 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.

4.1.7 Complete Flow of the Topology Generation Process for CT $\Sigma\Delta$ Modulator System

In this sub-section, we first present a brief survey of literature related to $\Sigma\Delta$ modulator synthesis and then describe the complete flow of the present methodology for generation of an optimal component-level topology for CT $\Sigma\Delta$ modulator system.

4.1.7.1 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 concentrates primarily on the optimization of modulator coefficients [20, 79, 80, 81]. 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 the designer's experience. Then the coefficients of the selected topology are calculated to optimize the modulator performances such as the 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 [22, 23, 24], a set of selected topologies with optimized coefficients is 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 [26, 41, 82, 83] has come up recently. The tasks of topology generation, exploration and selection are the primary focus of these works. In [26], a generic representation of a 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 is obtained. The topology generation problem is formulated as a mixed-integer non-linearly 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 combinations of integrator types and a set of local solutions is obtained. At the last stage, these are then checked for minimum signal path, sensitivity through Monte Carlo analysis and power consumption, which finally yield a global solution. This basic procedure is extended to develop a systematic methodology for designing reconfigurable continuous time $\Sigma\Delta$ modulator topologies in [41, 82]. This category of works follows the 'selection during sizing' principle discussed in Section 2.4.2 of this dissertation.

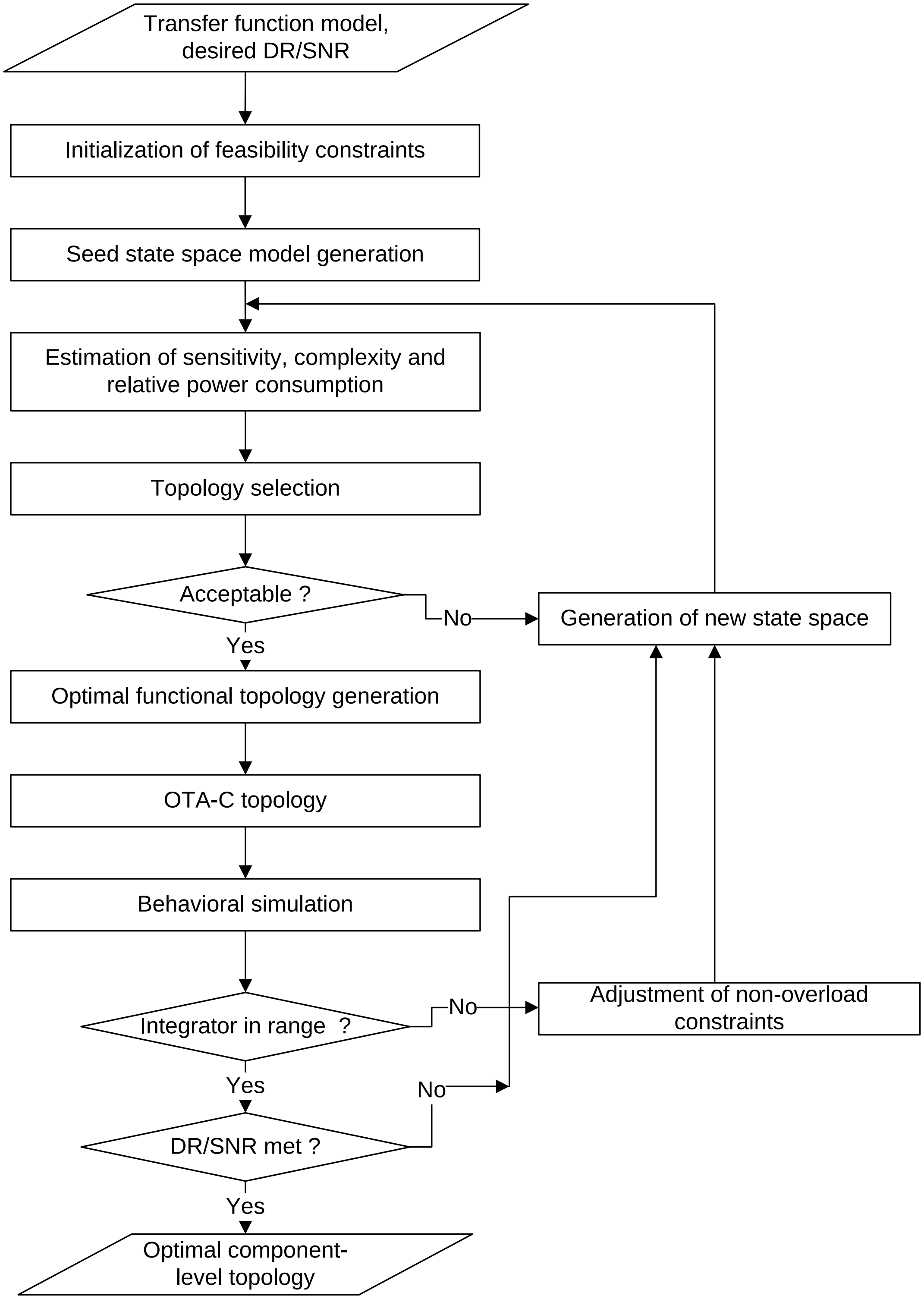

Transfer function model, desired DR/SNR
Initialization of feasibility constraints
Seed state space model generation
Estimation of sensitivity, complexity and relative power consumption
Topology selection
Acceptable ?
No
Generation of new state space
Yes
Optimal functional topology generation
OTA-C topology
Behavioral simulation
Integrator in range ?
No
Adjustment of non-overload constraints
Yes
No
DR/SNR met ?
Yes
Optimal component-level topology

Figure 4.7: Complete flow of the topology generation process for $\Sigma\Delta$ modulator system.

4.1.7.2 Methodology

A complete flow of the topology generation process is shown in Fig. 4.7. The inputs to the process are transfer function models of the system, a set of design constraints and the desired *DR*/*SNR*. The transfer functions include the feedforward and feedback transfer functions, i.e., $L_0(s)$ and $L_1(s)$ of the loop filter. The design constraints include the maximum input signal amplitude, DAC voltage etc. The feasibility constraints are initialized next. The non-overload scaling factor P is heuristically initialized to half of the modulator order. From the given transfer functions, a seed state space is generated. The sensitivity response, hardware complexity and relative power consumption of the state space model are estimated. 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 and the feasibility, non-overload constraints are satisfied. The similarity transformation matrix **T** is used to generate a new state space model from the seed state space model. From the optimized state space model, an optimal functional topology is generated using analog computation principles. From this an optimal OTA-C component-level topology is generated. The generated 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 are checked to see whether clipping occurs. If yes, the non-overload condition is adjusted by reducing the scale factor and the process is iterated by generating a new state space model. 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 specifications 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.

4.2 Comparison with Existing Methodologies

The methodologies closely related with the present methodology have been described in [7, 26, 27, 29]. In [26], a generic template-based methodology has been described for generation of an optimal topology for discrete time $\Sigma\Delta$ modulator system. The other works consider a general analog system. We present a comparison between our methodology and the existing methodologies below.

1. The component-level topology generation methodology in [7, 27] 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 functional topology. Such description is then converted to a signal flow graph, from which multiple component-level topologies are generated through mapping process or heuristic conversion rules. These methodologies therefore require an *a priori* knowledge of the functional topology of the system. On the other hand, our methodology generates component-level topologies from a transfer function model of the system. Therefore, an optimal functional topology is generated in our methodology, rather than assuming a definite functional topology of the system. Thus our methodology generates component-level topologies from a much higher level of abstraction compared to [7, 27].

2. The ARCHGEN methodology as described in [29] 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, the performances of the generated topology have not been verified under circuit-level non-idealities. Our methodology extends the ARCHGEN methodology by including a performance estimation and optimization procedure. In addition, our topology generation process includes a behavioral simulation-based checking process within the procedure and the generated topology is guaranteed to work satisfactorily even under circuit-level non-ideal conditions.

3. There is a major difference with our methodology for high-level design of $\Sigma\Delta$ modulator and that presented in [22, 24]. Both of these techniques deal with the task of optimal topology selection. On the other hand, our methodology deals with the task of optimal topology generation. In [22], a set of topologies with optimized coefficients are stored in a library. Given the design specification, e.g., *SNR* and *DR*, the procedure selects one from a set of pre-defined topologies with the smallest power consumption. Similar procedure has been reported in [24] also. With these methodologies, the topology control parameters are pre-defined which restrict the solution space for topology exploration.

4. The major difference between our methodology for generation of $\Sigma\Delta$ modulator topologies and that in [26] is that our methodology follows a top-down

approach in contrast to the template-based approach followed in [26]. The methodology in [26], starts with a generic template for ΣΔ modulator topology. In order to specialize the procedure for a modulator of fixed size, several steps of the procedure, e.g., derivation of the symbolic *NTF* and *STF* are to be carried out manually. This is a tedious and error prone task, especially for higher order modulators. On the other hand, in our methodology, the topologies are generated directly from the modulator transfer functions. Our topology generation process is free of any manual intervention between the steps. A fully automated implementation of our methodology is developed under Matlab environment, so that it can be used even by novice designers. Thus our methodology is advantageous over [26] from the view point of design automation.

4.3 Experimental Results

In this section, we provide experimental results demonstrating the methodology described above. The entire methodology has been implemented in Matlab. In order to verify the experimental results, the obtained results have been compared with both behavioral simulation results and SPICE simulation results as when required. The SPICE simulation results have been used as reference when one system simulation is required. Otherwise, the behavioral simulation results have been used as reference for verification purpose.

