Advanced Simulation Methods for Switched Networks

A Thesis submitted to the

Faculty of Engineering

of the University of Glasgow

for the degree of

Doctor of Philosophy

by

ZHONG QING SHANG

March 1995

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

SUMMARY

This thesis addresses several difficult simulation problems that have arisen in the rapid development of analogue sampled-data systems with the drive to more circuit complexity and demands by designers for more sophisticated simulation tools the current techniques and related software packages were not capable of providing satisfactory solutions.

Nonideal sensitivity analysis is found to be very useful in filter design and optimisation. A method for generating derivatives of the extended state transition matrix and the excitation response matrix is proposed. Most nonidealities of interest, such as gain and bandwidth product of the opamp, input and output impedances of the opamp, switch resistance and nodal parasitics can be exactly investigated. Group delay sensitivity and overall sum of sensitivities evaluation capabilities, offer accurate measures for choosing better circuit realisation in terms of low sensitivity properties.

Noise analysis of sampled-data systems is always believed to be very time consuming. An efficient and systematic noise analysis technique is presented. The use of adjoint techniques, spectrum analysis together with highly efficient software implementation makes the noise analysis of practical circuits feasible on a modest workstation. It is also suggested that various accurate noise models are used in order to obtain sensible results.

A polynomial interpolation method is used for semi-symbolic analysis of large nonideal switched linear networks to which other symbolic approaches cannot apply. Measures for retaining the accuracy of the interpolation method are discussed and the results are very encouraging. The application of symbolic analysis to evaluate the noise behaviour is described. The comparison of theoretical computation costs is introduced, from which the potential efficiency of the symbolic method can be estimated.

Nonlinear time domain analysis of sampled-data networks is usually performed by a general purposed circuit simulator. Unfortunately, this is very extravagant in computer time. An effective strategy is proposed from which the dominant nonlinearities present in sampled-data circuits can be efficiently analysed. The use of a piecewise linear description of the nonlinear characteristic allows many efficient linear techniques to be employed.

Oversampling techniques relax the precision requirement of analogue circuits and provide high resolution of analogue to digital conversion. However the simulation of oversampled sigma-delta modulator is almost impossible by traditional methods. A generalisation of present periodic linear techniques to handle this problem is presented. A new definition of network state is proposed and a series of efficient computer implementation strategies are described. This has facilitated the linking of a high refined switched network simulator to standard commercial logic simulator to provide a good mixed-mode simulation.

献给我的父母：李桂芳、尚衍信

I dedicate this thesis to my parents,

Gui-Fang Li and Yan-Xin Shang

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor Professor J. I. Sewell for his encouragement and considerate guidance of this work.

Thanks are also due to the facilities provided by the Department of Electronics and Electrical Engineering.

The financial support that made this research possible was provided by the Sino-British Friendship Scholarship scheme which is sponsored by the ODA, Y. K. Pao Foundation and the Chinese State Education Committee and is therefore most gratefully acknowledged.

I am gratitude to my friend and colleague Lu Yue for many useful discussions. The assistance of David Hossack in part of this work is also greatly appreciated.

Many thanks also to Ann Mackinnon, Dugald Campbell and Stephen Gallacher for their indispensable help in sorting out the problems of working with computers.

I am grateful also to Dr Bill Redman White and Judith Richardson of Philips Semiconductors Limited for providing the motivation for some of this work and for many helpful discussions and assistance.

Finally I would like to specially thank my wife Gui Fang Liu for her love and meticulous care during this research. My special thanks also to my son Xiu Hua Shang whose companionship has been a great source support to me over these years.

TABLE OF CONTENTS

CHAPTER 3: NOISE ANALYSIS

CHAPTER 4: SYMBOLIC ANALYSIS

CHAPTER 5: NON-LINEAR SWITCHED NETWORKS

CHAPTER 6 MIXED-MODE SIMULATION

CHAPTER 7: CONCLUSIONS

APPENDICES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION AND GENERAL AIM

Over the past fifteen years, switched capacitor (SC) techniques dominated analogue MOS integrated circuits for signal processing due to their several salient features, such as the capability of full integration using MOS technology and being VLSI compatible, which directly result in profound effects on the economy of application. In addition to this integratable character of SC circuits, other factors such as small size, light weight, high reliability and small dc bias power requirement prevail. The accuracy is only determined by the capacitor ratios which offers the realisation of high-selectivity responses with good accuracy and stability. SC circuits are sampled-data systems, but unlike digital filters, they handle signals in analogue form; in other words the amplitude of the sampled voltages are signal values, without the use of any encoding. Hence the basic operations (multiplication, addition, delay) needed in signal processing are performed directly; this results in a speed advantage over digital circuits. Driven by these remarkable features, SC techniques have been extensively used in the analogue signal processing area. A great number of design techniques for synthesising high-performance SC circuits have been developed and have resulted in a number of filter design systems [1-7]. Although the design techniques can be considerably different, one process common to all design procedures is that of simulation and the tools used play a very important role in evaluating the final performance of the system. During a filter design, the simulation tool is accessed frequently, as it can provide the circuit response at various design stages on demand. It is reasonable to assume that good simulation facilities exist for the computation of frequency response of even quite large SC networks with multirate clocks and that accurate determination of the effects of linear circuit nonidealities is included. Also the computation of ideal sensitivities, such as those due to circuit capacitances is well established. Some form of global sensitivity computation is invaluable to the comparison of different filter structures in the selection of a final topology for realisation. At a further stage in the design process, the effects of circuit nonidealities such as switch resistance, gain / bandwidth product of opamps can be most conveniently assessed by the computation of the sensitivity of circuit response with respect to these parameters. However, an efficient sensitivity analysis method which is able to satisfy all above requirements is conspicuously absent. It became one of the problems addressed in this thesis.

Recently, the trend in integrating the entire signal processing system (analogue and digital) onto a single chip is apparent. It has been realised that the SC techniques are not fully compatible with present standard digital CMOS processing technology. For

example, high quality linear capacitors may become unavailable as process dimensions shrink to deep submicron range; meanwhile the reduction in supply voltages due to the submicron process will directly reduce the maximum voltage swing available to switched-capacitor circuits and consequently narrow the dynamic range even further. Alternatively, the shrinking geometry and supply voltages will increase the operating speed of digital circuits, whereas the realisation of high speed high gain operational amplifiers will become more difficult. A new analogue sampled-data signal processing technique - switched current (SI) was proposed [8-10] to overcome the above difficulties and is highly suited to mixed analogue and digital applications. The need for CAD tools to support the development of the new technique is obvious. However, although many simulation tools have been used, the majority of these programs were specially designed for ideal SC networks (all switches are assumed to have infinite off-resistances and zero on-resistances and the amplifiers are assumed to have infinite bandwidth). Besides, they are based on a charge-conservation law and no current information is available. Therefore it is not surprising that only two years ago, it was still claimed that no switched current specific simulation tools were available [11]. The provision of such tools became an important issue during this research.

