

*******US Copyright Notice*******

No further reproduction or distribution of this copy is permitted by electronic transmission or any other means.

The user should review the copyright notice on the following scanned image(s) contained in the original work from which this electronic copy was made.

Section 108: United States Copyright Law

The copyright law of the United States [Title 17, United States Code] governs the making of photocopies or other reproductions of copyrighted materials.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the reproduction is not to be used for any purpose other than private study, scholarship, or research. If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use," that use may be liable for copyright infringement.

This institution reserves the right to refuse to accept a copying order if, in its judgement, fulfillment of the order would involve violation of copyright law. No further reproduction and distribution of this copy is permitted by transmission or any other means.

Northeastern University ILL



ILLiad TN: 95789

Borrower: MYG

Lending String: LIN,*NED,RRR,KLG,OSU

Patron: Aggarwal, V

Journal Title: Analog integrated circuits and signal processing.

Volume: 3 **Issue:** 3

Month/Year: May 1993 **Pages:** 217-228

Article Author:

Article Title: Henderson, R 'Analog integrated filter compilation'

Imprint: Boston/U.S.A. ; Kluwer Academic
Publishers

ILL Number: 12749701



Call #: T1 .A5420 v.3 1993

Location:

ARIEL

Charge

Maxcost: \$35.00IFM

Shipping Address:

Interlibrary Borrowing
Massachusetts Institute of Technology
Interlibrary Borrowing
Room 14-0551/ILB
Cambridge, MA 02139-4307

Ship Via: Ariel

Fax:

Ariel: 18.51.0.167

This material may be protected
by U.S. Copyright Law (Title
17, U.S. Code). The user of
this material is responsible
for compliance with the law.

OCT 03 2005

Analog Integrated Filter Compilation

R.K. HENDERSON

Centre Suisse d'Electronique et de Microtechnique, Neuchâtel, Switzerland

LI PING

Department of Electrical and Electronic Engineering, University of Melbourne, Parkville, Victoria 3052, Australia

J.I. SEWELL

Department of Electronics and Electrical Engineering, University of Glasgow, Glasgow, G12 8QQ Scotland

Abstract. A review of the progress in automated design of analog integrated filters is presented. Such tools are ahead of other analog circuit automation in terms of the acceptance by designers and practical applicability. A survey of the present-day commercial and academic systems is made and the range of facilities available is compared. The problems faced in the design of this type of software are typical of the problems of analog design systems in general; lack of openness for introduction of new design knowledge, difficulties of dealing simultaneously with expert and novice users, poor integration in design environments, and user-interface problems. The structure of a typical system is studied and the computer methods used within are discussed with regard to such issues as speed, flexibility, and ease-of-use. Some future directions for analog filter compilers are proposed.

1. Introduction

Synthesis and compilation tools for analog circuits have been slow to develop compared to their digital counterparts. This has been in part due to the greater difficulty in identifying the rules involved in analog design and a certain reluctance on the part of the designer to accept automation of his or her highly knowledge-intensive skill. Tools that do exist fall mainly into the categories of design capture, analysis and verification. Filter design is an exception; a set of clearly defined hierarchical steps backed up by a large body of well-established mathematical theory renders the discipline amenable to automation. The earliest design programs were developed in the 1950s and 1960s for passive *RLC* filters and demonstrated the feasibility of automatic circuit synthesis [1]. The domain witnessed the earliest application of several computer techniques to circuit design problems (notably optimization). When the modern integrated filter technologies such as active-*RC*, switched-capacitor (*SC*), and continuous-time arrived in the 1970s and 1980s they were followed up quickly by computer automation [2]. Moreover, these tools were successful in gaining acceptance by designers. There are two main reasons for this confidence. Unlike other analog blocks, high order filter circuits are a common requirement, demanding a considerable number of trade-offs and

tedious numerical design steps. These increase greatly with order, quickly exceeding the scope of manual design but ideally suited to the capabilities of the computer. Second, the filter technologies (for mainstream applications) are now very well mastered, allowing a sufficient degree of assurance in "what you design is what you get" to permit computer aids to take over.

What are the main aims in the development of CAD tools for filters?

1. *To reduce design time and cost.* Filter design turn-around is reduced from months to a matter of days. Quick estimates of silicon area and power allow designers to make important trade-offs at system level. Filters synthesized by compilers come with a "correct-by-construction" guarantee (meaning that if there are no errors in the CAD software then the network connectivity and component values must be correct!). No errors means no costly redesign.
2. *To provide optimal designs.* Filter attributes can be tailored to specifications reducing wasted area and power. Computer assistance is essential in this computationally expensive task.
3. *To adapt quickly to changes in the technology.* Now that filter design tools are accepted the challenge is to make them more capable to absorb new technological developments, new circuit topologies and

design methods. This need has been made particularly evident by the continual emergence of new technologies, recently switched-current and MOSFET-C circuits [3-5].

Several filter compilers are now offered commercially and many more reside in academia. Filter design aids are also increasingly found as extensions to digital signal processing packages, standard circuit analyzers, and mathematics and systems simulation languages. This article will look at some of the issues faced in the design of this kind of software as a small illustration of the problems facing the current generation of analog CAD tools. One particularly significant issue, in view of the rapid technological developments over the past 30 years, is the extent to which "technology independent" filter design can be achieved. This means the degree to which the shared design methodology of filter in various technologies can be exploited to provide reusable programs (a major goal of present-day computer science). Good design software must not only be flexible enough to adapt easily to the changing possibilities offered by the technology but must also be able to incorporate the increasing base of knowledge of circuit structures. Versatile databases and algorithms are essential to provide this flexibility and some important contributions to this are reviewed. CAD, in general, is encountering increasingly the problem of tool integration; tools which were developed as stand-alone entities are being asked to work together to build larger systems. Filter compilers are no exception and their lack of integration is hindering the development of mixed analog/digital filter systems. Another theme will be the extent to which automation techniques can hide complexity from the user to provide simpler design decisions. There is always a tension between offering an excessive number of options to a designer and hiding too much ("push-button design"). The former risks bewildering the newcomer and the latter risks losing the confidence of the expert. A continuum between these two extremes needs to be offered by well-designed software with a sufficiently uniform view of the design process. These themes will be illustrated by developments and examples drawn from the XFILT filter compiler [6].