4.3.1 Experiment 1

We consider a 3^{rd} order modulator. The design specifications are: (i) $DR \geq 75$ dB . (ii) $U_{max} = 100$ kHz. (iii) $V_{ref} = 200$ mV. The feedforward path and feedback path normalized transfer functions of the loop-filter are

$$L_0(s) = \frac{0.0221\,(s+2)}{s\,(s^2+0.001446)} \tag{4.52}$$

$$L_1(s) = \frac{42.441\,(s^2+0.3541s+0.0628)}{(s+2)} \tag{4.53}$$

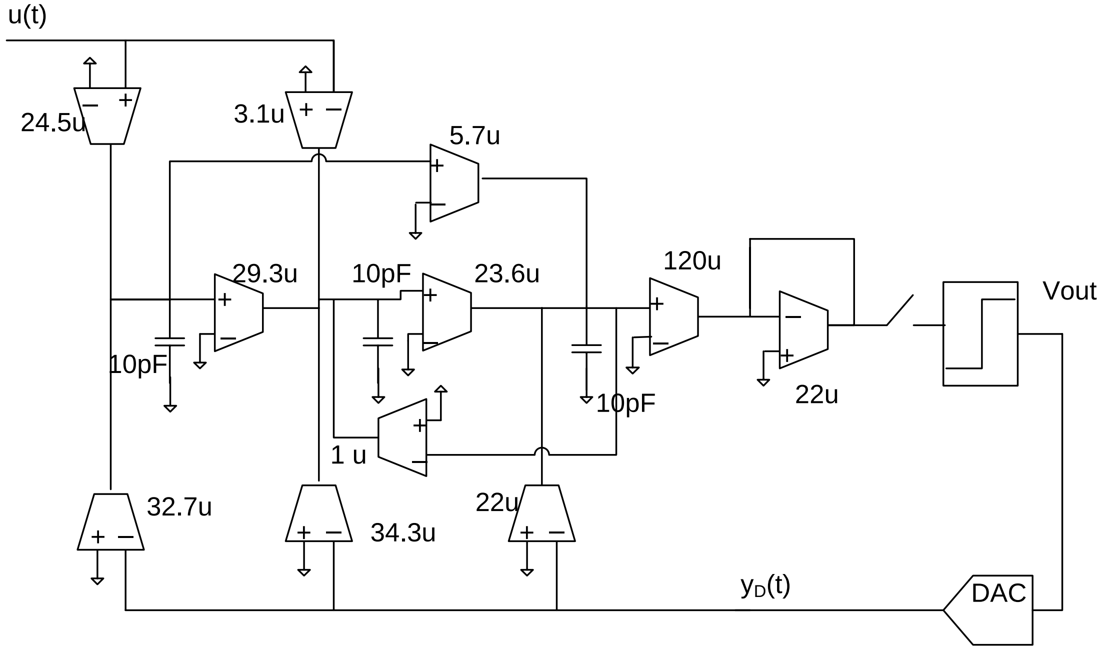


Figure 4.8: The generated 3^{rd} order modulator topology.

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

$$\mathbf{A}_{seed} = \begin{bmatrix} 0 & 0 & 0 \\ 0.3125 & 0 & -0.02313 \\ 0 & 0.0625 & 0 \end{bmatrix} \quad \mathbf{B}_{seed} = \begin{bmatrix} 5.643 & -7.525 \\ 0.08819 & -1.325 \\ 0 & -0.2339 \end{bmatrix} \tag{4.54}$$

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

Using this seed state space, the SA-based optimization process determines 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

Table 4.2: Non-idealities 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

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

Condition	DR	SNR peak
ideal	84 dB	83.1 dB
behavioral	80.91 dB	81.87 dB
SPICE	78.5	79.23

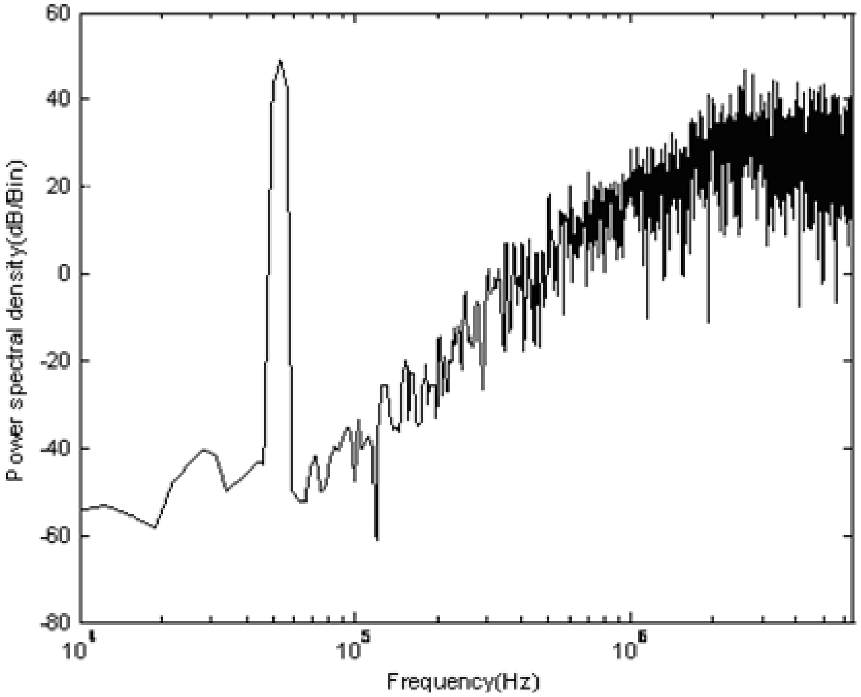
explored. The state space model corresponding to the generated topology is given below

$$\mathbf{A}_{gen} = \begin{bmatrix} 0 & 0 & 0 \\ 29.26 & 0 & -1 \\ 5.69 & 23.65 & 0 \end{bmatrix} \times 10^5 \quad \mathbf{B}_{gen} = \begin{bmatrix} 24.52 & -32.69 \\ -3.09 & -34.26 \\ 0 & -21.98 \end{bmatrix} \times 10^5 \tag{4.55}$$

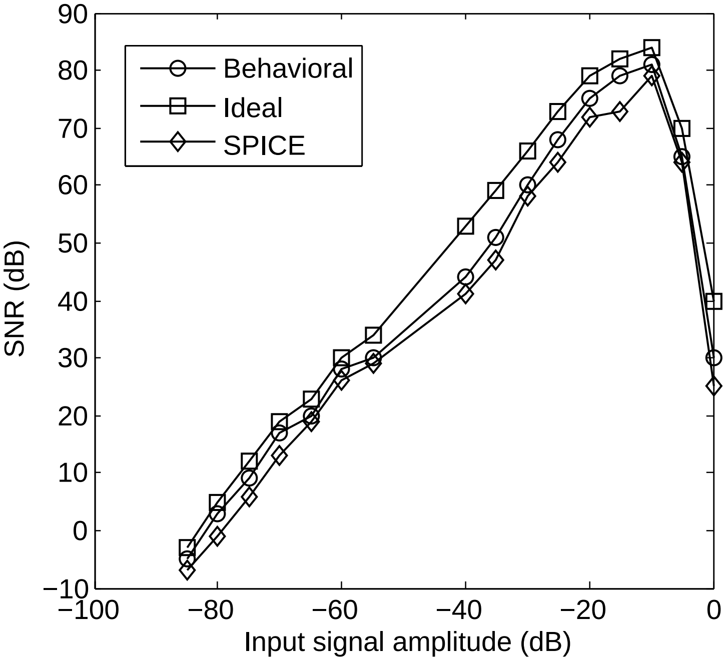
$$\mathbf{C}_{gen} = \begin{bmatrix} 0 & 0 & 5.45 \end{bmatrix} \qquad \mathbf{D}_{gen} = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

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.

To characterize the generated topology, we perform behavioral simulation including the non-idealities as listed in Table 4.2. The FFT-based noise power spectral density obtained from the simulation results is shown in Fig. 4.9(a). The variation of *SNR* against the input signal amplitude normalized to the maximum allowed, un-



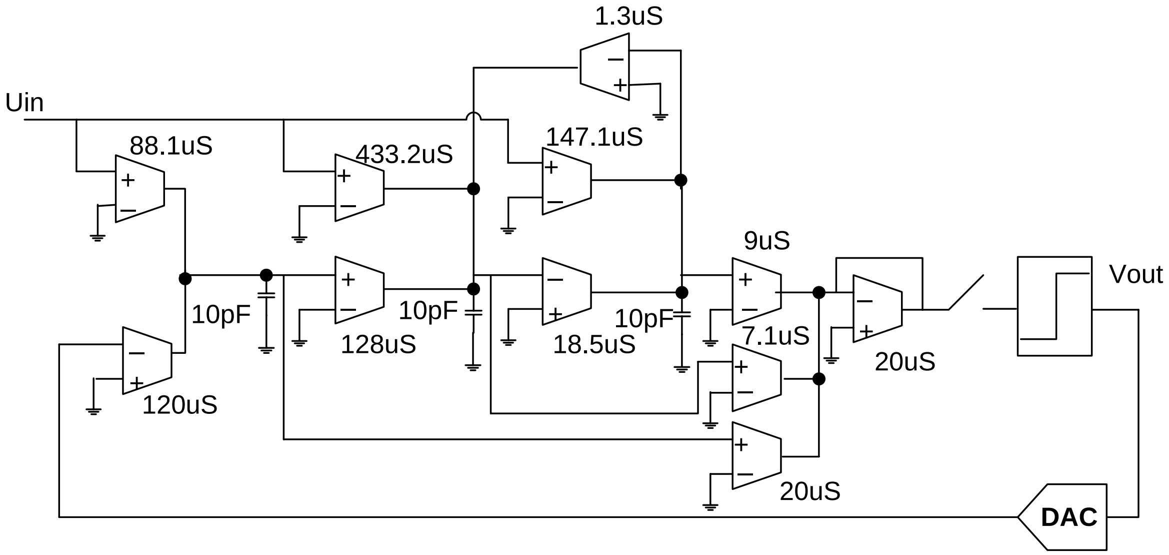
(a) Power spectral density for the generated 3^{rd} order modulator topology.



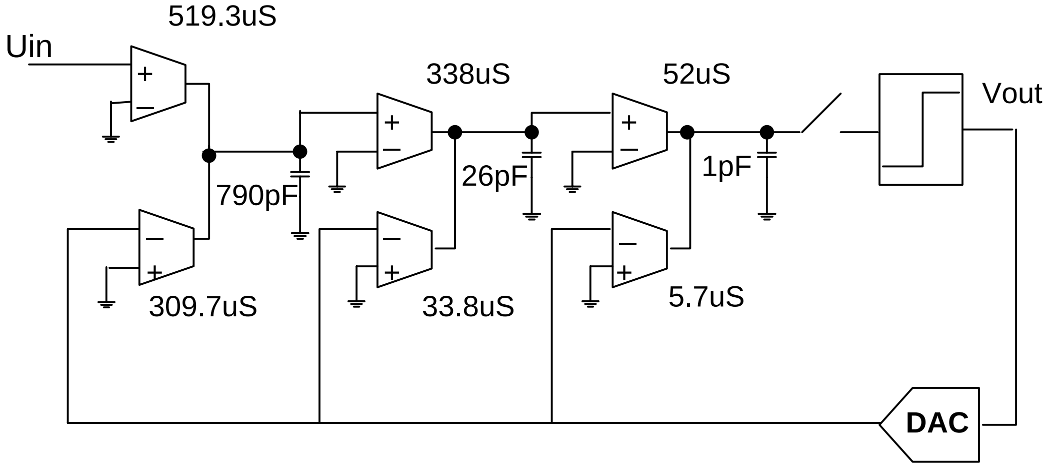
(b) Comparison of *SNR* plot for the generated topology under ideal, behavioral and SPICE simulation

Figure 4.9: Characterization of the generated 3^{rd} order modulator topology.

der ideal conditions as well as non-ideal conditions is shown in Fig. 4.9(b). From the figure, we make a comparison between the behavior of the generated topology under ideal, behavioral and SPICE simulation. These are listed in Table 4.3. We observe that both the *DR* and *SNR* (peak) obtained through SPICE simulation satisfies the desired specifications. This assures the quality of the generated topology.



(a) Cascaded integrator feedforward topology for 3^{rd} order modulator.



(b) Distributed feedback topology for 3^{rd} order modulator..

