AN OVERVIEW OF ANALOGUE OPTIMISATION USING "AD-OPT"

Eamonn Byrne, Oliver McCarthy: University of Limerick, Limerick.
David Lucas, Brian Donnellan: Analog Devices B.V, Limerick.

Abstract

A system for optimising analogue circuits is presented. The system "AD-OPT" (Analog Devices OPTimiser) uses numerical simulation and interpolation methods to determine an optimal set of circuit element values and device geometries to meet specific analogue performance requirements. This paper presents an overview of AD-OPT and of its three central modules : the ***datasheet generator****, the* ***database creation module*** *and the* ***optimiser****. (The detailed structure of AD-OPT will be the subject of a separate publication). The paper demonstrates that the increasing speed of work stations, and the judicious use of numerical methods, is altering the balance between theoretical and numerical approaches to problem solving.*

1: Introduction.

Advances in VLSI technology require the integration of both digital functional units and complicated analogue systems on to one chip. Many digital parts can be synthesized rapidly and easily using CAD tools developed for semi-custom design structures such as gate arrays, standard cells and macro-cells. Unfortunately, the state of analog circuit synthesis is quite primitive by comparison. Because of the absence of suitable analogue CAD tools, it is normal that analogue parts must still be designed by a specialist.

1.1: AD-OPT Design Goals

The primary role of AD-OPT is that of circuit optimisation, that is, to determine acceptable parameter values (resistors, capacitors, transistor geometries, etc) which allow the circuit to meet various user-supplied operational criteria (open loop gain, bandwidth, and so forth). The following goals were used as a guide to key decisions regarding the AD-OPT development :

- The basic action of AD-OPT should be to take a proven design of fixed topology, and to rapidly modify that design to meet new performance criteria, but without fundamental topological change.

- The optimisation should produce an accurate solution in a reasonable period of time.

- The design space for a topology should be thoroughly explored, permitting a high level of confidence that an optimum, or near optimum, solution will be found.

- The system should readily accept circuit modules of widely differing topologies (opamps, oscillators, A/D converters, etc).

- The introduction of new topologies to AD-OPT should not require a major new effort in regard to programming, or in regard to expert circuit modelling and design. It should use a proven design as a starting point, together with estimates of the reasonable ranges of design variables that will exploit the topology to its limits.

- The optimiser should be user-friendly and capable of operation by an inexperienced design engineer.

1066-1409/94 $3.00 © 1994 IEEE

The current AD-OPT system meets these goals satisfactorily, and its general features will soon be described. But before proceeding further, a short review of existing optimisation systems is in order.

2: Overview of existing systems

OPASYN [Sequ87] [Sequ90], IDAC [Degr90] and OASYS [Harj87] are some of the other analogue synthesis tools that have been developed. During optimisation, each of these systems employs an analytic module that has been derived by an analogue design engineer. Topology elements are sized by solving the analytic equations. Despite considerable sophistication, the drawbacks of the analytic approach are that it is topology dependent, and that it requires extensive use of analogue design expertise. By comparison, AD-OPT seeks to be a topology independent optimiser, avoiding the use of design engineers to write analytic modules, because the manual derivation of analytic equations is a time consuming, error-prone and difficult task.

PRECISE-OPTIM [Doga89], DELIGHT.SPICE [Nye88] and HSPICE-SUXES [Chen88] are tools that combine an optimisation algorithm with a circuit simulator to select an optimum set of values for a circuit topology. The optimisation normally requires numerous simulations and adjustments of the design parameters until the circuit performs according to specifications. The optimisation-simulation approach requires very little expert design knowledge except for the optimising variables, ranges of the variables and an initial guess for an optimum solution. The information entered by the user determines the design space that is explored for optimisation. The design engineer must choose a good initial guess for each design parameter. Without such a good starting point, an optimisation run may converge very slowly or converge to a local minimum whose performance is significantly worse than the circuit's best capabilities. This approach may be costly in CPU time because repeated circuit simulations are performed in the inner loop of the optimisation step.