2. Background

Integrated filter compilation is the translation of a filter from a high level design description into layout [7]. There is a hierarchy of levels of description, involving more and more detail as we approach layout. Moving

between levels is accomplished by a synthesis step which converts a design from a *behavioral* to a *structural* description. For example, *filter synthesis* commences with the description of the design in terms of a frequency response and terminates with a netlist of parameterized building blocks, e.g., op amps, transconductors, capacitors, resistors, and switches. Several synthesis steps follow before we finally arrive at silicon; for example an analog cell generator will translate from building block to sized device schematic and a layout tool from sized device to layout geometry. A typical organization of such a system is shown in figure 1 and the "state-of-the-art" is summarized in table 1 which compares some of the most reputed systems (this list is by no means exhaustive).

Some pertinent comments follow:

1. There is a very dominant forward path in the compilation process. Most compilers adopt the principle of making fast synthesis modules and placing the designer in control of a "weak" optimization (manual

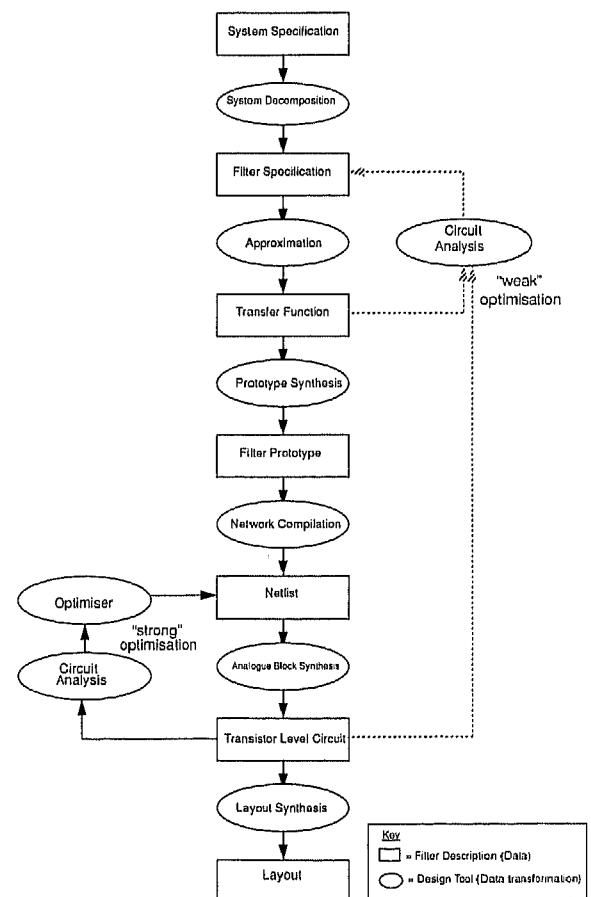


Fig. 1. Design flow in automated filter synthesis.

Table 1. Comparison of integrated filter design software.

Name	Classical approximations	Extended approximation	Allpass Equalisation	Ladder synthesis	Cascade biquad	Layout synthesis	Passive RLC	Switched-capacitor	Continuous-time/Active-RC	Commercially available	Special features/ comments
S/FILSYN	●	●	●	●	●		●	●	●	●	The original passive filter synthesis program. Now extended to various filter implementations and design methods [8].
filterX	●	●	●	●	●	●	●	●			A compendium of programs written by graduate students at the University of Toronto.[9-12]
VITOLD	●		●		●	●		●		●	Fairly complete commercial package. Z-domain approximation and synthesis. Sensitivity and THD optimisation.[13]
AutoFilter	●	●			●	●		●		●	Commercial package from Mentor Graphics [14].
PANDDA	●	●	●	●	●		●	●	●		High order touch point approximations. Wide variety of filter structures including new topologies. Non-Ideal optimisation.[17]
IMSYS	●			●		●		●		●	Exact z-domain ladder synthesis. Leapfrog simulation ANACAD product. Good practical design options.[15].
FIESTA	●	●			●	●			●		First OTA-C filter synthesiser [16].
SCSYN				●	●	●		●			Gate-array SC filter synthesiser. Unified design of biquad and ladder structures. Noise and capacitance optimisation.[18]
AROMA	●		●		●			●			One of the earliest SC filter compilers. Cascade biquad trade-offs. [2]
MASFIL		●	●		●			●			Simulated annealing design algorithm. Anti-alias filter design. [19]
CAST/ALEX						●		●			Dedicated SC filter layout synthesis tool. Amplifier and switch sizing.[20]
SCULPTOR	●	●			●	●		●			Japanese contribution to filter automation. [21]
PSpice	●	●	●		●			●	●	●	Promising PC synthesis tool in popular environment of SPICE analysis tools.
SCDS	●	●	●		●	●		●		●	Bell Northern Research SC synthesis tool plus University of Waterloo analysis software marketed by Cadence [22]

iteration round the tools, viewing circuit simulation results and altering specifications). The alternative "strong" optimization approach consists of a core circuit simulator, a multivariable optimization package, and a graphical user's interface for viewing simulation results. In this case, an ideal design is taken as a starting point and circuit element values are manipulated by the optimizer to improve the circuit response in the presence of nonidealities such as switch resistance or amplifier bandwidth. However the "weak" approach is more efficient since there are generally many fewer filter specifications than circuit element values, but it relies on designer expertise to interpret simulation results and modify specifications accordingly. The "strong" approach trades speed for generality and ease-of-use tending toward the "push-button" end of design automation. Both (and a range of possibilities in between) are necessary for a complete system.

2. Very few existing compilers offer the complete cycle as depicted in figure 1. In particular, the system

decomposition is rarely automated leaving this step to be performed by rule-of-thumb and system designer's expertise. This is a prospective area for future automation. The aim would be to help system designers to set-up realistic specifications and to obtain quickly a feel for the trade-offs involved between blocks and the possibilities offered by the technology.

3. At each of the filter description levels there is a potential to save the state of the design in some data format. There is very little standardization of these formats (except at netlist or layout level) leading to a difficulty in porting information between systems. For example, a standard frequency domain specification format or transfer function description format would allow much more shareable and extendable use of approximation software. At present however, advanced approximation techniques become an inaccessible part of a single system and there is very poor reuse of even standard software. Standard simulator formats for analog and switched-capacitor circuits are emerging by

default as SPICE and SWITCAP. Layout descriptions have already the GDSII and EDIF formats available. In certain cases this is the domain of analog behavioral modeling languages such as the projected analog extension to VHDL.

4. Compilers are normally devoted to only one design strategy and technology. Commercial compilers in particular restrict themselves to the mainstream design flow of cascade biquad designs in switched-capacitor technology derived from classical approximations (Butterworth, Chebyshev, etc.). This limits the designers ability to combine and compare technologies (e.g., active-RC and SC) and the benefits of different topologies (e.g., biquad and ladder). Not enough work has been put into an open framework of description for the design methods and the networks.