Figure 4.10: Chosen standard topologies of a 3^{rd} order modulator for comparison.

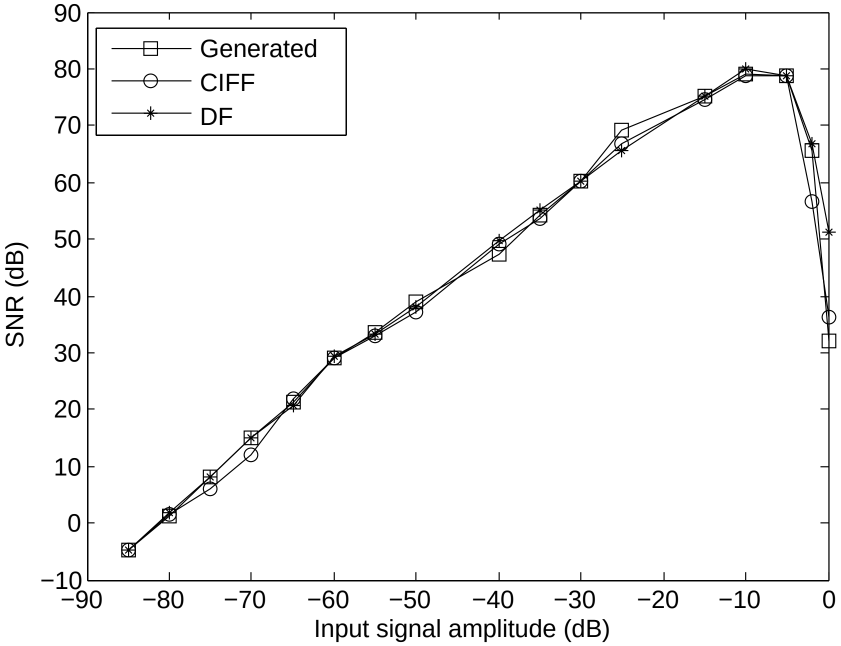


Figure 4.11: SNR/DR comparison between the generated, CIFF and DF topology.

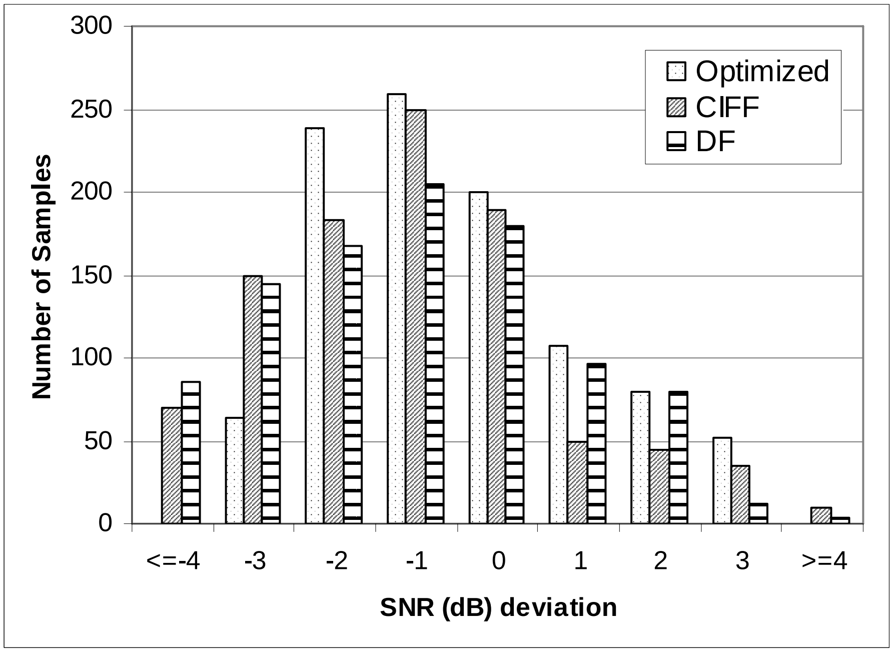
Table 4.4: Comparison in terms of yield for coefficient variation

Variation	Generated	CIFF	DF
± 2%	1000	982	975
± 5%	923	798	320
± 10%	536	380	107

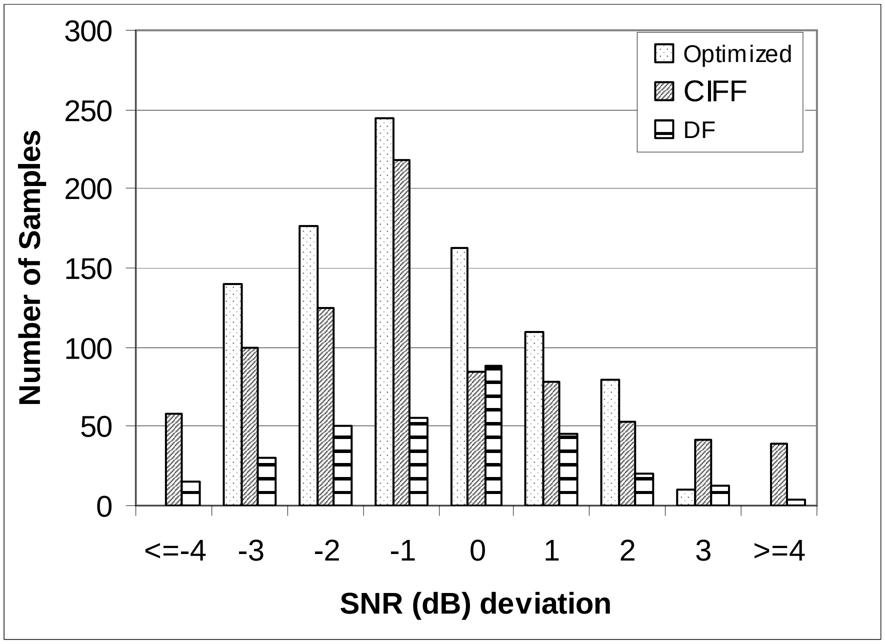
4.3.2 Experiment 2

In order to verify the optimality of the generated topology, we compare the performances of the generated topology with that of two standard topologies. Of them, one is referred to as the cascaded integrator feedforward (CIFF) topology and the other one as the distributed feedback (DF) topology [84]. These are shown in Fig. 4.10(a) and Fig. 4.10(b) respectively. The coefficients for these two standard topologies are manually calculated using Matlab $\Sigma\Delta$ toolbox [80] and DT-CT technique, described in [85, 68] such that they satisfy the desired specifications and obey the feasibility constraints. A comparison between the *SNR* plot of the generated topology and the standard topologies is shown in Fig. 4.11. This result shows that all the three topologies satisfy the desired DR/SNR specifications.

To compare the sensitivity characteristics of the generated topology, the CIFF



(a) Bar diagram of peak *SNR* deviation for 2% coefficient variation.



(b) Bar diagram of peak *SNR* deviation for 5% coefficient variation.

Figure 4.12: Monte Carlo analysis plot for sensitivity comparison between the generated (optimized), CIFF and DF topology

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 these three topologies are 1.23 , 2.34 and 3.82 respectively. However, to have a 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 a Gaussian distribution. We compare the change in the *SNR* value from the nominal value by applying an input signal with the 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 [26]. The first one is yield. This is equal to the total number of feasible samples among the 1000 runs. The samples are considered to be feasible if the fall of *SNR* from the nominal value lies within 5 dB. The second one is the deviation of *SNR* of the topologies from the nominal values for a fixed coefficient 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 generated 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 generated 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. 4.12(a) and Fig. 4.12(b) for 2 % and 5 % coefficient variation respectively. From Fig. 4.12(a), we observe that *SNR* deviation for the generated topology is concentrated within $|2\,\text{dB}|$ for 2% coefficient variation. On the other hand, for the other two topologies this is more than $|3\,\text{dB}|$. Similarly from Fig. 4.12(b), we observe that *SNR* deviation for the generated topology is concentrated within $|3\,\text{dB}|$ for 5% coefficient variation. On the other hand, for the other two topologies this is more than $|4\,\text{dB}|$. Thus we conclude that the generated topology is more tolerant to coefficient variation not only in terms of yield but also in terms of performance deviations.

A comparison between the generated, CIFF 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 this, it has high relative power consumption compared to the other two. The generated topology is

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

Metrics	Generated	CIFF	DF
relative power	1.967×10^{-8}	24.91×10^{-8}	48.31×10^{-8}
complexity	10	11	7

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

4.3.3 Experiment 3

In this case, we consider a 4^{th} order CT $\Sigma\Delta$ modulator. The design specifications are: (i) $DR \geq 90$ 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

$$L_0(s) = \frac{-0.00103\,(s^2-6)}{(s^2+2.78\times10^{-4})\,(s^2+1.79\times10^{-3})} \tag{4.56}$$

$$L_1(s) = \frac{-911.96\,(s+0.185)\,(s^2+0.1775s+0.04745)}{(s^2-6)} \tag{4.57}$$

The state space model of the generated 4^{th} order modulator topology is

$$\mathbf{A}_{gen} = \begin{bmatrix} 0 & -20.5 & 0 & 1.5 \\ 12.1 & 0 & 11.1 & 0 \\ 0 & 22.3 & 0 & -16.3 \\ -6.1 & 0 & 15.2 & 0 \end{bmatrix} \times 10^5 \quad \mathbf{B}_{gen} = \begin{bmatrix} 38.7 & -14.7 \\ 0 & -49.8 \\ 4.8 & -46.6 \\ 0 & -2.8 \end{bmatrix} \times 10^5 \tag{4.58}$$

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

The generated OTA-C topology is shown in Fig. 4.13(a). The load capacitances are taken as $10pF$. The Gm values are shown in the figure. The SNR plot of the generated 4^{th} order modulator topology under ideal and non-ideal conditions are shown in Fig. 4.13(b). The extracted DR/SNR parameters are reported in Table 4.6. From the results we see that the deviation of SNR/DR from the ideal value due to the chosen non-idealities is within an acceptable limit.