An efficient synthesis system was developed at Leuven [Isma90], [Giel90]. The user first enters the performance specifications and selects a circuit topology and the technology process. If the topology is not present in the database, an analytic circuit model is automatically generated by calls to the symbolic simulator ISACC (Interactive Symbolic Analysis of Analogue Circuits). Next the OPTIMAN program (OPTIMisation of Analogue circuits) sizes all circuit elements based on the analytic module. The main advantage of the system is that it is not limited to a fixed set of circuit topologies and allows automatic inclusion of new topologies. A circuit can be explored and optimised in all dimensions of the analogue design space. The equation generator cannot generate large signal characteristics. The analytic equations are approximations of the real circuit behaviour, and the resulting design must be carefully verified.

3: AD-OPT methodology.

The starting point for optimisation using AD-OPT is a circuit module of fixed topology. Each circuit module is considered to have a small number of parameters, *N*, which will serve as the design variables, and which constitute an *N*-dimensional design space. A reasonable value of *N*, for many purposes, is five.

The system AD-OPT (Analog Devices -OPTimiser) relies on the observation that, in the context of analogue circuits generally, output parameters change only slowly as a function of design variables. This makes it possible to characterise an analogue circuit module at points that are sparsely distributed over the design space. Such sparse characterisation becomes the design database for the module.

The database is created once, and is the starting point for subsequent optimisation. This involves full analogue simulation, using the ADICE simulator, for selected component values. It provides a coarse grid of points in the design space at which the circuit has been accurately explored. Database creation requires considerable computer time but is a once-only operation. High-speed work stations, sparse characterisation, and once-only database creation combine to make AD-OPT an attractive alternative to other methods.

When the database has been created, it serves as coarse grid of points at which circuit performance is known. One such point is chosen for its closeness to the target specification; then an interpolation algorithm is used to probe the space between the simulated points to obtain intermediate solutions. This process is quite rapid in comparison with database creation. A solution found by interpolation undergoes full simulation before it is finally accepted.

The design methodology is shown in Figure 1. The user begins by selecting a topology for optimisation. If the design space has not already been created, the database creation module is activated. During this lengthy simulation process, a variety of user-defined specifications are extracted for each permutation of the user-supplied design variables (R1, w1, etc).

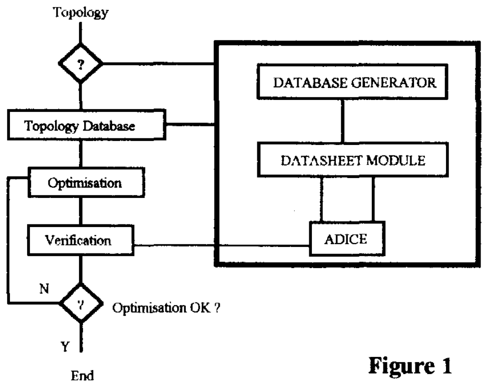

Figure 1

After the database has been created, the topology is optimised for a required performance, by exploring the design space and using interpolation techniques to select values for the variables to meet the requirements. Finally, the results are verified by further simulation.

4: The datasheet generator.

As a requirement of database creation, specifications are extracted and stored for various permutations of the topology variables. The datasheet module uses these specifications for the automatic generation of datasheets. It is not only a central element of the AD-OPT system, but also an independent extraction module that allows a user to quickly obtain formatted specifications for a topology. It extracts accurate specifications using Analog Device's simulator and presents the results in a user controlled format.

The operational amplifier is a common building block. For operational amplifiers, AD-OPT currently permits a total of thirty eight specifications to be extracted, using a five-port op-amp topology. For each specification, there exists a text block structure describing the specification display format, and the files required to extract the specifications.

To extract a new type of specification, three files must be generated by use of a simple programming procedure.

5: Database creation.