5. The importance of having more powerful approximation software is emerging [23]. As the design steps close to the fabrication technology are optimized the "room for improvement" shifts toward the higher levels of design (see also point 2) such as approximation. For example, traditional approximations based on classical functions (Butterworth, etc.) yield functions with *flat* passbands which are suitable for frequency division multiplexing applications. However, filters with *shaped* frequency response can also compensate for signal distortions from other parts of a communications system (typically antialiasing and transmission line losses). Compared to a standard equalizer/filter solution the incorporation of an equalizing capability in the filter results in smaller circuits and better overall performance. Approximation for such designs is increasingly being recognized as a necessary utility in a compiler and several packages offer some facility to optimize a frequency response to an arbitrarily shaped template [17, 19, 22].

Instrumentation and data communications require approximations based on phase and time domain requirements. There is very little software available for such tasks and mostly designs are either "handcrafted" or cast in terms of problem to be solved by an optimization package.

6. The majority of compilers originate from universities. As with all analog CAD software it covers a very narrow and highly specialized group of users. The ratio of difficulty of development of the software to potential market is high, making it commercially unattractive. One solution to this problem is to place the software under the "umbrella" of digital CAD tools (i.e., make use of the schematic capture, layout, database, and user-interface standards). Closer integration would

allow the digital tool to claim "full mixed analog/digital synthesis capability" while protecting the analog tool's interests.

3. Computer Methods for Analog Filter Compilation

The previous section has taken an external view of the existing filter compilers and the trends in their development. The present section will take an internal view and will examine some of the computer techniques being used inside such systems.

3.1. Approximation

In filter approximation a realizable transfer function must be computed to meet specifications of amplitude and delay in time or frequency domains. Obtaining a good approximation is a struggle between conflicting demands of filter selectivity, group delay variation, time domain response, and transfer function order. Approximation software can either help or hinder this process depending on its flexibility, speed and ability to satisfy both advanced and beginner users.

Table 2 shows a survey of the most successful algorithms for filter approximation. From a software perspective, a major remaining issue is simply the degree of accessibility to the user. Entering the specs and viewing the responses of classical filters such as Butterworth or elliptic is a fairly simple matter of entering fixed parameters and viewing the approximated solution. The order of the response can be determined from passband, stopband ripple, and frequency edge specifications (in fact any one of these can be left open, to be determined automatically from the specs on the other three). There is usually very little exploration of responses to be done (often the specification will even define the order and

Table 2. Comparison of different filter approximation algorithms.

User	Speed	Flexibility	Approximation Method	Properties	Ref
NOVICE	FAST	LESS FLEXIBLE	CLASSICAL	Butterworth Chebyshev etc. flat pass and stopband frequency domain amplitude specs.	[24]
			REMEZ	"Pole placer" algorithm family, allows weighted frequency domain passband and stopband specs.	[24]
			ALLPASS GROUP DELAY EQUALISATION	Equitipple group delay specifications Remez-type algorithms applicable Less efficient than joint amplitude/group delay optimisation.	[25]
			LEAST p^{th} OPTIMISATION	Simultaneous specs on amplitude, group delay and time domain response.	[26]
			QUADRATIC PROGRAMMING	As above but slow with convergence problems	
EXPERT	SLOW	MORE FLEXIBLE	SIMULATED ANNEALING	Slowest but avoids local minima problems and is very flexible.	[19]

type of approximation to choose) and menu-driven input is quite satisfactory.

Design of more complex responses is quite another matter, the designer has great freedom to vary the characteristics of the filter in the passband(s) and stop-band(s) by exploring different pole and zero placements. Here, graphical specification aids are important for the designer to visualize the response in the different domains of interest (frequency, time, and group delay). Often a problem is to view superimposed filter response plus a distortion function from another part of the system or from another cascaded filter. Easy manipulation of the plots and feedback into the pole placer algorithm is needed (e.g., the ability to select and place zeros or to interactively edit the shape of the template). Usually, a graphical input needs to go hand in hand with textual specification for detailed manipulation.

3.2. Prototype Synthesis

Prototype synthesis is the process of decomposition of the transfer function into simple terms corresponding to a realizable circuit building block. It is here that the basic topology of the filter begins to be seen. The algorithms are essentially numerical and are divided into the two principal categories of cascade biquad design and passive ladder synthesis (see table 3). The algorithms themselves are now well known. The problem is now largely how to present the bewildering (combinatorial) number of possible decompositions to the designer. Each decomposition of the transfer function will result in a circuit with different noise, dynamic range, area, and sensitivity.

Ladder synthesis is particularly difficult to automate satisfactorily because it requires an understanding of a number of theoretical conditions on the types of decomposition allowable, yet is worth the bother because of the higher quality circuits it generally provides. For the "push-button" user a default ladder structure can normally be proposed using inbuilt knowledge of the allowable synthesis steps. For the expert who wants to improve on the silicon area, or who needs a special structure of prototype the ladder synthesizer can propose a prioritized choice of structures.

Cascade biquad design requires a choice of a pole sequence and corresponding zero sequence ("pole-zero pairing") of which there are a factorial number. It is feasible up to around 14th order filters to investigate

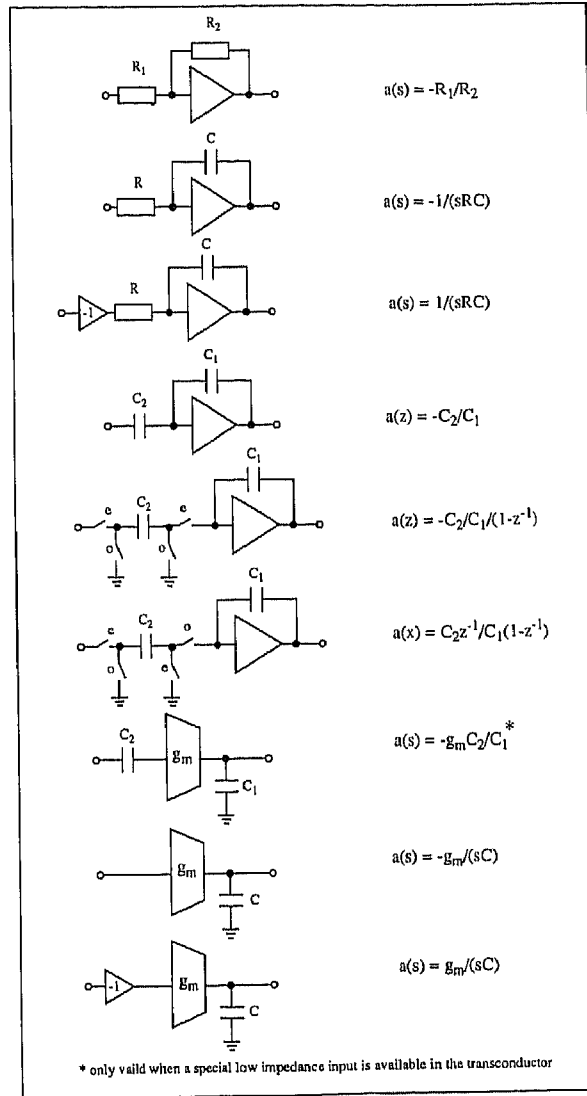


Fig. 2. Filter building-block library in different technologies.