In order to compare the performances of the generated topology, we choose a 4^{th} order CIFF topology as shown in Fig.4.14. The coefficients and the corresponding

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

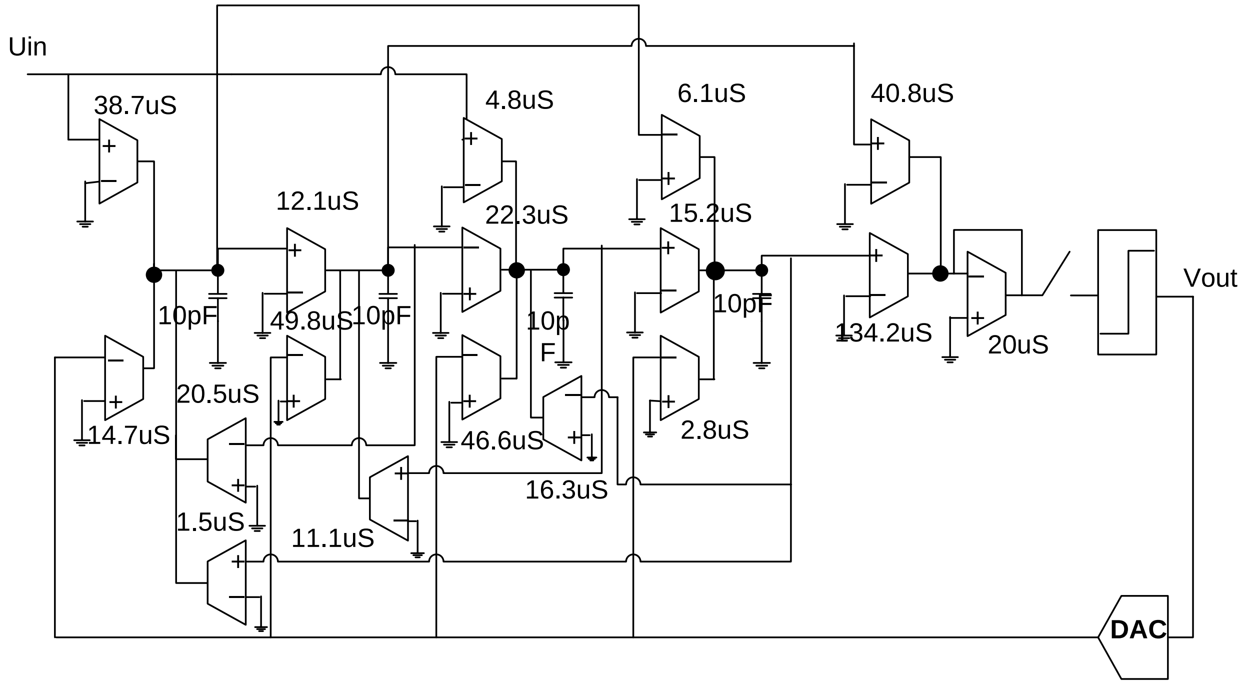
Condition	DR	SNR peak
ideal	95.84 dB	93.85 dB
non-ideal	90.54 dB	91.31 dB

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

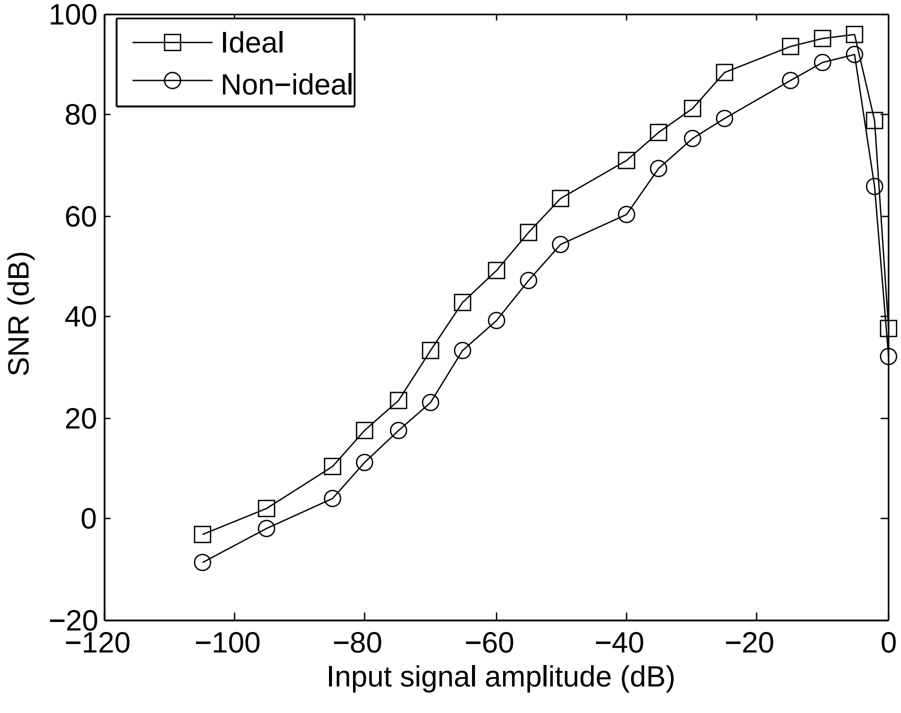
Variation	Generated	CIFF
$\pm 2\%$	1000	988
$\pm 5\%$	890	723
$\pm 10\%$	483	305

Gm values of the CIFF topology 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 ±2%, ±5% and ±10% variation is reported in Table 4.7. We find that the yield is better for the generated topology in comparison to the CIFF topology.

The distribution of the SNR deviation from the nominal value for ±5% coefficient variation is shown in Fig. 4.15. From the experimental results, we conclude that even for the 4^{th} order modulator, the generated topology is better than the standard topology in terms of the sensitivity performance. A comparison between the generated topology and the CIFF topology w.r.t. the relative power cost and hardware complexity is reported in Table 4.8. From it, we observe that the relative power cost is significantly less for the generated topology in comparison to the CIFF topology, although the hardware complexity is greater.



(a) The generated topology for 4^{th} order modulator.



(b) *SNR* plot of the generated 4^{th} order topology.

Figure 4.13: The generated 4^{th} order modulator topology and the corresponding *SNR* plot.

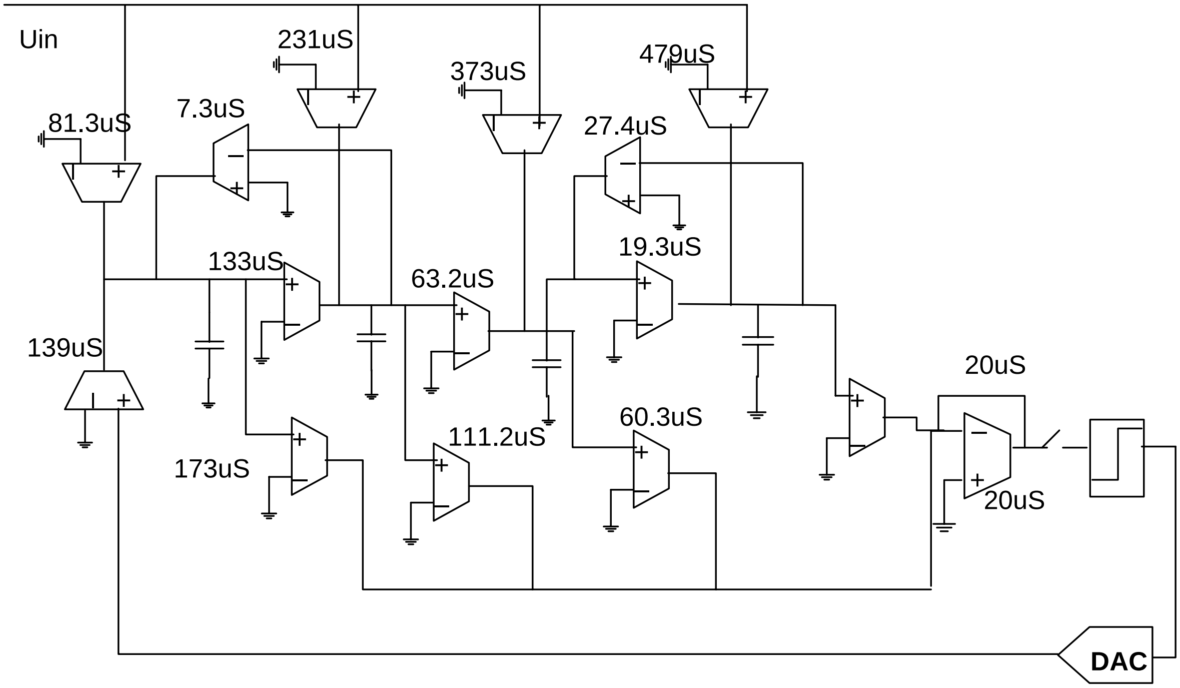


Figure 4.14: The CIFF 4^{th} order topology.

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

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

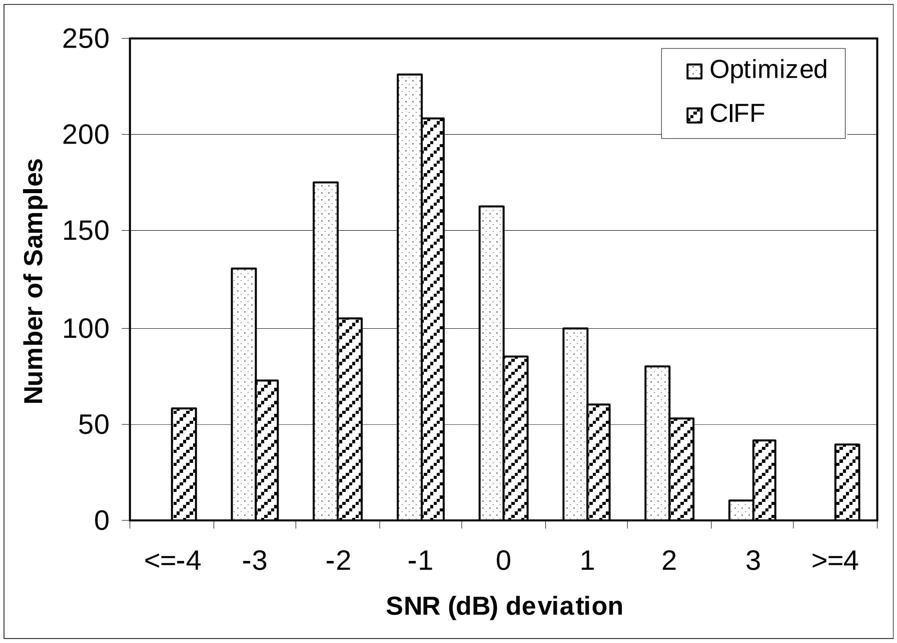


Figure 4.15: Bar diagram of peak *SNR* deviation for 5% coefficient variation for the generated (optimized) and the CIFF 4^{th} order modulator topology.

4.4 Conclusion

In this chapter a methodology for the top-down generation of an optimal functional and component-level topology for linear analog systems has been presented. The topology generation procedure starts from a transfer function model of the system. The transfer function is converted to a state space model, which acts as the basis for topology generation. The topology exploration process is implemented at the state space matrix level. Similarity transformation matrix is used as a topology transformation operator which generates a new state space model from an old one. The newly generated models have same behavioral properties, but they differ in performances. An optimal state space model is selected based upon the model performances employing a simulated annealing-based optimization procedure. The optimized state space model is realized by functional component blocks to generate an optimal functional topology, which is subsequently realized by appropriate circuit-level implementation style specific analog component blocks to generate an optimal component-level topology of the system. In order to ensure that the generated topology behaves satisfactorily under circuit-level non-ideal conditions, a behavioral simulation-based checking process has been included within the topology generation procedure. The state of the art behavioral models have been included within the behavioral simulation process. It must be mentioned that in order to give guarantee of the correct-at-the-first-time working of the obtained topology at the circuit-level, SPICE simulation needs to be included within the optimization procedure. However, this would increase the time complexity of the procedure to a large extent.