The database module creates the design space for a topology and generates the information that it must contain. The database is created once, and is the starting point for subsequent optimisations. The design space must contain the required specifications and design variables to accommodate future optimisations of the topology.

The database contains a datsheet for each permutation of the topology's design variables. The majority of these datasheets in the design space should represent a *stable* system. For an op-amp, the optimisation will produce an *unstable* system if a point in the design space has a phase margin of less than 60°. The user must specify the topology variables, and values of these variables, in order to allow all stable regions in the design space to be explored during optimisation.

Before database creation, a functioning topology, a list of database variables, and values for these variables, are provided by the user. In order to reduce the database creation time the user is encouraged to restrict the dimensions of the database to *five* independent variables and *four* values for each variable. This produces 1024 (4^5) datasheets in the database. The database variables can be any of the following : netlist elements, sources, device geometric parameters, model parameters, temperature etc.

5.1 Expert knowledge.

Some decisions require input from a designer who has knowledge of the topology and its characteristics.

The designer must specify the heuristic equations that are inherent to the topology's correct operation. For example, if one of the database variables is w_1, the width of a device in a differential pair then w_2, the width of the other device, must always have the same value as w_1 if the topology is to function correctly.

The designer also has the option of specifying criteria which determine circuit stability in the design space. For op-amps, thc uscr can sclcct from thc following spccifications: phasc margin, gain margin or pcak gain. The stability specifications are initially extracted for each permutation of the database variables. During the database creation process, the number of unstable points is reported to the user. If a large number of unstable points exist, the user can change the database variable ranges, and the database creation process is restarted. Once the stability specifications are extracted for each permutation of the database variables, then the remaining specifications are extracted for all stable points in the design space.

6: Optimisation.

The optimisation module interpolates between the simulated points in the design space, so as to determine the values of database variables which satisfy user specifications. A *cost function* is used to determine a relative figure-of-merit for the design points in a database. The optimising specifications can be of three types: *precise*, *maximum* or *minimum*. In each situation, the ideal cost is zero, and the cost function is an exponential expression. A user would use the *precise* specification to obtain the required performance with no deviation in *either* direction, otherwise the *maximum* or *minimum* form is used. The use of weighting factors is necessary so that one can adjust the relative importance of each performance factor, in arriving at an overall solution. The *precise* cost function is defined as follows:

$$Cost = \sum_{i=1}^{n} w_i \left\{ \exp\left(1 - \tfrac{AS_i}{RS_i}\right) - 1 \right\}, \qquad \tfrac{AS_i}{RS_i} < 1$$

$$= \sum_{i-1}^{n} w_i \left\{ \exp\left(1 - \tfrac{RS_i}{AS_i}\right) - 1 \right\}, \qquad \tfrac{AS_i}{RS_i} > 1$$

where w_i = *Weighting Priority*, AS_i = *Actual Specification*, and RS_i = *Required Specification*.

The cost functions for *maximum* and *minimum* specifications are similar to the *precise* cost function. The cost is zero for a maximum specification if the actual specification exceeds the requirement. For a minimum specification, the cost is zero if the actual specification is less than the requirement.

During optimisation, the database is scanned, evaluating the cost at all stable points. The points with minimum cost are selected for interpolation. To avoid what are merely local optima, a user specified number of points is selected for interpolation. During the first interpolation, a binary stepping algorithm is employed. The cost is estimated for the interpolated points and the best solution is simulated. After displaying the results, the user decides whether to opt for further binary stepping, or to accept the solution provided.

A cubic spline algorithm [Gera84] is used to interpolate between the specification values. The user also has the option of exploring the end points of the design space by performing an end point extrapolation. The end point extrapolation extends a half point outside the boundary values for the variables. For example, with a database variable whose values are 1k ,2k, 3k, 4k, the region 0.5k to 4.5k is available for exploration. An end point extrapolation is a straight line extension of the slope at the endpoints.

7: Results and Discussion.