Table 3. Comparison of different filter prototype design methods.

Ease-of-use	Algorithm	Quality of Circuit	Prototype Design Method	Properties	Ref
NOVICE	SIMPLE	LOW	RULE-OF-THUMB POLE-ZERO PAIRING	Cascade biquad circuits. Choose a fixed sequence of poles and zeros according to pre-defined rule.	[27]
			COMBINATORIAL POLE-ZERO PAIRING	Cascade biquad circuits. Test all possible pole-zero pairings to find one giving best circuit performance (below 14th order)	[27]
			ITERATIVE LADDER DESIGN	Design a fixed structure of ladder by matching its transfer function iteratively with desired one.	[28]
			PASSIVE LADDER SYNTHESIS	Synthesize a ladder by decomposing its transfer function using polynomial manipulation. Accuracy problems. Often needs expert but very general.	[29]
EXPERT	COMPLICATED	HIGH			

all possible combinations. However the assessment needs to be based on some performance measure of the filter, e.g., area and sensitivity. A compiler can have a built-in performance measure or can allow a user to supply his own based on access to analysis results. The expert should also be permitted to enter a user-defined pairing while the "push-button" user can be offered default pairings based on reasonable rule-of-thumb choices, e.g. increasing Q factor for lowest noise transmission.

3.3. Network Compilation

Network compilation is the stage at which a filter prototype is decomposed into a netlist of ideal linear network blocks (capacitors, transconductance amplifiers, switches, op amps, etc.). There is very little standardization here, and the algorithms used are often rewritten for each different structure of filter. Yet a certain degree of technology independence can be maintained at this level so that the designs do not need to be recoded when the implementation of the building blocks change. The main task at this stage is the construction of a linearized flow graph of the circuit in terms of the building blocks. *The flow graph structure itself stays essentially constant between implementations. The basic integrator or resonator building blocks do change in implementation (switched-capacitor, switched-current, transconductor-D, active-RC).*

A convenient and portable representation of a flow graph is a matrix system. Several authors make use of such a system but no convention has emerged [30–32]. One possible form is a pseudostate space description

$$X = AX + BY \quad (1)$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a_{11}(p) & a_{12}(p) & a_{13}(p) \\ a_{21}(p) & a_{22}(p) & a_{23}(p) \\ a_{31}(p) & a_{32}(p) & a_{33}(p) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_{11}(p) & b_{12}(p) & b_{13}(p) \\ b_{21}(p) & b_{22}(p) & b_{23}(p) \\ b_{31}(p) & b_{32}(p) & b_{33}(p) \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad (2)$$

where p can be either s or z and where the functions $a_{ij}(p)$ and $b_{ij}(p)$ represent the transfer function of the building block (typically but not necessarily integrator or resonator transfer functions). Figure 2 shows examples of possible block functions plus their realizations. Note that such a scheme has several advantages for computer implementation:

1. The netlist can be generated directly from the matrix form by representing each x variable by an active device (op amp, transconductor, etc.) output and linking the appropriate block as represented by an a_{ij} from output i to input j . The y vector represents an input signal (normally only one), and they should be linked to the corresponding x input via the blocks indicated by the b terms.
2. A library of the standard cells and their block transfer functions can be set up and modified according to changes in the technology or development of new cell topologies, e.g., low offset integrator structures.
3. Being based on a matrix representation the scheme is ideal for computer implementation and storage. A library of matrix manipulation routines can be made available to allow quick coding of new designs.
4. The matrix can be analyzed directly by substituting the numerical values of the block transfer functions and solving by conventional LU decomposition methods. This provides a quick first check before an external circuit simulator is called and can supply estimates of sensitivity and dynamic range. Note that since the matrix rank is normally dependent on the order of the filter rather than the number of components in the eventual circuit that the analysis is very efficient. The matrix is built up from the transfer functions of the building blocks rather than those of the individual components inside them (as it would in a standard MNA scheme). This avoids reanalyzing each occurrence of a building block in the circuit to determine its transfer function.
5. The matrix allows scaling for minimum area and maximum dynamic range to be performed with ease. Simple row and column multiplications are required.

Both cascade biquad and ladder structures in a variety of technologies can be represented by the above scheme. The designer is then faced with problem of choosing an appropriate topology of cascade biquad or ladder. There are many possibilities such as single op amp biquads, E or F-type biquads, low C-spread biquads as well as leapfrog, coupled-biquad, LU, gyrator-based ladder simulations [29–30]. Each structure has its own characteristic properties of noise, dynamic range, sensitivity and area requirement. The choice is strongly dependent on

1. The class of filtering (bandpass, low-pass, all-pass, etc.)
2. The bandwidth or Q -factor of the filter (narrow-band, wide-band)