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. In addition, it satisfies the feasible Gm constraints and non-overload conditions. Detailed experimentation has been carried out for a 3^{rd} order and a 4^{th} order modulator topology. It is concluded from the experimental results that the generated topologies are better in performances compared to that of the commonly used topologies and satisfy the desired specifications under circuit-level non-ideal conditions.

The advantage of the methodology is that the designer is able to specify the design goal and the 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 ensured to work satisfactorily even under circuit-level non-ideal conditions.

Considering general analog systems, the methodology is limited to linear analog systems, i.e., the systems which can be modeled by state space equations. Many analog circuits like amplifiers, filters etc., are linear and hence for them optimized topology can be generated using the current methodology. In addition, considering the $\Sigma\Delta$ modulator application, the current procedure can be used to develop a fully automated customized topology generation tool.

Chapter 5

High-Level Specification Translation

In Chapters 3 and 4, we have discussed the methodologies for generation/selection of an optimal component-level topology of a system. In this chapter, we present a methodology for determining the specifications of the individual component blocks of the generated/selected topology. This task is referred to as specification translation and is performed through a design space exploration procedure. A meet-in-the-middle approach is followed for constructing the feasible design space. Least squares support vector machine (LS-SVM)-based classification principle is used to accurately identify the actual geometry of the feasible design space. Genetic algorithm (GA) is used in the exploration procedure.

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 is described in Section 4. A comparison between the present methodology and the existing methodologies have been described in Section 5. The experimental results are provided 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 topology of the system is first decomposed into several component blocks, defined by their behavioral models. The specifications of the system under design are then mapped into individual specifications for each of the component blocks within the system topology, so that the complete system meets its desired specifications, while optimizing the design towards some design objectives (e.g., minimal power consumption) [2, 86]. This is referred to as the task of high-level specification translation. A hierarchical specification translation process is generally implemented through a design space exploration (DSE) procedure. In order to ensure that infeasible specifications are not generated for the component blocks through the DSE procedure, recent approaches for specification translation process perform a preprocessing step of feasible design space identification prior to the DSE [59]. The feasible design space defines a region within a design space in which a component block is realizable at the circuit-level of abstraction [30]. [1]

The specification translation problem is mathematically defined by the following transformation function [31]

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

where Φ_j is the j^{th} desired 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:

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

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 $\mathcal{D}_c$. The intersection of these two spaces define the feasible design space. Φ_{jt} is the target value for the j^{th} system specification

---

[1] In Chapter 3 of the dissertation, the GA-based topology sizing procedure performed the task of specification translation. However, in that case the feasibility of the specification parameters of the component blocks were not guaranteed. In this chapter, this issue will be taken into consideration in depth.

and Φ_{js} is the simulated/estimated 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 sub-section 5.2.1 and the circuit realizable space is described in sub-section 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 the 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 [87]. In interval arithmetic, real numbers are replaced by intervals which are combinations of a lower bound and an upper bound on the allowable value range of a variable. In order to perform computations in 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 the whole set of solutions, if more than one) is enclosed within an interval. Whenever there is more than one variable in the problem, the solution is enclosed within a multidimensional interval rectangle. The commonly used methods for solving equations/inequalities using interval analysis technique are Krawczyk method, Hansen and Sengupta method [87]. In the present work, these have been implemented using Matlab Intlab interval analysis toolbox [88]. In simpler cases, as in (5.5)-(5.8), these 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 it 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 this dissertation. Each component block is implemented at the circuit-level of abstraction and is simulated through SPICE. The performance 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 re-

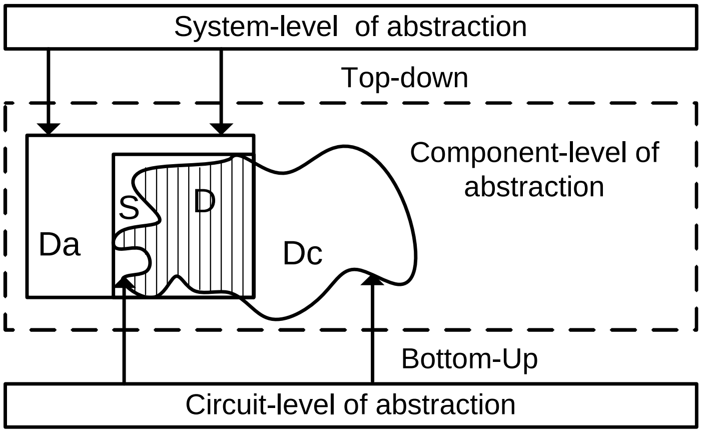

System-level of abstraction
Top-down
Component-level of abstraction
S
D
Da
Dc
Bottom-Up
Circuit-level of abstraction

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.

sults. 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 with in detail in the subsequent sections.

5.3 Feasible Design Space Identification

A two class LS-SVM classification 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 Appendix A.2 and Matlab *lssvmlab* toolbox[64]. The training set consists of samples from both the classes. Large number of infeasible samples are taken compared to feasible samples [34]. 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 hyper parameters, 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 specification of the system or with the change of the desired specification values, the space $\mathcal{D}_a$ may need to be computed afresh and the SVM models may 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:

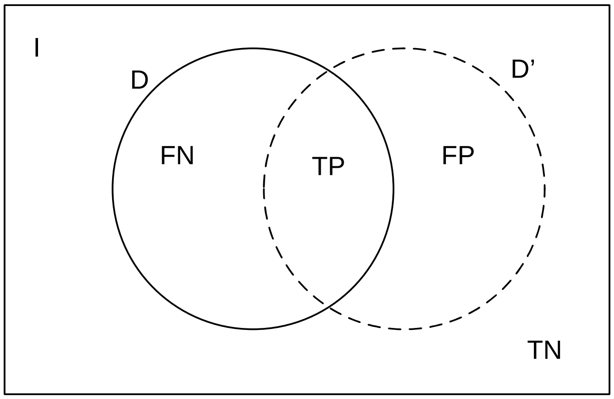


Figure 5.2: Feasible design space and its subspaces.

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 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 DSE procedure is illustrated in Fig. 5.3. The desired functional and performance specifications [2] of the system are taken as inputs from the users. The

---

[2] Those specifications which are to be achieved by the system in order to be functional are the functional specifications, e.g., gain, bandwidth etc. On the other hand, those specifications which are to be optimized, e.g., power consumption, etc. are the performance specifications.

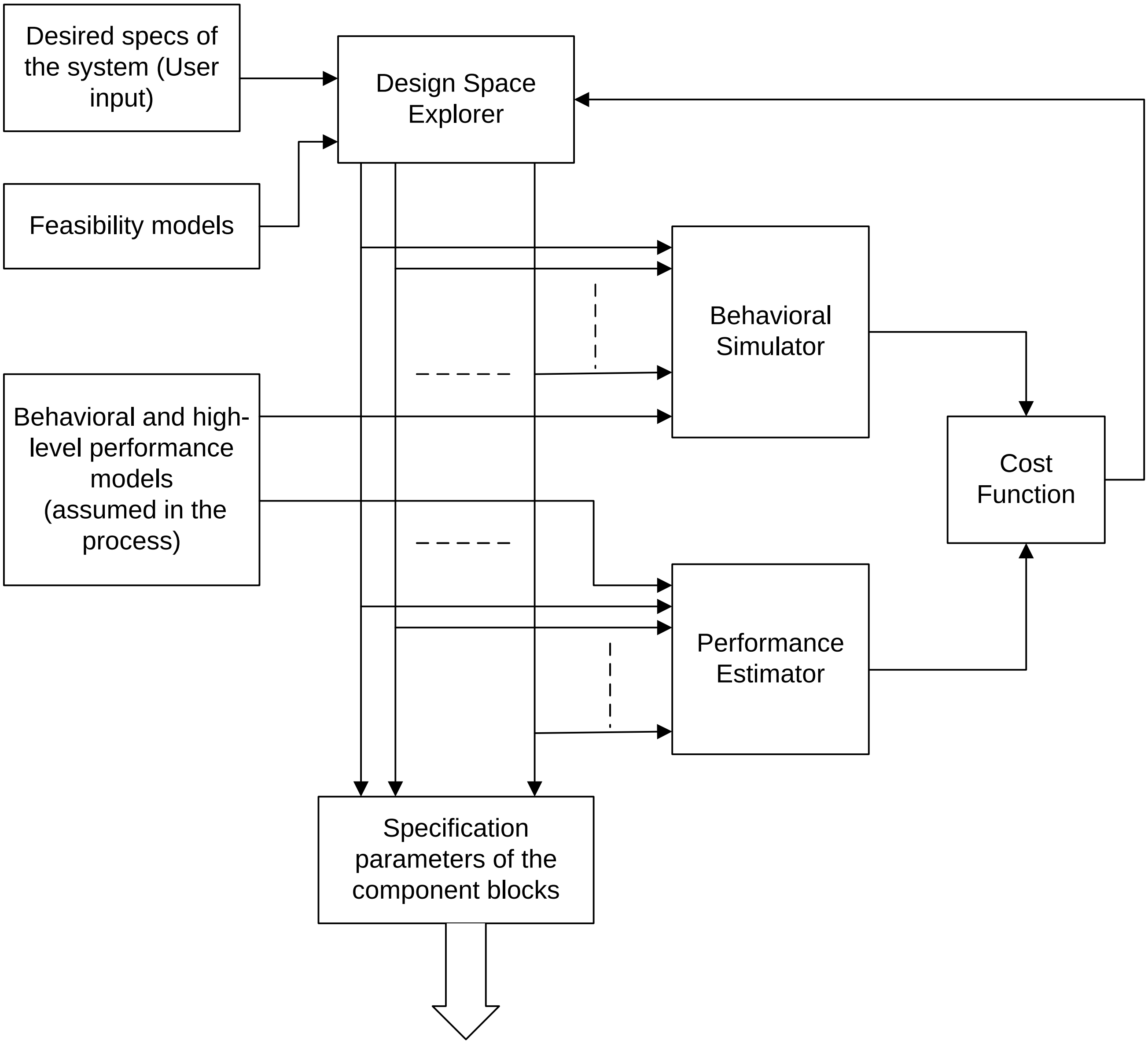

Desired specs of the system (User input)
Design Space Explorer
Feasibility models
Behavioral Simulator
Behavioral and high-level performance models (assumed in the process)
Cost Function
Performance Estimator
Specification parameters of the component blocks

Figure 5.3: DSE mechanism.

behavioral models as functions of the specification parameters of the component blocks, constructed using the analytical techniques discussed in Section 2.2.1.1 of this dissertation are assumed in the DSE process. The high-level performance models as functions of the specification parameters of the component blocks, constructed using the methodology discussed in Chapter 3 of this dissertation are also assumed. The equation-based approach as well as the simulation-based approach discussed in Section 2.1 of this dissertation are used for computing a cost function. 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 estimated values are therefore, used to formulate the cost function. The feasibility models as constructed above are used to define the feasible design space. Genetic algorithm is used as the design space explorer, which determines an optimal set of specifications for the component blocks such that the desired functional specifications of the system are achieved and the performance specifications are optimized.