The AD-OPT system has been implemented on a SPARC 1 workstation. Currently two opamps and an oscillator circuit have been incorporated into the system. The optimisation results are demonstrated using a general purpose basic two stage BICMOS opamp, as shown in Figure 2.

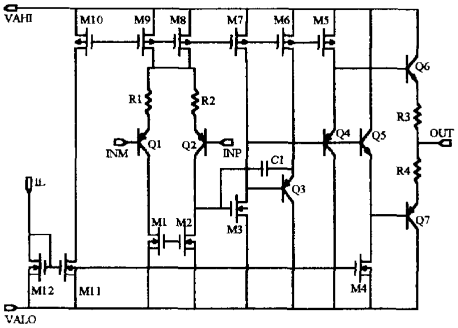

Figure 2. Basic two stage opamp

7.1 Datasheet extraction.

The extraction of numerous specifications, including settling time, input common mode range, etc was made possible by the versatility of the ADICE simulator. The result is automatic datasheet preparation, a significant utility even when no optimisation is required.

7.2 Database creation.

A database was created for *five* database variables, and for *four* values for each variable. The variables were resistor R1 (with R2=R1), bias current I1, M1 device width w1 (with w2 = w1 and w3 = 2·w1), M8 device width w8 (with w9 = w8 and w7 = 2·w8), and capacitance C1.

The database was checked for stability by extracting the specifications phase margin and gain margin. A phase margin less than 60° or a gain margin greater than -10 dB were the criteria used in determining the stability of the opamp. The total number of unstable points in the database was 100. The specifications phase margin, gain margin, open loop gain and unity gain frequency exist for all points in the database. The other specifications were extracted for the remaining 924 stable permutations of the database variables.

7.3 Optimisation.

There were six design objectives for the optimisation : unity gain frequency ≥ 6.5MHz; open-loop gain ≥ 110dB; CMRR @ 1kHz ≥ 105dB; PSRR (+) @ 1kHz ≥ 120 dB; cquivalcnt input noisc @ 10Hz ≤ 300nV/√Hz; slew rate (-) ≥ 10V/μsec. Open loop gain and unity gain frequency were specified at a higher weighting priority than the other four specifications. Two points, having lowest cost values, were chosen for further exploration. The region surrounding the two chosen locations in the design space was explored during each interpolation. The search space during interpolation included a half point step outside the boundary values of the database variable.

Optimisation results

Var1	r1	0.5	1.5K
Var2	i1	3.25	29.75u
Var3	w1	4.0	300u
Var4	w8	2.25	125u
Var5	c1	1.25	11.25p

Specification Name	Required Value	Interpolated Value	Simulated Value	Error %	Units	Cost
Phase margin	> 60	61.5896	59.9976	2.6	deg	N/A
Gain Margin	< -10	-10.2041	-9.7687	4.5	dB	N/A
Unity gain freq	> 6.5	6.25541	6.61591	5.4	MHz	.119
Open loop gain	> 110	108.044	108.121	.071	dB	.054
CMRR @ 1KhZ	> 105	104.713	104.842	.12	dB	.005
PSRR plus @ 1KHz	> 120	124.495	124.694	.16	dB	0
Eq input noise @10Hz	< 300	299.64	299.98	.11	nV/Sqrt(Hz)	0
Slew rate minus	> 10	12.269	12.067	1.7	V/us	0

Total cost	= 0.179
Interpolated grid Step	= 0.125
Number of configurations interpolated	= 4628
Interpolation time	= 8 min 46 sec
Simulation time	= 2 min 35 sec

Table 1

Table 1 shows the optimisation results after three stages of interpolation. The optimum solution obtained is outside the boundary of the database creation values for the variables (the value of R1 is less than 2k), therefore an end point extrapolation was necessary to reach the optimum configuration. After extrapolation the optimum solution was simulated and the extrapolated results were compared with the simulated results. The maximum error turned out to be less than 6 percent. The specifications *gain margin* and *unity gain frequency* display the greatest divergence between the simulated results and extrapolated results. Considering that an end point extrapolation was necessary for optimisation, the predicted results compare very well with those simulated.