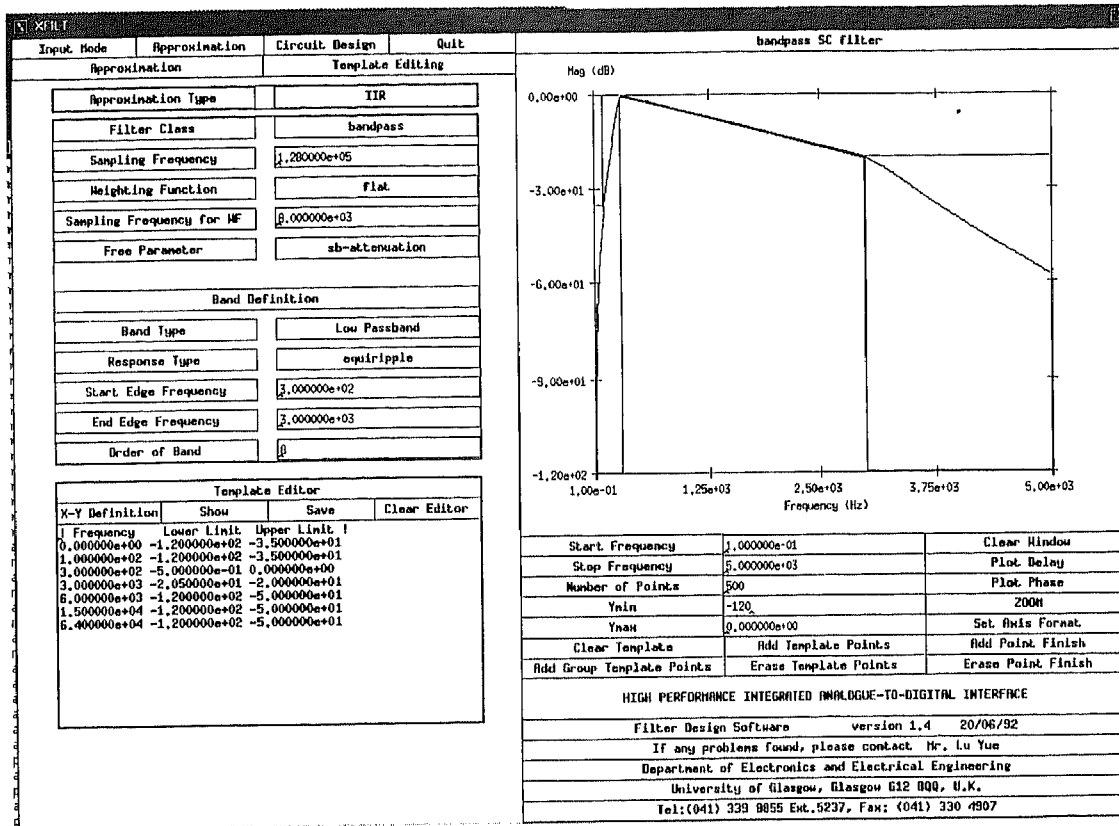


Fig. 3. Screen shot of XFILT showing template specification of lower passband and approximation of eighth order filter.

3. The order of the transfer function (particularly for ladders)

It is difficult to make general rules but usually some helpful defaults can be proposed, e.g., leapfrog structures are good for low-pass odd order designs, coupled-biquad, and LUD are best for bandpass, whereas cascade biquads excel at bandstop designs. Based on experimental running of a compiler over a range of specs a rule base can be established to make a reasonable suggestion of a structure. It is as important to preserve knowledge of successful as well as unsuccessful trial designs to avoid repeating the same mistakes. Otherwise this involves the "weak" optimization of a designer comparing possibilities by repeatedly running the compiler. This is not necessarily a bad thing in terms of designer "psychology" as it keeps him or her as a valuable part of the design process.

3.4. Module Synthesis and Layout

The interface between the block-level network description and silicon is filled by analog cell generators and

layout tools. These have been the slowest parts of the design cycle to automate because they deal with difficult nonlinear design problems and are near the fast-changing demands of the technology. Nevertheless, several systems have been presented dedicated to switched-capacitor filters [33]. These programs interpret the loading demands and required charging times from the netlist as specifications on amplifier and switch designs. They then dimension the devices in order to minimize the power and area of the circuit either by an optimizer or by using rearranged device equations. This is normally the least accessible part of the process. Analog block generators are making efforts to become more open to the designer but are still evolving [34-35]. Often the solution is just to use a handcrafted standard amplifier cell and to accept the overspecification of the design.

Analog routers capable of taking into account special requirements of sensitive and noisy nets, power supplies and variable analog transistor sizes and styles have been developed [36]. The inherent regularity of the filter topologies makes this a more amenable task than for more general classes of analog network. Several

dedicated layout strategies for switched-capacitor networks have been reported [18-21]. Normally a fixed topology is adopted (as in random logic layout), row of op amps, row of capacitors, row of switches. The routing problems are greatly simplified and dedicated algorithms can be used. Sensitive nets such as the connections to the virtual ground of the op amp are known in advance and crosstalk can be avoided. Capacitors are designed in units with constant area-perimeter ratios for high-accuracy matching. The user can often control the relative positions of switches, op amps, and capacitors.

4. Automated Filter Synthesis Example

In this section an example of a complete synthesis of a nonstandard filter will be illustrated from the XFILT compiler. Figure 3 shows a screenshot of a filter frequency response being defined to the filter compiler. The response characteristics are defined in stopband and passband by a piecewise template of lower and upper bounds on amplitude and delay. The order and

form of the transfer function are specified band by band (in this case the passband). Passbands can be assigned forms anywhere between equiripple and maximally flat and stopbands can have user-defined zero distributions. Of particular note is the 20 dB/decade slope in the passband for preemphasis. Three zeros have been placed at the origin to reduce capacitance spread over an elliptic-style zero distribution. Two notches are placed in the upper stopband but are not seen because of the frequency range of the plot. Figure 4 shows the designer evaluating different realizations of the filter. Depending on the designer's level of expertise so-called automatic, interactive or expert modes of design offer progressively more possibility for user intervention and customization of the circuit design. For example, capacitance spread can be reduced by different pole-zero pairing algorithms. In automatic mode, all possible combinations are tried, in interactive mode the designer chooses from among certain simple preprogrammed pairing rules aimed at making certain specific trade-offs while in expert mode he chooses the sequence by hand. The designer will normally also try a number of different