In the following sub-sections, we discuss the formulation of the cost function and the optimization algorithm that have been used for the exploration purpose.

5.4.1 Cost Function Formulation

The cost function consists of two types of objectives, namely, primary objectives and secondary objectives. The primary objectives are given more importance than the secondary objectives. These are discussed below.

5.4.1.1 Primary Objectives

The primary objectives are: (i) to minimize the relative error between the desired value of the 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 a better tolerance to circuit parameter variations than the peripheral points. The tolerance space for those points have a higher probability to be completely within $\mathcal{D}$ [86]. To achieve this, a secondary objective function is added within the cost function. In this formulation, we assume a hyperellipsoid shape for $\mathcal{D}$ [86] for the secondary objective.

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

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

where ω_{1j} and ω_2 are the weights associated with the primary and secondary objectives. $\bar{X}_C$ 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 Appendix B of this dissertation 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. The fitness of the 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 if the GA iterates for a very large number of loops.

5.5 Comparison with Existing Methodologies

The present methodology is closely related to the methodology described in [59, 31], [19] and [34, 37]. In [59, 31], 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 under-estimation 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 [19] the feasible design space has been identified through box constraints. The circuit realizability of the feasible specification parameters has not been considered. In [34, 37], 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 the increased overhead for construction and accurate identification of the feasible design space. This overhead is much greater than that in [59, 31], and comparable to that in [34, 37]. The present methodology is thus suitable for systems with less number of specification parameters, which may have tight non-linear coupling.

5.6 Experimental Results

To demonstrate the entire methodology, we choose two complete systems - the read-out electronics for a MEMS capacitive accelerometer sensor and the $\Sigma\Delta$ modulator system as case studies.

5.6.1 Experiment 1

The basic block diagram of the topology along with the desired 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 (c.f. Fig. 3.6) and capacitors. The chosen design parameters are gain (A), input linearity (Lin), bandwidth (BW) and output swing (OS) of all the synthesizable blocks. The desired functional specification is to achieve an output voltage sensitivity $\geq 105mV/g$ and the performance – input referred thermal noise is to be minimized.

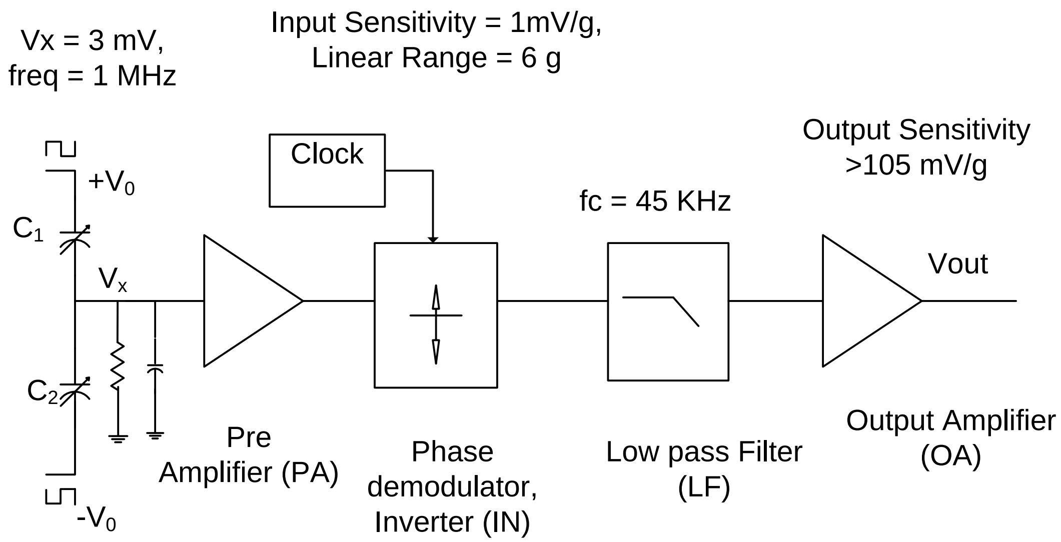


Figure 5.4: Considered system for experimentation.

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

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

A_{IN} and A_{LF} lie in the interval $[0.9, 1.1]$ and $A_{PA} = 2.5 \times A_{OA}$ Through interval analysis method, the intervals of the gain parameter of the individual component blocks are determined. The maximum input signal amplitude is $V = \text{input sensitivity} \times \text{linear range} = 6mV$. 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 $2 \times \text{input signal frequency} = 2MHz$. The upper bound for each of them is taken some higher value. The bandwidth of the LF block is same as the cut-off frequency (f_c), 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 5.1: 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)	$[2, 10]$	$[2, 40]$	$[0.0447, 0.0453]$	$[2, 20]$

Table 5.2: Circuit Realizable Constraints: PA block

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

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 hyper parameter 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 of CPU time for a 512MB RAM, 3.00 GHz PIV PC. The entire data generation process took nearly $4\frac{1}{2}$ hours of CPU time. The total training time for constructing the SVM models of all the blocks is about 44 minutes.

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. 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. We observe from the experimental results of Table 5.4, that the optimization process with secondary objective leaves sufficient margin for most of the parameters. The time complexities are shown in the first row, last column 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

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

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

output curve is shown in Fig. 5.5. The satisfaction of SPICE results with the desired functional specification of the system validates the overall procedure.

Table 5.6: End results: Experiment 1 and 2

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 %

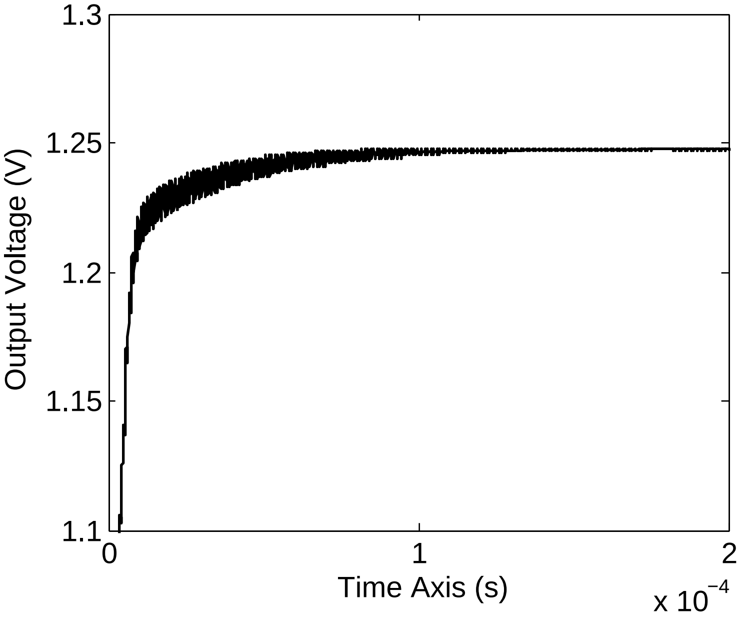


Figure 5.5: SPICE output amplitude with 3 mV input

5.6.2 Experiment 2

In this experiment, the previous system is considered. The desired specification is to achieve an output sensitivity $\geq 40mV/g$ with minimum noise. The application bounded space is computed afresh using the same technique as discussed 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 SVM models of the PA and the OA blocks are reconstructed and that for the IN and the LF blocks are reused. The performances of the constructed SVM models are reported in Table 5.8. The results show that the models are constructed quite accurately. The pre- processing time in this experiment is about 20 minutes. The specification parameters determined through the DSE procedure are tabulated in Table 5.9. The

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
OA	484	5	520	0.945	0.994	0.992

Table 5.9: Translated Specifications: Experiment 2

			Nominal Design	
Blocks	Parameters	Feasible Range	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

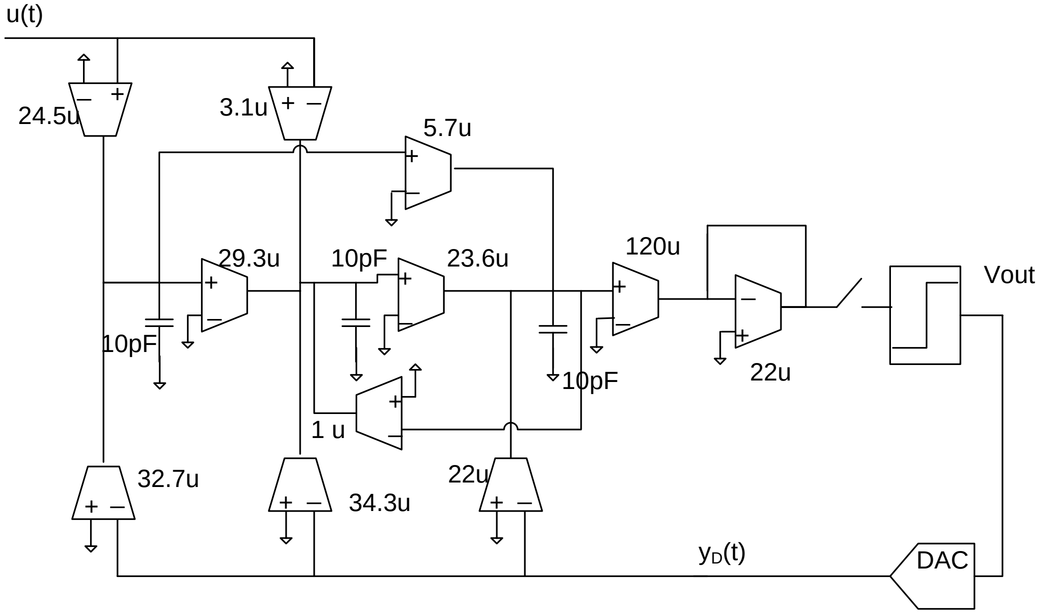


Figure 5.6: 3^{rd} order OTA-C based ΣΔ modulator topology

GA parameters used in the procedure as well as the time complexity of the DSE procedure are reported in the second row of Table 5.5. The end results are reported in the last two rows of Table 5.6. The experimental results demonstrate the fact that with a change in the 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 ΣΔ modulator system, discussed in Chapter 4 of this dissertation. A 3^{rd} order, single bit continuous time (CT) OTA-C ΣΔ modulator along with the Gm values is shown in Fig. 5.6. The target functional specification is dynamic range $(DR) \geq 80dB$. The design constraints are based on [84].