Efficient noise analysis of sampled-data circuits has always been considered as a very difficult task. Not only because of the sampling feature which causes the folding of wideband white noise from high frequency bands to baseband and the associated high computational cost in simulating the process, but also the inherent problem in accurately modeling the noise characteristics of noise elements and building blocks. Over the years noise analysis for SC circuits has been discussed frequently [12-18], however most of the techniques presented are only suitable for small circuits. Applications to large practical networks are seldom mentioned. Conventionally, the system equations have to be solved repeatedly to obtain transfer functions from each noise source to the output. Furthermore, as the noise bandwidths are normally orders of magnitude higher than the sampling frequency, a very large number of extra system solutions are needed to cover the wide frequency range for taking folding effects into account. Therefore, very efficient general methods are crucial to noise analysis. Efforts towards these aspects are discussed in this thesis.

An alternative strategy to improve the noise calculation efficiency is concerned with symbolic analysis methods since one of the ideal applications of symbolic analysis is repetitive evaluation of frequency response from polynomials. Historically, symbolic analysis methods were proposed as complementary to numerical techniques. Since such methods can provide designers with additional insight into the circuit behaviour

which enables them to interactively improve their circuits, to generate macromodels for analogue automation design and so on, there is still some attention by a number of researchers today. Many algorithms and techniques have been developed. Unfortunately, the restrictions of circuit size are still imposed on these approaches and very little progress has been made with medium to large networks. Whilst attention has been focused on continuous-time analogue circuits, some work has been carried out on the symbolic analysis of ideal SC networks[19] and only exploratory investigations have been reported into the semi-symbolic analysis of nonideal SC networks[20] and noise performance [21]. For large networks of any type, the only practical scheme is a semi-symbolic one, when polynomials in s or z or both, with purely numeric coefficients, are generated. A polynomial interpolation method is found very suitable in this case. However, it has only been applied to continuous-time and ideal SC networks. The application of symbolic analysis to large nonideal switched linear networks still remains an open research area.

In practice, when SC or SI networks are realised as integrated circuits, various unavoidable imperfections occur. These have to be considered carefully and minimised or eliminated by appropriate design techniques, otherwise low yields in manufacture could be expected. The nonideal linear imperfections such as switch resistance (R_{on} and R_{off}), and the finite gain and bandwidth of the opamp can be properly investigated by general linear sampled-data network analysis tools[22]. However, nonlinear imperfections, for instance, the finite slew-rate and output swing of opamp, clock feed-through and charge injection introduced by switches, usually require recourse to a general purpose circuit simulator. The harmonic distortion products can be obtained by using some FFT algorithm. Unfortunately, a very large amount of computation time is required [23] due to the evaluation of sophisticated nonlinear device models and a large number of sharp waveform transitions in a switched network. Therefore it is essential to provide designers with efficient tools for evaluating these effects and an enhanced piecewise linear technique suitable for these networks is developed.

Recently, oversampling techniques have been receiving increased attention as candidates for high-resolution analogue-to-digital conversion, since circuit precision requirements can be significantly relaxed by oversampling input waveforms, coarse quantization and digital decoding[24-29]. The performance of the oversampled sigma-delta modulator is determined by the ratio of oversampling as well as the order of the modulator. For a given frequency range, the best achievable performance is very dependent upon the maximum sampling frequency at which the modulator can work properly. However, the simulation of the sigma-delta modulators is extremely

difficult, due to the fact that tens of thousands of clock cycles are required in order to obtain meaningful measures of signal to noise ratio (SNR) or signal to distortion ratio. Normally, using a transistor-level circuit simulator, SPICE for instance, the transient analysis of a single clock cycle of the analogue integrator (SC or SI) section may take several minutes of CPU time on a modest workstation. In this case, the computation of detailed SNR curves would not be realistic. Alternatively, difference equation models for the integrator circuits are introduced which can reduce the simulation time dramatically, but at the cost of ignoring the effects of nonidealities in the circuit. Therefore, efficient circuit simulation techniques which can fully account for the nonidealities of the system are essential to the design of oversampled sigma-delta A/D converters. This problem is also tackled.

The aim of this thesis is to address the problem of developing advanced simulation methods for solving several difficult simulation tasks required in the modern filter and data converter design. The techniques presented in the succeeding Chapters will offer various solutions and form the basis of several efficient software packages for analogue sampled-data networks analysis called SCNAP4, SCNAPDIS and SCNAP5, respectively.

1.2 COMPARISON OF SIMULATION SOFTWARE

There are many simulation software packages available for the analysis of ideal switched linear networks[30], however many of them do not provide any solutions to the new challenges. The most well-known simulators for general sampled-data networks are examined. Among them, SCNAP4 is the package in which many advanced techniques discussed in this thesis are implemented. Table (1.1) offers a brief comparison of their main features.

SWITCAP

SWITCAP is a well known general simulation program for analysing switched capacitor networks [31-32], it was developed by Columbia University in 1987. Many copies of SWITCAP are still being used world-wide and it is generally regarded as a benchmark standard. Table (1.2) illustrates the performance comparison of SCNAP4 and SWITCAP in the frequency domain.

Table (1.1) Comparison of simulation software

	SWITCAP [31-32] (87)	SWAP2 [34-36] (84)	QUICKSCNAP [33] (87)	WATSNAP [37] (87)	SCNAP4 [22] (94)
nonideal sensitivity	No	No	No	Yes	Yes
noise analysis	No	Yes	No	No	Yes
spectrum analysis	Yes	Yes	No	Yes	Yes
transient analysis	Yes	Yes	Yes	Yes	Yes
frequency analysis	Yes	Yes	Yes	Yes	Yes
switched capacitor	Yes	Yes	Yes	Yes	Yes
switched current	No	Yes	Yes	Yes	Yes
continuous time	No	No	Yes	No	Yes
extension to nonlinear	No	Yes	No	No	Yes
extension to mixed-mode	Yes	No	No	No	Yes

Table (1.2) Comparison of SWITCAP and SCNAP4

circuit name †	circuit size	clock phases	per-frequency point (sec)		speed up
			SCNAP4	SWITCAP (ideal)	
bp2	14	2	0.0004	0.0102	25.50
but3	10	3	0.0004	-	-
cbp6	35	2	0.0031	0.0211	6.81
elp15	87	2	0.0175	0.1082	6.18
elp3	13	12	0.0023	-	-
elp6	35	2	0.0029	0.0386	13.31
fet7	35	6	0.0079	-	-
lp15	87	2	0.0181	-	-
lp5	28	2	0.00175	0.0228	13.03
nos11	47	4	0.011	-	-
nos5	23	4	0.0027	-	-
spft	99	36	0.2966	-	-

Note: † These are a standard set of examples adopted previously [33] by the University of Glasgow as a benchmark for testing SC simulations.