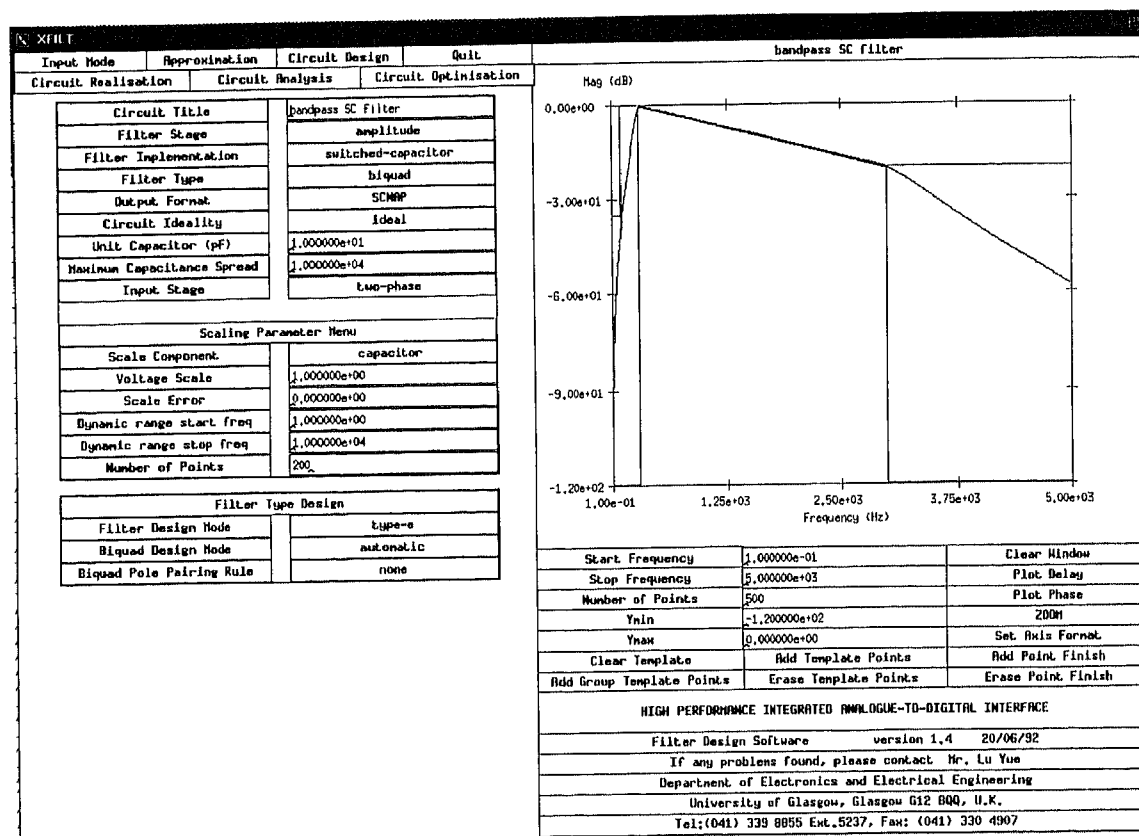


Fig. 4. Screen shot of XFILT showing circuit design of eighth order SC biquad.

circuit structures, comparing component area, power, dynamic range, noise, and sensitivity before arriving at a final choice. A selection of ladder simulation and cascade biquad structures are available. For this filtering problem, an eighth order cascade biquad circuit provides the best compromise between area and sensitivity considerations. Normally the final structure will have been selected after a number of iterations between approximation and circuit realization stages. The measured results of the filter (figures 5 and 6) show excellent agreement with the original template and ideal response. Finally a filter layout is shown in figure 7. The op amps and capacitors are standard cells taken from a library.

5. Conclusions

As regards the basic algorithms for filter design the field is now fairly mature. Moreover, the reliability with which filter compilers can produce good quality designs

has been established. More work is now necessary on unifying the framework and offering better access to the design facilities. Integration with digital tool environments and more standardization of data formats is to be hoped for in this direction. Greater gains in terms of design efficiency should now become available at a system level. For example, by allowing the designer to more easily observe trade-offs between different blocks in his filter system design and offering more guidance in the setup of reasonable specifications. This means using the compiler in fast first-cut design mode and by storing information about already explored design space. Filter compilers should lead the way toward more integrated analog system design exploration tools.

Acknowledgments

The assistance of Wolfson Microelectronics Ltd. and Lu Yue at the University of Glasgow in producing screen shots of XFILT is gratefully acknowledged.

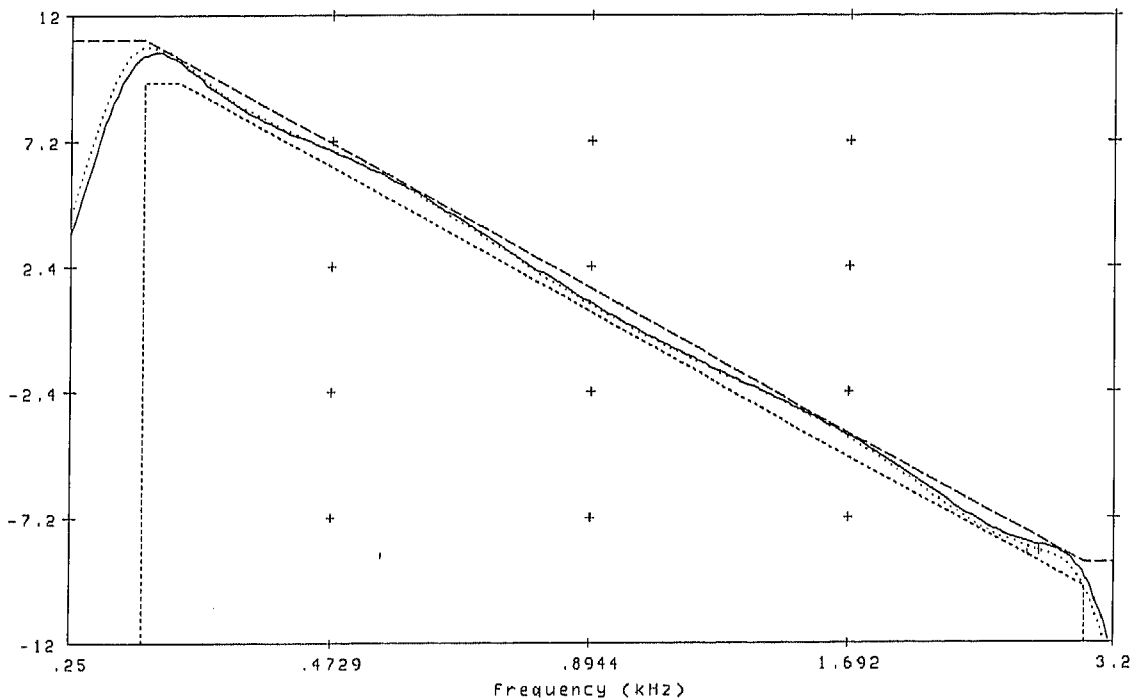


Fig. 5. Ideal passband response (solid) and measured passband response (dotted) of eighth order SC filter.