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 using the circuit shown in Fig. 3.6 and the comparator by the circuit shown in 5.7. The application bounded constraints as well as the sizing constraints for the OTAs and the comparator are tabulated in Table 5.10. The determination of the

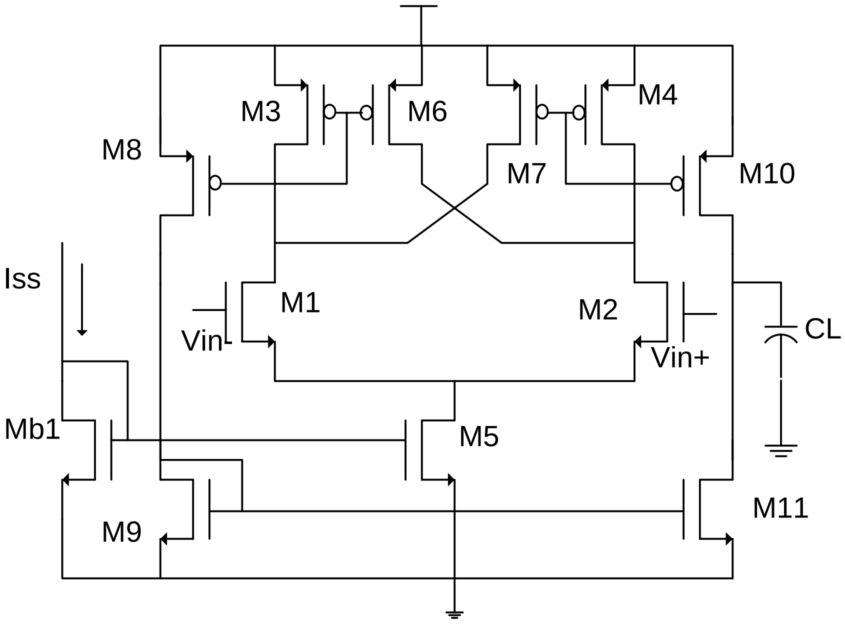


Figure 5.7: Comparator circuit

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 μm]
	$W_3 = W_4 = W_6 = W_7$	[280 nm, 50 μm]
	$W_8 = W_9$	[280 nm, 50 μm]
	I_b	[1 μA, 40 μA]
Circuit params Comparator	$W_{b1} = W_5$	3 μm
	$W_3 = W_4 = W_8 = W_{10}$	[1 μm, 10 μm]
	$W_9 = W_{11}$	[280 nm, 50 μm]
	$W_6 = W_7$	[1 μm, 10 μm]
	$W_6/W_3 = W_7/W_4$	$\geq$ 1
	I_b	50 μA

Table 5.11: Translated Specifications: 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

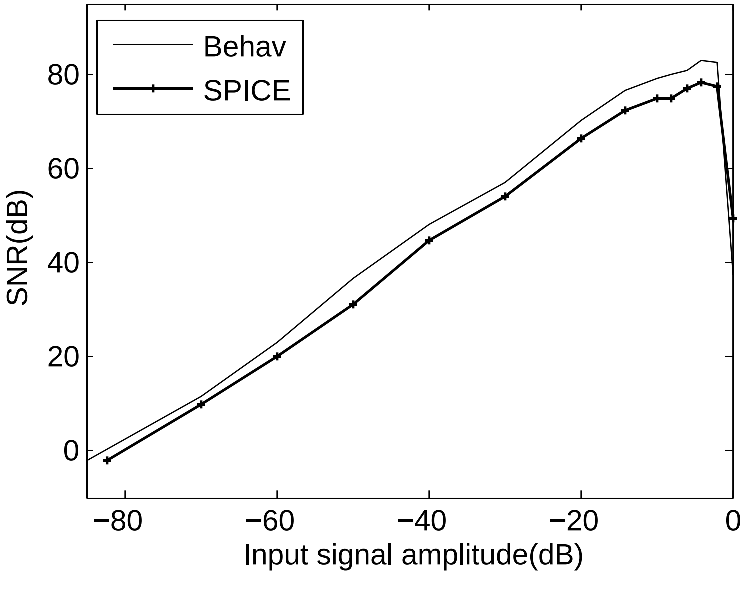


Figure 5.8: Behavioral and SPICE simulated SNR plot of the modulator.

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

swing constraint for the OTAs is 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]. The upper bounds of the application bounded constraints are determined through experience. Efficient SVM models are constructed as discussed 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 to be same as the earlier ones. 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. 5.8. 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

In this chapter, a DSE procedure has been presented for determining specifications of the individual component blocks of an analog system such that the given functional specifications 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 the intersection of an application bounded space and a circuit realizable space. The former is constructed through a top-down procedure using interval analysis techniques and the latter, via a bottom-up procedure through actual circuit simulation. The reduced design space speeds up the exploration procedure. LS-SVM-based classification principle is used to accurately identify the actual geometry of the feasible design space. GA is used for exploring the design space.

The cost function is computed through behavioral simulation of the entire system and by evaluating the high-level performance models of the component blocks. The final solution point is kept away from the feasible design space boundary, in order to increase the tolerance of the component specifications. 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 computational complexity of the entire procedure is determined primarily by the construction of circuit-realizable specification space and identification of the geometry of the feasible design space. The CPU time scales up with the number of performance and design variables. It has been experimentally demonstrated that once constructed, the feasibility models of many of the component blocks can be reused.

The benefit of the methodology is that it is able 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 this 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 for linear analog systems and (iii) high-level specification translation.

This thesis first presents a methodology for generation of high-level performance models for analog component blocks using non-parametric regression technique. The transistor sizes of the circuit-level implementations of the component blocks along with a set of geometry constraints applied over them define the sample space. Performance data are generated by simulating each sampled circuit configuration through SPICE. Least squares support vector machine (LS-SVM) is used as a regression function. The generalization ability of the constructed models has been estimated through a hold-out method and a 5-fold cross validation method. Optimal values of the model hyper parameters are determined through a grid search-based technique and a GA-based technique. The entire methodology has been implemented under Matlab environment. The SVM models have been trained using Matlab toolbox. The methodology has been demonstrated with a set of experiments. It is found from the experimental results that the training time is considerably less for the GA-based training technique compared to the grid search-based training technique, with almost the same generalization ability. It has also been found that when the cross validation method is used for estimating the generalization error during the hyper parameter determination process, the constructed models have a better generalization ability in comparison to that when the hold-out method is used for estimation. The accuracy of the constructed models has been tested by comparing the model-predicted results with actual SPICE simulation results. The two sets of results match closely. The constructed performance models have been used to implement a GA-based topology sizing process. The process has been demonstrated by considering the interface electronics for a MEMS capacitive accelerometer sensor as an example. The topology parameters of this system have been determined such that the desired specifications are satisfied with optimized performances.

This thesis then presents a methodology for top-down generation of an optimal functional and component-level topology for linear analog systems. The transfer function model and the desired specifications of the system are taken as inputs. The transfer functions are converted to state space model which acts as the basis for topology generation. The topology exploration process is modeled as a state space matrix exploration process. Similarity transformation matrix is used as a topology transformation operator. New state space models are generated through this transformation. The newly generated topologies have the same behavioral properties, but different performance properties. An optimal state space model is selected using an iterative optimization procedure employing simulated annealing algorithm. The optimal topology generation process is performed in two steps. In the first step, an optimal functional topology of the system is generated using analog computation techniques and in the second step, an optimal component-level topology of the system is generated from the optimized functional topology of the system. In order to ensure that the generated topology works satisfactorily even under the presence of circuit-level non-idealities, a behavioral simulation-based checking process using non-ideal behavioral models of the component blocks has been included within the topology generation procedure. As a case study, the thesis presents a methodology for generation of an optimal topology for continuous-time $\Sigma\Delta$ modulators. A 3^{rd} order and a 4^{th} order modulator have been chosen as examples for experimentation. It is concluded from the experimental results that the generated topologies are better in performances when compared to that of the commonly used topologies and satisfy the desired specifications even under circuit-level non-ideal conditions.

Finally, the thesis presents a methodology for high-level specification translation. A meet-in-the-middle approach is followed for the construction of a 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 interval analysis techniques and the latter via a bottom-up procedure through actual circuit simulations. Least squares support vector machine (LS-SVM)-based classification 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 cost function is computed through behavioral simulation of the entire system and by evaluating the high-level performance models of the component blocks. The final solution point is kept away from the feasible design space boundary, in order to increase the tolerance of the 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. 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 desired specifications of the system, validating the overall procedure.

6.1.1 Contributions

To summarize, the main contributions of the present thesis are as follows:

1. A methodology is developed for the generation of good high-level performance estimation models for analog component blocks using least squares support vector machine (LS-SVM). The constructed models are accurate with respect to real circuit-level simulation results, fast to evaluate and have a good generalization ability. In addition, the model construction time is low and the construction process does not require any detailed circuit design knowledge.

2. A methodology is developed for the generation of an optimal functional and

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 the 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 construction and accurate identification of a feasible design space and also for an exploration technique 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.

The present methodologies make the high-level design process fast and accurate. In addition, these can even be followed by novice users.

6.1.2 Tools developed

The above methodologies have been implemented under Matlab environment. The following software tools have been developed. These are:

1. The *performance model generation* tool takes as input the dominant specification parameters of a component block and the component block topology. The tool sets up a number of experiments using the design of experiments (DOE) engine. SPICE simulations are performed using a standard simulation tool, e.g., Spectre, HSpice etc. The training data are then collected and fed to lssvmlab toolbox of Matlab for training and construction of the model. Currently the data generation task is performed as a discontinuous step. The trained models are then used to implement a GA-based topology sizing process.

2. The *topology generation* tool takes as input the transfer function model of the modulator, a set of design constraints and the desired DR/SNR specifications. The control system toolbox of Matlab is used to implement the topology transformation process and estimate the modulator performances. The behavioral models of the component blocks have been developed using Simulink. The Simulink simulator is used as the behavioral simulator. The tool gives as

output a state space model corresponding to the generated topology and the required transconductance values.

3. The *high-level specification translation* tool takes as inputs, the desired specifications of the system and parameterized behavioral/performance models of the component blocks. The application bounded space is calculated through Intlab toolbox of Matlab. The user may however, define some of the feasible ranges. The circuit realizable space is constructed using the performance model generation tool and lssvmlab toolbox. The DSE procedure is implemented through GA, Simulink-based behavioral simulation and evaluation of SVM models. The tool gives as output the values of the specification parameters of the component blocks.

6.2 Directions for Further Research

There are a number of areas of further research work related to the present work. These may include the following:

1. A static sampling method has been used for data generation in Chapter 3 of the dissertation. This may be replaced by a dynamic sampling method. In dynamic sampling method, the sampling process is controlled by a so-called ‘learning machine’, which determines the new feature vectors to be sampled based on the 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 the same accuracy. This is very useful when the data generation process is expensive.

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 very high. Efficient iterative algorithms, such as Krylov subspace and Conjugate Gradient (CG) method needs to be used in such cases. Such techniques reduce the training time significantly. The dominant specification parameters considered as inputs in the constructed high-level performance models have been identified in the present work using

the designer's knowledge. These can be identified using data mining techniques. This will make the process more systematic.

3. The methodology for generating high-level performance models can be applied for generating circuit-level performance models with little modifications. In future, circut-level performance models will be generated and the methodology shall be compared with recent related works such as the CAFFEINE methodology [13].