‡ Above data obtained on SUN-Sparc 10 station.

From the table above, it shows that SCNAP4 is at least 6 times faster than SWITCAP. Considering that SWITCAP only performs ideal analysis while SCNAP4 always performs full nonideal analysis, the comparison result is even more remarkable. It is also seen that for some circuits, SWITCAP fails to give any answers due to matrix singularity problems while SCNAP4 can run them all without modification to the circuit netlist. In addition, SWITCAP is based on charge conservation law, hence it is not applicable to analysis switched current circuits.

QUICKSCNAP

QUICKSCNAP was developed by Glasgow University in 1987 and was written in FORTRAN [33]. It performs full nonideal analysis for switched capacitor circuits. QUICKSCNAP is not a stand alone software, it only accepts processed netlist

information. The circuit description files must be processed separately by another program called CADRE which was written in PASCAL.

Table (1.3) Comparison of QUICKSCNAP and SCNAP4

circuit name	symbolic pre-proc.(sec)		numerical pre-proc.(sec)		time per-frequency point (sec)	
	QUICK	SCNAP4	QUICK	SCNAP4	QUICK	SCNAP4
bp2	0.06	0.02	0.08	0.06	0.0056	0.0014
but3	0.06	0.01	0.04	0.04	0.0060	0.00105
cbp6	0.24	0.07	0.54	0.38	0.025	0.0074
elp15	1.80	0.60	5.3	3.16	0.1414	0.0445
elp3	0.08	0.02	0.44	0.30	0.0146	0.0056
elp6	0.22	0.08	0.60	0.39	0.0230	0.0069
fet7	0.24	0.08	1.66	0.93	0.0512	0.0194
lp15	1.82	0.84	5.38	3.16	0.1468	0.0504
lp5	0.18	0.06	0.36	0.26	0.0164	0.00450
nos11	0.46	0.20	2.90	1.70	0.0732	0.00262
nos5	0.12	0.02	0.54	0.36	0.019	0.00645
spft	3.02	0.88	171.98	81.03	1.794	0.7549

Note: Above data obtained on SUN-Sparc ELC station.

Before SCNAP4 was developed, it was generally believed QUICKSCNAP to be the fastest SC simulation program available [33]. But from Table (1.3), it is evident that SCNAP4 is now at least three times faster then QUICKSCNAP. More importantly, SCNAP4 can easily extend its applications into other simulation areas, such as nonideal sensitivity analysis, noise analysis and mixed-mode simulation, while QUICKSCNAP cannot be taken any further because of its inflexible data structures.

SWAP2

SWAP2 [34-36] was developed by the Katholieke Universiteit Leuven, ESAT Laboratorium. The program consists of ideal algorithms and an additional nonideal analysis facility (called intermediate-level frequency analysis). It runs on a VAX computer under VMS operating system and is not available on our SUN workstation. Hence it was not possible to make a direct comparison of SWAP2 with SCNAP4 in the same environment. However, experience with SWAP2 reveals that for ideal and

small SC circuits, the efficiency of simulation is acceptable, but when nonideal information is required, it was very slow. For example, noise analysis using SWAP2 for a medium sized SC filter took a day and half while SCNAP4 only spent a few minutes for the same size circuit though the computer were not equivalent.

WATSNAP

WATSNAP [37] was developed by Waterloo University at approximately the same time as QUICKSCNAP. The basic theory involved is very similar to SWAP2, but WATSNAP performs full nonideal analysis. An evaluation of WATSNAP and SCNAP4 was carried out independently by the CADENCE company over a multi-aspect comparison [38]. It was stated that SCNAP4 is superior to WATSNAP in terms of efficiency.

The evaluation of SCNAP4 was also done separately by Philips Semiconductor Ltd. and the report [39] shows SCNAP4 to be the most efficient program amongst all the software packages considered.

SPICE

SCNAP4 can analyse not only sampled-data circuits, but also general linear networks. For example, active RC, LC ladders can be efficiently analysed and the results compared with SPICE. Without losing accuracy, SCNAP4 works more efficiently. Some simulation results are illustrated in the SCNAP4 user's guide. A set of continuous time circuits are examined and comparative results are presented in the Table (1.4) and Table (1.5).

Table (1.4) Comparison in frequency domain

circuit name	per-frequency point (sec) SPICE	per-frequency point (sec) SCNAP4	speed up
elp9	0.0268	0.0018	14.89
lc5	0.0089	0.0004	22.25
lc7	0.0116	0.0006	19.33
lc10	‡	0.0008	—

Table (1.5) Comparison in time domain

circuit name	sweep (0-500us) SPICE (sec)	sweep (0-500us) SCNAP4 (sec)	speed up
elp9	2.18	0.21	10.38
lc5	0.46	0.03	15.33
lc7	0.66	0.02	33.00
lc10	‡	0.04	—

‡ SPICE fails at matrix reorder.

The data in Table (1.4) and (1.5) were obtained on a SUN-Sparc ELC station. In both time and frequency domain, SCNAP4 is much faster than SPICE.

At the outset of the work it was decided that any attempt to modify QUICKSCNAP would not be wise as this could not provide a general basis for future coherent development. Neither was a simple rewrite of QUICKSCNAP in C a sensible alternative. Obviously some of the very robust techniques developed for QUICKSCNAP could be taken advantage of, but otherwise the design of SCNAP4 would begin again, with an entirely new architecture and data structure, be written in C with the intention of being highly portable within UNIX and other environments.

In summary, SCNAP4 is a very powerful and versatile program for analogue filter simulation. The program is capable of analysing SC, SI, OTC and all types of circuit encountered during the filter design. It also provides the analysis facilities that a filter synthesis system would require. Many effective techniques have been incorporated in SCNAP4, such as interpretable code generation, a large amount of frequency iindependent pre-processing, sparse matrix technique, numerical Laplace transform iinversion, adjoint system technique and Hessenburg approach [23][40]. All these advanced techniques are carefully implemented on the computer and many efficient strategies have been involved. SCNAP4 is a large modularised program. It is designed to be flexible and consistent with high performance. The program architecture and data structure are easy to understand and flexible to extend. The successful extension to the symbolic analysis, nonlinear analysis and even mixed-mode simulation confirms that new circuit elements, new types of simulation and new analysis algorithms can be added easily.