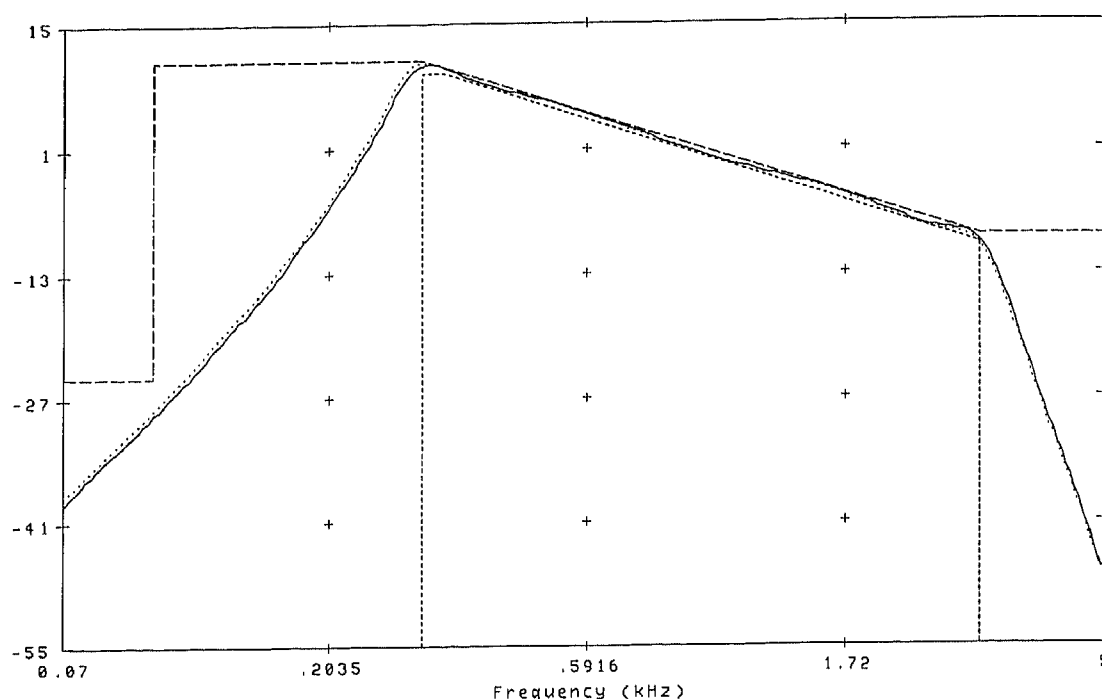


Fig. 6. Overall ideal response (solid) and measured response (dotted) of eighth order SC filter.

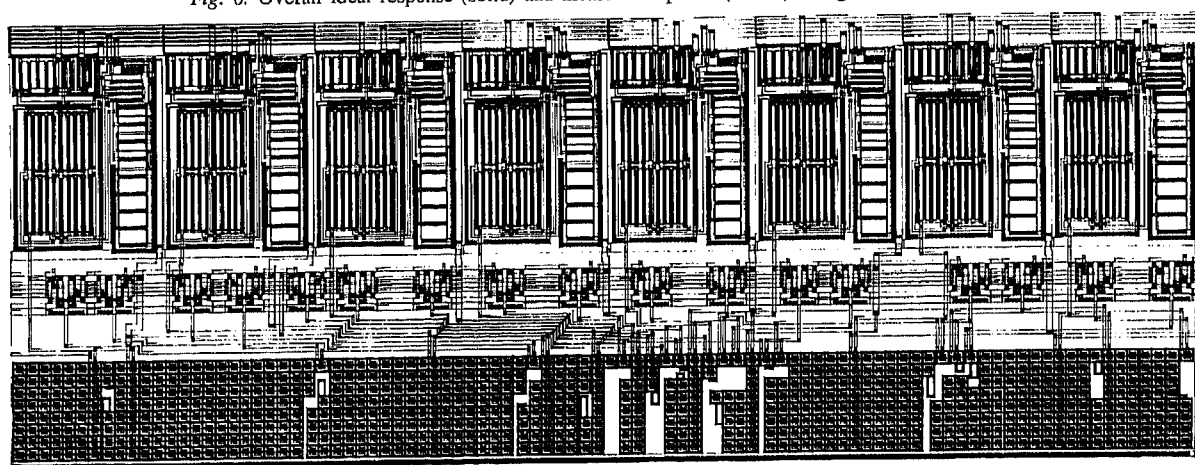


Fig. 7. Circuit layout of eighth order SC biquad filter.

References

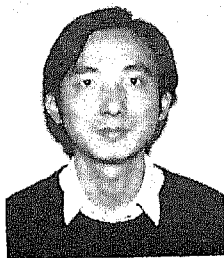
1. G. Szentirmai, *Computer-Aided Filter Design*, IEEE Press: New York, 1973.
2. E. Sanchez-Sinencio and J. Ramirez-Angulo, "AROMA: an area optimized CAD program for cascade SC filter design," *IEEE Trans. Computer-Aided Design*, Vol. CAD-14, pp. 296-303, 1985.
3. R. Schaumann, "Design of continuous-time fully-integrated filters: a review," *Proc. IEE, Part G*, Vol. 136, No. 4, pp. 184-190, 1989.
4. T.S. Fiez and J.D.J. Allstot, "CMOS switched-current ladder filters," *IEEE J. Solid-State Circuits*, Vol. SC-25, No. 6, pp. 1360-1367, 1990.
5. G.S. Moschytz, ed., *MOS Switched-Capacitor Filters: Analysis and Design*, IEEE Press: New York, 1975.
6. Lu Yue, N.P.J. Greer, and J.I. Sewell, "Software for the design of transconductor-capacitor filters and equalizers," *Proc. IEE Saraga Colloq. on Digital and Analogue Filters and Filtering Systems*, London, pp. 7/1-7/6, 1991.