4. The topology exploration process using state space models used in Chapter 4 of the dissertation do not consider the circuit-level non-idealities of the component blocks. The effects of these have been considered in the present work through a behavioral simulation process. The circuit-level non-idealities may be incorporated within the topology 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.

5. 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 too.

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

7. The specification translation methodology as discussed in Chapter 5 of the dissertation determines only nominal values. A more practical output would be a range of values for each specification parameter so that within these ranges, the desired specifications of the system are satisfied within an 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

Least Squares Support Vector Machine

Support Vector Machines (SVM) were first proposed in the year 1995 to solve machine learning problems [89]. Traditional neural network approaches have suffered difficulties with generalization, producing models that can overfit the data. These are consequences of the optimization 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 [90], 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. 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 [63]. In the literature, the terminologies for SVMs are 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 has been referred to both classification and regression methods, and the terms Support Vector Classification (SVC) and Support Vector Regression (SVR) have been used for support vector machine based classification and regression respectively. A modified version of SVM techniques, referred to as the least squares SVM (LS-SVM) had been proposed by Suykens et al. [61]. LS-SVM technique simplifies the traditional SVM technique to some extent. In this dissertation, the LS-SVM technique has been used. 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

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

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 non-linear which makes it possible to approximate non-linear 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 the smallest summed quadratic error of the training samples are determined. Since this can easily lead to overfitting, ridge regression (a form of regression) technique is used to smoothen the approximation. The minimization of the error together with the regression is given as

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

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. The second term takes into account the regression error for all the samples.

The optimization problem (A.3) is considered to be a constrained optimization problem and a Lagrange function is used 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

$$\mathcal{L}(w, b, e, \alpha) = \mathcal{J}(w, e) - \sum_{k=1}^{n} \alpha_k \left(w^T \phi(x_k) + b + e_k - y_k \right) \tag{A.5}$$

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

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w} &= 0 \rightarrow w = \sum_{k=1}^{n} \alpha_k \phi(x_k) && \text{(A.6)}\\
\frac{\partial \mathcal{L}}{\partial b} &= 0 \rightarrow w = \sum_{k=1}^{n} \alpha_k = 0 && \text{(A.7)}\\
\frac{\partial \mathcal{L}}{\partial e_k} &= 0 \rightarrow \alpha_k = \gamma e_k && \text{(A.8)}\\
\frac{\partial \mathcal{L}}{\partial \alpha_k} &= 0 \rightarrow w^T \phi(x_k) + b + e_k - y_k = 0 && \text{(A.9)}
\end{aligned}$$

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

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

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

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

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

The two hyper parameters, 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. In this dissertation, these have been determined through a grid-search based technique and a GA-based technique.

The implementation of the entire LS-SVR technique is available in a Matlab toolbox *lssvmlab* [64] developed by the authors of [61]. This has been extensively utilized in this work.

A.2 Least-Squares Support Vector Classification

The classification problem is restricted to the consideration of the two-class problem without any loss of generality. In this problem, the goal is to separate two classes of data 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

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

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

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

but, on the other hand, is never evaluated in this form. Vapnik's classifier formulation was modified in [61] into 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

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

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

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

where $K(x,x_k)$ is the kernel function. As earlier, the hyper parameters are to be tuned by the model developers.

Appendix B

Global Optimization Techniques

The computer-aided analog design problems are generally formulated as computation of global optimal solutions of an optimization problem [3]. The classical non-linear programming techniques often 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 have been considered. Metaheuristic techniques have been used to solve the optimization problems. These techniques contain all heuristics methods that show evidence of achieving good quality solutions for the problems of interest within an acceptable time. However, these methods do not provide any guarantee of obtaining the global solutions. Metaheuristics are classified into two broad categories; 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 starts in the next iteration. On the other hand, the population-based methods invoke a set of many solutions at the end of each iteration.

In the following sections, we discuss briefly the principles of genetic algorithm as an example of the population-based method, and simulated annealing as an example of the point-to-point method.

B.1 Genetic Algorithm

A Genetic Algorithm (GA) is a search based optimization method that draws inspiration from the concept of 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). The GA starts with an initial population whose elements are called *chromosomes*. A chromosome consists of a fixed number of variables, which are called *genes*. In order to evaluate and rank the 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, the chromosomes with higher fitness function values have a greater chance to be chosen than those with lower fitness function values. The crossover operator defines how the selected chromosomes (parents) are recombined to create new structures (offsprings) 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. The GA operators; selection, crossover and mutation have been extensively studied. Several implementation techniques of these operators have been proposed to fit a wide variety of problems. More details about the GA elements are discussed below before stating a standard GA procedure.

1. *Fitness Function:* A fitness function F is a designed function that measures the goodness of a solution. It is designed in such 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 defines the forms 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 limitations, the 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 selected 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 this dissertation, the proportionate selection operator has been used. This is the most commonly used selection method and is usually implemented with a roulette-wheel simulation method. 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:* A 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 possibly collect all good features that exist in his parents. Crossover operator is applied with probability p_c. An uniform crossover technique has been used as the crossover operator in this 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 possibly 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 do not degrade with generations, the elite-preserving operator is often used in GAs. Typically, the best α% 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 this background on GA operators, a simple GA procedure utilizing these operators is presented below, based upon [91].

1. Select an appropriate coding scheme to represent the design parameters, a selection operator, a crossover operator and a mutation operator. Select 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 operation on the population.
5. Perform crossover operation on random pairs of chromosomes.
6. Perform mutation operation 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 the three operators discussed earlier to a population of points.

B.2 Simulated Annealing

A simulated annealing (SA) procedure simulates an 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 to be 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 called cooling schedule. The main control parameter in the cooling schedule is the temperature parameter T. This makes the probability of accepting a new move to be close to 1 in the earlier stage of the search and to 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, based upon [91]:

1. Select 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. Generally, 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 to escape easily 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. MTT*, Vol.50:pp.360–368, January 2002.

[6] A. Nunez and R Vemuri. An Analog Performance Estimator for Improving 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] T. McConaghy and G Gielen. Double-Strength CAFFEINE: Fast Template-Free Symbolic Modeling of Analog Circuits via Implicit Canonical Form Functions and Explicit Introns. In *Proc. of DATE*, pages 1–6, 2006.

[14] 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.

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

[16] T. Kiely and G. Gielen. Performance Modeling of Analog Integrated Circuits using Least-Squares Support Vector Machines. In *Proc. of DATE*, pages 448–453, Feb 2004.

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

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

[19] 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.

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

[21] 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.

[22] 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.

[23] J.R. Amaya, J.M. de la Rosa, F.V. Fernandez, F. Medeiro, R. del Rio, B. Perez-Verdu, and A. Rodriguez-Vazquez. High-Level Synthesis of Switched-Capacitor, Switched-Current and Continuous -Time Sigma Delta Modulator Using SIMULINK-Based Time-Domain Behavioral Models. *IEEE Trans. CAS -I*, Vol.52:pp.1795 –1810, September 2005.

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

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

[26] 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.

[27] 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.

[28] E. Martens and Georges Gielen. Top-Down Heterogeneous Synthesis of Analog and Mixed-Signal Systems. In *Proc. of DATE*, pages 1–6, March 2006.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] F.De. Bernardinis, M.I. Jordan, and A. Sangiovanni Vincentelli. Support Vector Machines for Analog Circuit Performance Representation. In *Proc. of DAC*, pages 964–969, June 2003.

[35] 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.

[36] 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.

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

[38] 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.

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

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

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

[42] 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 2005.

[43] G. Manganaro, S.U. Kwak, S. Cho, and A. Pulinchery. A Behavioral Modeling Approach to the Design of a Low Jitter Clock Source. *IEEE Trans. CAS -II*, Vol.50:pp.804–814, November 2003.

[44] 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.

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

[46] 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.

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

[48] H.Y. Koh, C.H. Sequin, and P.R. Gray. OPASYN:A Compiler for CMOS Operational Amplifiers. *IEEE Trans. CADICS*, Vol.9:pp.113–125, February 1990.

[49] 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.640–669, March 2007.

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

[51] E. Lauwers and G. Gielen. A Power Estimation Model for High-Speed CMOS A/D Converters. In *Proc. of DATE*, pages 401–405, 1999.

[52] 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, March 2005.

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

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

[55] 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.

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

[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] 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.

[59] 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.

[60] Q.J. Zhang, K.C. Gupta, and V.K. Devabhaktuni. Artificial Neural Networks for RF and Microwave Design: From Theory to Practice. *IEEE Trans. MTT*, Vol.51:1339–1350, April 2003.

[61] 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.

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

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

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

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

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

[67] Thomas Kailath. *Linear Systems*. Prentice Hall Inc., Englewood Cliffs, 1980.

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

[69] 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, August 2005.

[70] 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.

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

[72] 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.

[73] 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.

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

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

[76] L. Thiele. On the Sensitivity of Linear State-Space Systems. *IEEE Trans. CAS*, Vol.33:pp.502–510, May 1986.

[77] 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.

[78] 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.

[79] 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.

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

[81] 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.

[82] S. Pengbo, Y. Wei, and A. Doboli. Flexibility-oriented Design Methodology for Reconfigurable Delta Sigma Modulators. In *Proc. of DATE*, pages 1–6, April 2007.

[83] 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 1–6, April 2007.

[84] H. Aboushady, L.de. Lamarre, N. Beilleau, and M.M. Louerat. A 5mW, 100kHz Bandwidth, Current-Mode Continuous-Time Sigma-Delta Modulator with 84dB Dynamic Range. In *Proc. of ESSCIRC*, pages 283– 286, 2002.

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

[86] H. E. Graeb. *Analog Design Centering and Sizing.* Springer, 2007.

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

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

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

[90] 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.

[91] K. Deb. *Optimization For Engineering Design*. Prentice Hall of India, 2003.

Publications out of this work

Journals:

1. 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.* August 2008, Vol 27, Issue 8, pages 1493-1497
2. Soumya Pandit, C.R.Mandal and A.Patra. An Automated High-Level Topology Generation Procedure for Continuous-Time $\Sigma - \Delta$ Modulator. *Integration -the VLSI Journal, Elsevier* (Communicated).
3. Soumya Pandit, C.R.Mandal and A.Patra. Generation of High-Level Performance Models for the Sizing of Analog System-Level Topologies. *Analog Integrated Circuits and Signal Processing, Springer* (Communicated).

Conferences:

1. Soumya Pandit, C.R.Mandal and A.Patra. Systematic Methodology for High-Level Performance Modeling of Analog Systems, In *Proc. of IEEE/ACM VLSI Design 2009*, pages 361-366,
2. 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,
3. 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,
4. 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,

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

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	Univ. of Calcutta	B.Sc (Physics Hons)	1998
2	Univ. of Calcutta	M.Sc (Electronic Science)	2000
3	Univ. of Calcutta	M.Tech (Radio Physics & Electronics)	2002
4	Indian Institute of Technology, Kharagpur	Ph.D.	Continuing