1.3 OUTLINE OF THE THESIS

The motivation for the research presented in this thesis derives from several difficult simulation tasks in the analogue sampled-data and mixed-mode circuit design. A general outline of the contents and structure of the thesis now follows:

Chapter 2 covers the derivation of an efficient sensitivity analysis method for nonideal switched linear networks. The problem is formulated mathematically and the difficulty of applying adjoint network techniques directly is discussed. A method of generating derivatives is then presented. A new definition for switch resistance sensitivity is given and the scheme to calculate the gain and bandwidth product sensitivity of opamps is demonstrated. Methods of calculating the group delay and sum sensitivities are presented and they are particularly useful for filter synthesis.

Chapter 3 addresses the problem of efficient noise analysis of general analogue sampled-data networks. The system equations are generalised to allow multi-independent sources, a spectrum analysis technique has been developed to take into account the folding effects which are common to switched linear networks. Two types of noise (white and flicker) are considered. Computer implementation requires only one system solution to obtain the network noise behaviour.

In Chapter 4 the application of semi-symbolic analysis to the large nonideal switched linear networks is considered. The state of the art of symbolic analysis methods is examined. Only the polynomial interpolation method is adequate for application to large nonideal switched linear networks. Measures to retain the accuracy of the polynomial interpolation method are discussed. Application to noise calculation is demonstrated. An efficiency assessment of the symbolic method versus the numerical approach is carried out by comparing theoretical computation costs.

Chapter 5 considers the efficient time domain analysis of nonlinear switched networks. Dominant nonlinearities in SC and SI circuits are discussed. Piecewise linear approximations of the element nonlinearities are introduced and hence the original nonlinear network is replaced by a piecewise linear network. A modified Katzenelson algorithm is developed which makes the re-use of efficient linear techniques feasible. Computer implementation is also described.

Chapter 6 is concerned with the efficient time domain analysis of mixed-mode circuits. The primary approaches of mixed-mode simulation are reviewed. A new

definition of network state is proposed, from which techniques for periodically switched networks are generalised to handle non-periodic situations. The principles of communication between the analogue and digital simulator are described. A number of efficient computer implementation strategies are presented.

Finally the summary of the major achievements of the research presented and suggestions for future extensions are given in Chapter 7.

1.4 STATEMENT OF ORIGINALITY

The following major results of the research in this thesis are, as far as is known, original and have been published or submitted for publication:

- In Chapter 2, the derivation of exact sensitivity analysis methods for nonideal switched linear networks. The definition of the switch resistance sensitivity and the sensitivity calculation of the GB product of opamps. The derivation of the method for calculating the group delay.

[Z. Q. Shang and J. I. Sewell, "Efficient sensitivity analysis for large nonideal switched capacitor networks," Proc. of IEEE ISCAS, Chicago, May 1993, pp.1405-1407]

- In Chapter 3, the use of the adjoint network techniques for efficient noise analysis of general nonideal switched linear networks. The extension of the Hessenburg approach to the transposed system and the derivation of the method for multi-independent sources which is essential for noise analysis and time domain analysis. The development of the spectrum analysis method which is crucial to the incorporation of folding effects.

[Z. Q. Shang and J. I. Sewell, "Efficient noise analysis methods for large nonideal SC and SI circuits," Proc. of IEEE ISCAS, London, June 1994, pp.5.565-5.568]

- In Chapter 4, the development of a semi-symbolic analysis technique based on polynomial interpolation method for large nonideal switched linear networks. The introduction of the concept of theoretical computation cost which can be used to compare the efficiency of symbolic methods against numerical approaches. The application of symbolic analysis to the calculation of circuit noise behaviour.

[Z. Q. Shang and J. I. Sewell, "Accurate semi-symbolic analysis of large nonideal switched linear networks," to be published in Proc. of IEEE ISCAS-95]

- In Chapter 5, the development of the time domain analysis method based on piecewise linear approximation of the nonlinear characteristics of the elements. The derivation of a modified Katzenelson algorithm and the extension of present efficient linear techniques for solving nonlinear problems.

[Z. Q. Shang and J. I. Sewell, "Efficient analysis of some non-linearities in SC and SI filter networks," Digest of the IEE 14th Saraga Colloquium on "Digital and Analogue filter and filtering systems," London, 1994, pp.10/1-10/5]

- In Chapter 6, the development of a mixed-mode circuit simulation method for mixed analogue sampled-data circuits and digital networks. The introduction of a new definition of the network state and its efficient computer expression. The design of a breakpoint table which is crucial to event handling, the strategy of the efficient evaluation of the clock state (periodic and non-periodic). The development of an elegant interface to a commercial digital simulator.

[Z. Q. Shang and J. I. Sewell, "Efficient analysis of non-periodic analogue sampled-data circuits in a mixed mode environment," in preparation.]

REFERENCES

[1] G. Szentirmai, "FILSYN - A general purpose filter synthesis program," Proc IEEE, vol. 65, no. 10, Oct. 1977, pp.1443-1458

[2] W. M. Snelgrove and A. S. Sedra, "FILTOR2: A computer filter design package," Matrix publishers, I11., 1979.

[3] G. Szentirmai, S/FILSYN user manual, DGS Associates, Santa Clara, 1988

[4] D. G. Nairn and A. S. Sedra, "Auto-SC, Automated switched-capacitor ladder filter design program," IEEE Circuits and Devices Magazine, March 1988, pp.5-9

[5] G. V. Eaton, D. G. Nairn, W. M. Snelgrove and A. S. Sedra, "SICOMP: A silicon compiler for switched-capacitor filters," Proc. IEEE ISCAS, Philadelphia, 1987, pp.321-324

[6] R. K. Henderson, Li Ping and J. I. Sewell, "A design program for digital and analogue filters: PANDDA," Proc. ECCTD'89, Brighton, Sept. 1989, pp.289-293

[7] Y. Lu, R. K. Henderson and J. I. Sewell, "XFILT: an-Xwindow based modern filter and equliser design system," Proc. ECCTD' 93, Davos, 1993, pp.305-310

[8] J. B. Hughes, N. C. Bird and I. C. Macbeth, "A new technique for analog sample data signal processing," Proc. IEEE ISCAS, Portland, May 1989, pp. 1584-1587