7. N.K. Dutt and D.D. Gajski, "Design synthesis and silicon compilation," *IEEE Design Test Computers*, Vol. 7, No. 6, pp. 8-23, 1990.
8. G. Szentirmai, *S/FILSYN User Manual*, DGS Associates, Santa Clara, CA.
9. D.G. Nairn and A.S. Sedra, "Auto-SC, an automated switched-capacitor filter silicon compiler," *IEEE Circuits Dev. Mag.*, Vol. 4, pp. 5-8, 1988.
10. G.V. Eaton, D.G. Nairn, W.M. Snelgrove, and A.S. Sedra, "SICOMP: a silicon compiler for switched-capacitor filters," in *Proc. IEEE ISCAS*, pp. 321-324, Philadelphia, 1987.
11. C. Ouslis, M. Snelgrove, and A.S. Sedra, "A filter designer's filter design aid: filterX, in *Proc. IEEE ISCAS*, Singapore, pp. 376-379, 1991.
12. W. M. Snelgrove and A.S. Sedra, *FILTORG2: A Computer Filter Design Package*, Matrix Publishers: Illinois, 1979.
13. Y. Therasse, L. Reynders, R. Lannoo, and B. Dupont, "A switched capacitor filter compiler," *VLSI Systems Design*, pp. 85-88, 1987.
14. Mentor Graphics, *Autofilter Users Guide*, 1990.
15. J. Assael, P. Senn, and M.S. Tawfik, "A switched-capacitor filter silicon compiler," *IEEE J. Solid-State Circuits*, Vol. SC-23, pp. 166-174, 1988.
16. M.R. Kobe, E. Sanchez-Sinencio, and J. Ramirez-Rodrigo, "OTA-C filter silicon compiler" (to appear).
17. R.K. Henderson, Li Ping, and J.I. Sewell, "A design program for digital and analogue filters: PANDDA," in *Proc. ECCTD*, Brighton, UK, pp. 289-293, 1989.
18. R.P. Sigg, A. Kaelin, A. Murali, W.C. Black, Jr., and G.S. Moschytz, "Automated SC filter generation with a CMOS gate-array-type filter chip," in *Proc. IEEE ISCAS*, Helsinki, pp. 1725-1728, 1988.
19. K. Theqvist, H. Alarotu, P. Ylisirnio, and H. Kaskelma, "Adaptive simulated annealing used in the synthesis of switched-capacitor filters," in *Proc. IEEE ISCAS*, Helsinki, pp. 1729-1731, 1988.
20. M. Negahban and D. Gajski, "Silicon compilation of switched-capacitor networks," in *Proc. European Design Automation Conf.*, Glasgow, pp. 164-168, 1990.
21. A. Barlow, K. Takasuka, Y. Nambu, T. Adachi, and J. Konno, "An integrated switched capacitor filter design system," in *Proc. IEEE Custom Integrated Circuits Conference*, Boston, 1989.
22. Bell Northern Research, *SWCAP Tools Reference Manual*, 1990.
23. M. Ismail and J. Franca, ed., *Introduction to Analog Design Automation*, Kluwer: Boston, Chap. 6, 1990.
24. R. W. Daniels, *Approximation Methods for Electronic Filter Design*, McGraw-Hill: New York, 1974.
25. R.K. Henderson, Li Ping, and J.I. Sewell, "Extended Remez algorithms for filter amplitude and group delay approximation," *IEE Proc.*, Part G, Vol. 138, No. 3, pp. 289-300, 1991.
26. G. Cortellazo and M.R. Lightner, "Simultaneous design in both magnitude and delay of IIR and FIR filters based on multiple criterion optimisation," *IEEE Trans. Acoustics Speech Signal Process.*, Vol. ASSP-32, No. 5, pp. 949-967, 1984.
27. C. Xuexiang, E. Sanchez-Sinencio, and R.L. Geiger, "Pole-zero pairing strategy for area and sensitivity reduction in SC filters," in *Proc. IEEE ISCAS*, pp. 690-611, 1986.
28. H.J. Orchard, "Filter design using iterated analysis," *IEEE Trans. Circuits Systems*, Vol. CAS-33, No. 11, pp. 1089-1096, 1985.
29. A.S. Sedra and P.O. Brackett, *Filter Theory and Design: Active and Passive*, Pitman: London, 1979.
30. Li Ping, R.K. Henderson, and J.I. Sewell, "A methodology for integrated ladder filter design," *IEEE Trans. Circuits Systems*, vol. CAS-38, No. 8, pp. 853-868, 1991.
31. A. Murali and G.S. Moschytz, "Performance optimization of elliptic SC ladder filters using signal flow graph transformations," in *Proc. IEEE ISCAS*, pp. 2205-2208, 1990.
32. W.M. Snelgrove and A.S. Sedra, "Synthesis and analysis of state-space active filters using intermediate transfer functions," *IEEE Trans. Circuits Systems*, Vol. CAS-33, pp. 287-301, 1986.
33. T. Kaneko, Y. Akazawa, and M. Natagani, "Switched capacitor and active-RC filter layout using a parameterizable generator," in *Proc. IEEE ISCAS*, Singapore, pp. 2012-2015, 1991.
34. M.G.R. Degrauwe, B.L.A.G. Goffart, C. Meixenberger, M.L.A. Pierre, J.B. Litsios, J. Rijmenants, O.J.A.P. Nys, E. Dijkstra, B. Joss, M.K.C.M. Meyvaert, T.R. Schwartz, and M.D. Pardoën, "Towards an analog system design environment," *IEEE J. Solid-State Circuits*, Vol. SC-24, No. 3, pp. 659-671, 1989.
35. R. Harjani, R.A. Rutenbar, and L.R. Carley, "OASYS: A framework for analog circuit synthesis," *IEEE Trans. Computer-Aided Design*, Vol. CAD-8, No. 12, pp. 1247-1266, 1989.
36. J.M. Cohn, D.J. Garrod, R.A. Rutenbar, and L.R. Carley, "KOAN/ANAGRAM II: new tools for device level analog placement and routing," *IEEE J. Solid-State Circuits*, Vol. SC-26, No. 26, pp. 330-342, 1991.



Robert K. Henderson received the B.Sc. (Hons.) degree in electronics and electrical engineering with computing science and the Ph.D. degree in electrical engineering in 1986 and 1990 respectively, both from the University of Glasgow, Scotland.

From 1989-1990 he was engaged at the University of Glasgow as a research assistant. He is currently a research engineer at the Centre Suisse d'Electronique et de Microtechnique S.A., Neuchâtel, Switzerland. During his doctoral work he co-authored a compiler for analogue filters called PANDDA. His research interests are in the areas of analog signal processing, numerical algorithms and computer-aided circuit and system design.



Li Ping received the B.S. degree from Nanjing Institute of Posts and Telecommunications, China, in 1981, M.S. degree from Shanghai Jiao Tong University, China, in 1984, and Ph.D. degree from Glasgow University, U.K., in 1990.

He was a research associate at Shanghai Institute of Railway Technology from 1984 to 1986 and a Royal Society-British Telecom Research Fellow at Glasgow University in 1987. From 1988 to 1990 he was a consultant and later an engineer with Wolfson Microelectronics, Edinburgh, U.K. Since 1990 he has been a lecturer at the Department of Electrical and Electronic Engineering, the University of Melbourne, Australia. His research interests are in analog integrated circuits and telecommunications.



John I. Sewell received the B.Sc. (Hons.) degree in Electrical Engineering from the University of Durham in 1963 and the Ph.D. degree from the University of Newcastle upon Tyne in 1966.

From 1966 to 1968 he was a Senior Research Associate at the University of Newcastle upon Tyne. During the period 1968 to 1985 he held faculty positions at the University of Hull. In 1985 he was appointed as Professor of Electronic Systems in the University of Glasgow and in 1990 he became Dean of Engineering. His research interests are concerned with the development of CAD design tools for front-end signal processing including Si and GaAs implementations, integrated filters and signal converters.

Dr. Sewell has served on numerous IEEE committees in the UK and also on the IEE Professional Group on Circuit Theory. He has been involved with organizing committees for a number of Conferences, Colloquia and Workshops under the auspices of IEEE and IEE. He is Fellow IEE and Senior Member IEEE.