8: Conclusions.

The Analog Devices optimiser AD-OPT has been presented. The database for the module is created by a lengthy simulation run which is performed once only, and this database becomes the starting point for subsequent optimisations. The topology is optimised for a particular set of specifications by interpolating between the simulated grid points of the database. Optimisation is performed in minutes, so that many variations on a basic topology can be quickly explored, and the outer boundaries of its capabilities established, all in very short time.

8.1 Advantages of system.

1. The AD-OPT system contains a datasheet module which allows a user to extract a wide range of specifications, in a user controlled format for a topology. The module also lends itself to further development toward the automatic generation of *macro-models* for the parts in question.

2. The database module is an integral part of the AD-OPT system, but it can also be used as a workhorse simulator, allowing users to extract specifications for a wide range of variables: model parameters, temperature, netlist variables etc.

3. The AD-OPT system is *topology independent*. As a further illustration of this, the optimisation of a BI-CMOS oscillator circuit consisting of 46 components was readily accomplished using AD-OPT. The addition of the oscillator circuit to the library required the attention of an analogue design engineer and an ADICE programmer and was achieved in approximately four hours.

4. The optimisation results are of good accuracy, the maximum deviation between the simulated results and interpolated results being less than 6 percent in the example presented.

5. The optimisation system requires very little design knowledge, requiring only the specification of database variables and variable ranges, and net-list heuristic equations.

8.2 Limitations of the system.

1. The database creation time can take up to approximately 40 hours on a SPARC 1 workstation for a sophisticated topology. But given the current rapid development in workstation performance, the optimisation times and database times will continue to decrease.

2. More sophisticated search techniques could be incorporated, yielding modest gains in overall performance.

REFERENCES.

[Chen88] J. Chen, J. Kueng, H. Chen and F. Fernandez, "ADOPT-A CAD System for Analog Circuit Design", proc. IEEE Custom Integrated Circuits Conference, 1988, p. 3.2-1.4.

[Degr89] Marc Degrauwe, et, "Towards an Analog System Design Environment", IEEE Journal of Solid-State Circuits, Vol. 24, No. 3, June 1989, p. 659-669.

[Doga89] Kyriakos Doganis, Kevin Walsh and Prof. Panos Linardes, "Synthesis of Analog ASIC's Using Optimisation in Conjunction with Circuit Simulation Techniques", Proc. Second Annual IEEE ASIC Seminar, September 1989.

[Gera84] Curtis Gerald and Patrick O Wheatley, "Applied Numerical Analysis", Addison-Wesley, 1984.

[Giel90] Georges Gielen, Koen Swings and Willy Sansen, "An Intelligent Design System for Analogue Integrated Circuits",

[Harj87] Ramesh Harjani, Rob Rutenbar and Richard Carley, "A prototype framework for knowledge-based analog circuit synthesis", proc. Design Automation Conference, 1987, p.42-49.

[Isma90] Mohammed Ismail and Jose Franca, "Introduction to Analog VLSI Design Automation", Kluwer Academic Publishers, 1990.

[Nye88] William Nye, David Riley and Andre Tits, "DELIGHT.SPICE: An Optimisation_Based System for the Design of Integrated Circuits", IEEE Transactions on Computer-Aided Design, Vol. 7, No. 4, April 1988, P. 501-519.

[Sequ90] Han Young Koh, Carlo H. Sequin and Paul R. Gray, "OPASYN: A Compiler for CMOS Operational Amplifiers", IEEE Transactions on Computer Aided Design, Vol.9, No.2 February 1990, p.113-125.

[Sequ87] Han Young Koh, Carlo H. Sequin and Paul R. Gray, "Automated synthesis of operational amplifiers based on analog circuit synthesis", proc. IICAD, 1987, p. 502-505.