[9] G. Wegmann and E. A. Vittoz, "Very accurate dynamic current mirrors," Electronics Letter, vol. 25, May 1989, pp.644-646

[10] S. J. Daubert, D. Vallancourt and Y. P. Tsividis, "Current copier cells," Electronics Letters, vol. 24, Dec. 1988, pp. 1560-1562

[11] N. C. Battersby, "Switched-current techniques for analogue sampled-data signal processing," PhD Thesis, Imperial College, University of London, 1993

[12] J. H. Fischer, "Noise sources and calculation techniques for switched capacitor filters," IEEE J. of Solid-State Circuits, vol. SC-17, No.4, Aug. 1982, pp.742-752

[13] C. A. Gobet and A. Knob, "Noise analysis of switched capacitor networks", IEEE Trans. on CAS, vol. CAS-30, No.1, Jan. 1983, pp.37-43

[14] J. Rabaey, J. Vandewalle and H. De Man, "A general and efficient noise analysis technique for switched capacitor filters", Proc. of IEEE ISCAS, Newport Beach, May. 1983, pp.570-573

[15] S. W. Li, F. W. Stephenson and J. Velazquez-Ramos, "Noise analysis of switched capacitor filters," Proc. of IEEE ISCAS, Montreal, May. 1984, pp.1312-1315

[16] C. K. Pun, J.I.Sewell and A. G. Hall, "Noise analysis for switched capacitor networks in symbolic form," Proc. of 29th Midwest symposium on circuits and systems, Louisville, Aug. 1985, pp.807-810

[17] J. Gotte and C. Gobet, "Exact noise analysis of SC circuits and an approximate computer implementation," IEEE Trans. on CAS, vol. CAS-36, No.4, Apr. 1989, pp.508-521

[18] L. Toth and K. Suyama, "Exact noise analysis of 'ideal' SC networks", Proc. of IEEE ISCAS, New Orleans, May 1990, pp.1585-1588

[19] D. G. Johnson and J. I. Sewell, "Improved z plane polynomial interpolative analysis of switched capacitor networks," IEEE Trans. CAS. vol. CAS-31, no.7, 1984, pp.666-668

[20] C. K. Pun and J. I. Sewell, "Symbolic analysis of ideal and nonideal switched capacitor networks," Proc. IEEE ISCAS, Kyoto, June 1985, pp. 1165-1168

[21] C. K. Pun, A. G. Hall and J. I. Sewell, "Noise analysis of switched capacitor networks in symbolic form," Proc. 29th Midwest Symp., Lincoln, Na., August 1986, pp. 807-810

[22] Z. Q. Shang and J. I. Sewell, "SCNAP4 users's guide (version 1.6)," Dept. of Electronics and Electrical Engineering, University of Glasgow, Aug. 1994

[25] J. C. Candy, Y. C. Ching and A. S. Alexander, "Using triangular weighted interpolation to get 13-bit PCM from ∑Δ modulator," IEEE Trans. Commun., vol. COM-24, Nov. 1976, pp.1268-1275

[26] J. C. Candy and O. J. Benjamin, "The structure of quantization noise from sigma-delta modulation," IEEE Trans. Commun., vol. COM-29, Sep. 1981, pp.1316-1323

[27] J. C. Candy, "A use of double integration in sigma-delta modulation," IEEE Trans. Commun., vol. COM-33, Mar. 1985, pp.249-258

[28] R. M. Gray, "Oversampled sigma-delta modulation," IEEE Trans. Commun., vol. COM-35, May. 1987, pp.481-489

[29] K. Uchimura, T. Hayashi, T. Kimura and Y. Inable, "Oversampling A-to-D and D-to-A converters with multistage noise shaping modulators," IEEE Trans. Acoust., Speech, Signal Processing, vol.36, Dec. 1988, pp. 1899-1905

[30] M. L. Liu, Y. L. Kuo and C. F. Lee, "A tutorial on computer aided analysis of switched capacitor circuits," Proc. IEEE, Vol. 71, No.8, Aug. 1983, pp.987-1005

[31] S. C. Fang, Y. P. Tsividis and O. Wing, "SWITCAP, a switched capacitor network analysis program," IEEE Circuits and Systems Magazine, Dec. 1983, pp.4-9 and pp.42-46.

[32] "Users's guide for SWITCAP (version 5)," Columbia University, New York, Aug. 1987

[33] L. B. Wolovitz and J. I. Sewell, "General analysis of large linear switched capacitor networks," IEE Proc. Vol.135, pt.G, No.3, Jun. 1988, pp.119-124

[34] J. Rabaey, J. Vandewalle, H. De Man, "On frequency domain analysis of switched capacitor networks including all parasitics," Proc. of IEEE ISCAS, Chicago, Apr. 1981, pp.868-871

[35] J. Vandewalle, H. De Man and J. Rabaey, "Time, frequency and z-domain modified nodal analysis of switched capacitor networks," IEEE Trans. CAS, Vol. CAS-28, No.3, Mar. 1981, pp.186-195

[36] "SWAP 2.2 Tutorial," Laboratorium ESAT, Katholieke Universiteit Leuven, Belgium, Sept. 1984

[37] A. Opal and J. Vlach," Analysis and sensitivity of periodically switched linear-networks," IEEE Trans. Circuits and Systems, vol.36, no.4, 1989, pp.522-532

[38] R. Soin, private communication, Cadence Design System Ltd., 1994

[39] J. B. Hughes, private communication, Philips Semiconductor Ltd., 1994

[40] R. S. Martin and J. H. Wilkinson, "Similarity reduction of a general matrix to Hessenburg form," Numer. Math., Vol. 12, 1968, pp. 349-368

CHAPTER 2

SENSITIVITY ANALYSIS

2.1 INTRODUCTION

The sensitivity analysis of ideal switched capacitor networks is well established and forms part of the designer's standard tool-kit [1-5]. The present trend to extend the application of switched capacitor networks to high frequency front-ends with multi-rate clocks raises serious questions of sensitivity of the circuit structures with respect to nonideal parameters such as switch resistance and finite amplifier parameters-gain/bandwidth, input and output impedances; sensitivity of parasitic capacitance is also required. More recently, the switched current (SI) signal processing technique has received considerable attention as it can be implemented in a standard digital CMOS fabrication process. Substantial research has been undertaken to develop various generations of current memory cell to improve sensitivity behaviour [6-8]. Hence efficient sensitivity analysis is crucial to these challenges.

A number of programs can undertake various frequency and time domain analysis of nonideal switched linear networks [9-12]. A range of techniques have been utilised to increase speed, improve accuracy and cope with large networks with multiple clock waveforms [12]. However an efficient sensitivity analysis of general nonideal switched linear networks is conspicuously absent.

In this Chapter, computational methods for sensitivity analysis will be examined. Among them, the adjoint network technique seems to be the best solution in terms of both accuracy and efficiency for computer application. Unfortunately, it cannot be applied directly to general nonideal switched linear networks, the difficulties are discussed in detail. A new method, related to the numerical inverse Laplace transform technique, is proposed. The computer implementation algorithm of the proposed method is illustrated.

In the remainder of the Chapter, several important applications of sensitivity analysis are demonstrated. An effective strategy for gain and bandwidth product sensitivity calculation of the opamp is proposed. Definitions for switch resistance sensitivity are also given. Finally, the derivation of group delay evaluation and the sum sensitivities for networks are presented.

2.2 SENSITIVITY ANALYSIS METHODS

2.2.1 Definitions

(a) The magnitude and phase sensitivities

The magnitude and phase sensitivities of the network are usually of prime interest to designers. Let the network function be written in polar form as $H = |H|e^{j\varphi}$, where $|H|$ is the magnitude and φ is the phase function. Taking the natural logarithm and differentiating with respect to an arbitrary network parameter μ gives,

$$\frac{1}{H}\frac{\partial H}{\partial \mu} = \frac{1}{|H|}\frac{\partial |H|}{\partial \mu} + j\frac{\partial \varphi}{\partial \mu} \tag{2.1}$$

From the above complex equation, the normalised magnitude sensitivity can be defined as

$$S_{\mu}^{|H|} \equiv \frac{\mu}{|H|}\frac{\partial |H|}{\partial \mu} = Re\left(\frac{\mu}{H}\frac{\partial H}{\partial \mu}\right) \tag{2.2}$$

The phase sensitivity is defined as

$$S_{\mu}^{\varphi} \equiv \mu\frac{\partial \varphi}{\partial \mu} = Im\left(\frac{\mu}{H}\frac{\partial H}{\partial \mu}\right) \tag{2.3}$$

where Re and Im stand for the real and imaginary parts, respectively.

Sensitivities are mathematical measures of the magnitude or phase variation due to the relative deviation of the parameter. Since the magnitude response of the network is normally measured in decibels, the magnitude sensitivity is converted to dB by multiplying a constant $20\log_{10}e$, the unit then becomes dB per percent parameter variation. The phase sensitivity is measured in degrees per percent parameter deviation by multiplying a constant $180/\pi$.

(b) Group delay and magnitude slope

If the parameter μ is chosen to be the frequency, another two important sensitivities are introduced.

The group delay, which is of great importance in filter design, is defined as the negative of the derivative of the phase

$$\tau(\omega) \equiv -\frac{\partial \varphi}{\partial \omega} = -\mathrm{Im}\left(\frac{1}{H}\frac{\partial H}{\partial \omega}\right) \tag{2.4}$$

The slope of the magnitude frequency characteristic, which is often of interest in stability analysis, is given by

$$\kappa(\omega) \equiv \frac{\partial}{\partial \omega}(\ln|H|) = \mathrm{Re}\left(\frac{1}{H}\frac{\partial H}{\partial \omega}\right) \tag{2.5}$$

(c) Group delay sensitivity

Group delay sensitivity with respect to an arbitrary parameter μ is obtained by differentiating (2.4)

$$S_{\mu}^{\tau(\omega)} \equiv \frac{\mu}{\tau}\frac{\partial \tau}{\partial \mu} \tag{2.6}$$

When comparing different prototype realisations during the filter design process, group delay sensitivity is a useful criteria which helps designers to choose better structures for their design.

2.2.2 Computational techniques

For computer application, consider a system of linear equations in the form

$$MX = W \tag{2.7}$$

where M represents the system matrix, X denotes the unknown vector and W is the excitation vector. Equation (2.7) will be used frequently throughout the chapter.

(a) Perturbation approach

This approach is to perturb each circuit element or parameter by $\Delta\mu$ and re-analyse the network. Comparison of the output for the perturbed and unperturbed cases to give ΔX could possibly indicate the unnormalised sensitivity by using the relation $\partial X/\partial\mu \approx \Delta X/\Delta\mu$. This technique is simple and straightforward. When the network function cannot be written as an explicit function in terms of the network parameters, the perturbation approach might be a useful alternative to estimate the network sensitivities. However, it requires that a small change be made in each parameter and the network re-analysed. This method requires one solution of (2.7) for each

perturbed parameter in addition to the solution of (2.7) with original parameters, and is highly prone to numerical inaccuracies.

(b) Sensitivity network approach

Differentiate (2.7) with respect to μ,

$$M\frac{\partial X}{\partial \mu}+\frac{\partial M}{\partial \mu}X=\frac{\partial W}{\partial \mu} \tag{2.8}$$

and rewrite as follows

$$M\frac{\partial X}{\partial \mu}=-\left(\frac{\partial M}{\partial \mu}X-\frac{\partial W}{\partial \mu}\right) \tag{2.9}$$

The solution of (2.9) generates the sensitivity of the whole vector X with respect to a single variable μ. If sensitivity with respect to many parameters is required, (2.9) should be solved for each parameter in turn. However, it can be noticed that the only difference between (2.7) and (2.9) is the right hand side. Therefore, after the right hand side of (2.9) was formed, the LU factors of (2.7) can be utilised again with merely the additional effort of forward and backward substitutions to get the final solutions. Although the sensitivity can be accurately evaluated, computationally this approach is still not the best solution to sensitivity analysis problem. In addition, it ignores the fact that the sensitivity of all components of X is seldom required.

(c) Adjoint network techniques

Let the output of interest ϕ, be a linear combination of the components of X,

$$\phi = d^t X \tag{2.10}$$

where d is a constant vector and d^t denotes the transpose.

The sensitivity of ϕ with respect to the parameter μ is

$$\frac{\partial \phi}{\partial \mu} = d^t \frac{\partial X}{\partial \mu} \tag{2.11}$$

and the solution of (2.9) is

$$\frac{\partial X}{\partial \mu} = -M^{-1}\left[\frac{\partial M}{\partial \mu}X - \frac{\partial W}{\partial \mu}\right] \tag{2.12}$$

substitute for $\partial X/\partial \mu$ from (2.12) into (2.11), gives

$$\frac{\partial \phi}{\partial \mu} = -d^t M^{-1}\left[\frac{\partial M}{\partial \mu}X - \frac{\partial W}{\partial \mu}\right] \tag{2.13}$$

Define the adjoint vector X_a by

$$X_a^t = -d^t M^{-1} \tag{2.14a}$$

or $$M^t X_a = -d \tag{2.14b}$$

and substitute in (2.13), which gives the final result

$$\frac{\partial \phi}{\partial \mu} = X_a^t \frac{\partial M}{\partial \mu}X - X_a^t \frac{\partial W}{\partial \mu} \tag{2.15}$$

By utilising the equation (2.15), sensitivity of any output variables with respect to any parameter of interest can be calculated. The advantage of this method is that the vectors X and X_a are only calculated once, irrespective of the number of parameters. Apparently only two solutions for (2.14) and (2.7) are required. The transpose solution X_a can be efficiently obtained using similar steps to those used in solving for X, and does not require another network analysis.

The computation procedure for the adjoint approach can be summarised:

Step 1: Solve (2.7) for X;
Step 2: Calculate X_a from (2.14b);
Step 3: Formulate $\partial M/\partial \mu$ and $\partial W/\partial \mu$;
Step 4: Calculate $\partial \phi/\partial \mu$ by (2.15);

This technique has been widely adopted in many circuit analysis software packages and proven to be both efficient and accurate. Unfortunately, it is not amenable to direct application to nonideal switched linear systems since the system matrix is not expressed explicitly in terms of network parameters and numerically estimated matrices are involved. In next section, the problem will be discussed in detail.

2.2.3 Special problem of nonideal switched linear networks

For nonideal switched linear networks, the system matrix equation in z domain can be written in the form [12]

$$\underbrace{\begin{bmatrix} I & & & -P_1 \\ -P_2 & I & & \\ & \ddots & \ddots & \\ & & -P_N & zI \end{bmatrix}}_{M} \underbrace{\begin{bmatrix} V_1(z) \\ V_2(z) \\ \vdots \\ V_N(z) \end{bmatrix}}_{X} = \underbrace{\begin{bmatrix} W_1(z) \\ W_2(z) \\ \vdots \\ zW_N(z) \end{bmatrix}}_{W} \tag{2.16}$$

where

$$P_k = p_k C_k$$

and $$W_k = \sum_{n=0}^{\infty} [\alpha_k B_k] z^{-n}$$

p_k is so called the extended state transition matrix (EST) and defined as

$$p_k = \pounds^{-1}\left\{\left[G_k + sC_k\right]^{-1}\right\} \tag{2.17}$$

B_k is called the excitation response matrix and obtained by

$$B_k = \pounds^{-1}\left\{s^{-1}\left[G_k + sC_k\right]^{-1}\right\} \tag{2.18}$$

$\pounds^{-1}$ denotes the inverse Laplace transformation; α_k are coefficients of the polynomial approximations of the excitation; G_k and C_k are conductance and capacitance matrices, respectively.

The physical explanation of the extended state transition matrix can be inferred from its definition which relates the final nodal voltages and branch currents at the end of a particular time slot to the initial values of that slot under zero excitations. It has been found that the determination of the extended state transition matrix is equivalent to computing the matrix exponential which has no analytical solution but only numerical approximation with various degrees of accuracy. This implies that after numerical approximation in a pre-processing stage, the extended state transition matrix is entirely numerical and no information of network parameters is retained. Therefore, $\partial M/\partial \mu$ is not available for adjoint network approach described in previous section, neither is $\partial W/\partial \mu$.

2.2.4 Generation of derivatives

As discussed in last section, p_k and B_k can only be produced by numerical techniques, so the major problem of sensitivity analysis remains to determine the partial derivative of p_k and B_k with respect to μ. Since the method developed in this section highly depends on having a reliable, efficient and accurate technique for computing the extended state transition matrix p_k and the excitation response matrix B_k, a brief review of methods for computing a matrix exponential is given first.

(a) Computing the inverse Laplace transform

It has already been concluded [13] that only few methods are able to compute the matrix exponential, especially when singular matrices exist. Among them, the Padé-approximation [14] is suggested as a reliable one. However it is not chosen in this application due to the high computational costs for the desired accuracy [35].

An alternative method was first proposed in [15] by using I_{mn} approximant to compute the matrix exponential and found to be both reliable and efficient. The approach used here is based on the derivation in [16], which uses a numerical approximation of the Laplace transform inversion integral

$$v(t) = \frac{1}{2\pi j}\int_{c-\infty}^{c+\infty} V(s)e^{st}ds \tag{2.19}$$

The quadrature approximation is derived in [16] as

$$\hat{v}(t) = \frac{1}{t}\sum_{i=1}^{m} K_i V(z_i/t) \tag{2.20}$$

where K_i and z_i are tabulated complex constants. m is the approximation order. The I_{mn} constants for m=1,...,10 are tabulated in [17]. The computation of (2.20) can be further reduced when m is even, due to the occurrence of complex conjugate pairs. Therefore

$$\hat{v}(t) = \frac{1}{t}\sum_{i=1}^{m/2} Re\left[K_i V(z_i/t)\right] \tag{2.21}$$

The extended state transition matrix can then be approximated by

$$p_k \approx \sum_{i=1}^{m/2} Re\left\{\frac{2K_i}{t}\left[\frac{z_i}{t}C_k + G_k\right]^{-1}\right\} \tag{2.22}$$

Similarly, the excitation response matrix is calculated according to

$$B_k \approx \sum_{i=1}^{m/2} Re\left\{\frac{2K_i}{z_i}\left[\frac{z_i}{t}C_k + G_k\right]^{-1}\right\} \tag{2.23}$$

(b) Derivatives of p_k and B_k

Having a reliable, and more importantly, an explicit approximation equation, the problem can be changed to the same task as calculating the extended state transition matrix. The principle is that the partial derivative of p_k needs to be obtained before the system matrix M is created numerically.

Differentiate (2.17) with respect to μ

$$\frac{\partial p_k}{\partial \mu} = \frac{\partial}{\partial \mu}\left\{\pounds^{-1}\left[G_k + sC_k\right]^{-1}\right\} \tag{2.24}$$

According to the definition of the inverse Laplace transform, rewrite (2.24) as

$$\frac{\partial p_k}{\partial \mu} = \frac{\partial}{\partial \mu}\left\{\frac{1}{2\pi j}\int_{c-\infty}^{c+\infty}\left[G_k + sC_k\right]^{-1} e^{st} ds\right\} \tag{2.25}$$

Swap the sequence of integration and differentiation

$$\frac{\partial p_k}{\partial \mu} = \frac{1}{2\pi j}\int_{c-\infty}^{c+\infty}\left\{\frac{\partial}{\partial \mu}\left[G_k + sC_k\right]^{-1}\right\} e^{st} ds \tag{2.26}$$

referring to the definition of p_k, rewrite (2.26) as

$$\frac{\partial p_k}{\partial \mu} = \pounds^{-1}\left\{\frac{\partial}{\partial \mu}\left[G_k + sC_k\right]^{-1}\right\} \tag{2.27}$$

By examining (2.27) and (2.17), it is evident that they both require computation of the inverse Laplace transform. It now remains to derive the expression of the partial derivative of the matrix inverse with respect to μ.

Define the sub-system matrix in time-slot k as

$$M_k = G_k + sC_k \tag{2.28}$$

Because

$$M_k M_k^{-1} = I \tag{2.29}$$

Differentiate (2.29) with respect to μ to obtain

$$\frac{\partial M_k}{\partial \mu} M_k^{-1} + M_k \frac{\partial M_k^{-1}}{\partial \mu} = 0 \tag{2.30}$$

and rewrite (2.30) as follows

$$\frac{\partial M_k^{-1}}{\partial \mu} = -M_k^{-1} \frac{\partial M_k}{\partial \mu} M_k^{-1} \tag{2.31}$$

Substitute (2.31) into (2.27) to give

$$\frac{\partial p_k}{\partial \mu} = \pounds^{-1}\left\{-M_k^{-1} \frac{\partial M_k}{\partial \mu} M_k^{-1}\right\} \tag{2.32}$$

Similarly

$$\frac{\partial B_k}{\partial \mu} = \pounds^{-1}\left\{\frac{1}{s}\left[-M_k^{-1} \frac{\partial M_k}{\partial \mu} M_k^{-1}\right]\right\} \tag{2.33}$$

The inverse system matrix M_k^{-1} and the partial derivative $\partial M_k / \partial \mu$ can be calculated easily. Therefore, applying I_{nm} approximation method to (2.32) and (2.33) gives

$$\frac{\partial p_k}{\partial \mu} \approx \sum_{i=1}^{m/2} \mathrm{Re}\left\{\frac{2K_i}{t}\left[-M_k^{-1} \frac{\partial M_k}{\partial \mu} M_k^{-1}\right]\right\} \tag{2.34}$$

$$\frac{\partial B_k}{\partial \mu} \approx \sum_{i=1}^{m/2} \mathrm{Re}\left\{\frac{2K_i}{z_i}\left[-M_k^{-1} \frac{\partial M_k}{\partial \mu} M_k^{-1}\right]\right\} \tag{2.35}$$

Hence the computation of $\partial M/\partial \mu$ and $\partial W/\partial \mu$ follow in a straightforward manner and the general procedure for nonideal sensitivity analysis can be accomplished. Salient features of the method are summarised as:

(1) It is completely general. The analysis is applicable to general switched linear networks. No restrictions are placed on the type of signal, circuit configuration and

the number of clock phases. Therefore the method is suited for mixed-mode analysis applications.

(2) The method is also very efficient. After pre-processing of all frequency independent components in the equations, the sensitivity analysis procedure follows a similar pattern used for ideal switched capacitor network analysis.

(3) The results of the method are reliable and accurate. The only approximation employed in this method is I_{mn} approximation which experience has shown to be stable and accurate over wide range of systems applications.

2.2.5 Computer implementation

The above approach was implemented in the program SCNAP4 [18-19]. The algorithm for sensitivity analysis can be described as follows,

/* perform I_{mn} approximation */

for (k=1; k < N; k++) {

for (i = 1; i < m/2; i++) {

$$p_k = p_k + \mathrm{Re}\left\{\frac{2K_i}{t}\left[\frac{z_i}{t}C_k + G_k\right]^{-1}\right\};$$

$$B_k = B_k + \mathrm{Re}\left\{\frac{2K_i}{z_i}\left[\frac{z_i}{t}C_k + G_k\right]^{-1}\right\};$$

$$\frac{\partial p_k}{\partial \mu} = \frac{\partial p_k}{\partial \mu} + Re\left\{\frac{2K_i}{t}\left[-M_k^{-1}\frac{\partial M_k}{\partial \mu}M_k^{-1}\right]\right\};$$

$$\frac{\partial B_k}{\partial \mu} = \frac{\partial B_k}{\partial \mu} + Re\left\{\frac{2K_i}{z_i}\left[-M_k^{-1}\frac{\partial M_k}{\partial \mu}M_k^{-1}\right]\right\};$$

}

$$P_k = p_k C_k;$$

$$\frac{\partial P_k}{\partial \mu} = \frac{\partial p_k}{\partial \mu}C_k + p_k\frac{\partial C_k}{\partial \mu};$$

}

/* frequency loop */

for (freq = fstart; freq <= fstop; freq = freq + fstep) {

$X = M^{-1}W;$

$X_a^t = -\tilde{d}^t M^{-1};$

$\phi = \tilde{d}^t X;$

$$\frac{\partial \phi}{\partial \mu} = X_a^t \frac{\partial M}{\partial \mu} X - X_a^t \frac{\partial W}{\partial \mu};$$

$$S_\mu^\phi = \frac{\mu}{\phi} \frac{\partial \phi}{\partial \mu};$$

$$S_\mu^\varphi = \mu \frac{\partial \phi}{\partial \mu};$$

}

Steps in italics indicate that they are specially designed for sensitivity analysis. As shown in the algorithms, these steps conveniently fit into the original frequency analysis schemes. As discussed in section (2.2.4), the proposed sensitivity analysis method offers the opportunity to share the same effective I_{nm} numerical approximation techniques with frequency analysis. Other common techniques are the sparse matrix routines, the Hessenburg approach and interpretable code generation, all can be utilised in this procedure. Hence maximum efficiency of the methods can be expected. The sensitivity analysis statistics illustrated in the Table (2.1) indicates the efficiency of the proposed method and corresponding computer implementation.

It is not possible to compare run time efficiencies with other switched-network analysis software since no other package produces nonideal sensitivities available. But since an independent survey [20] of the direct frequency response computations evaluated a number of different packages and showed SCNAP4 to be two orders of magnitude faster, in some instances, it is reasonable to assume that such efficiency gains would be apparent in general nonideal sensitivity computations if facilities were available.

Table (2.1) Statistics of sensitivity analysis

circuit name	circuit size	clock phases	number of parameters	per-frequency point (sec)
bp2[21]	14	2	6	0.0032
but3[32]	10	3	5	0.0024
cbp6[22]	35	2	4	0.0138
elp15[23]	87	2	3	0.0683
elp3[24]	13	12	6	0.01715
elp6[25]	35	2	7	0.018
fet7[24]	35	6	3	0.0356
lp15[23]	87	2	2	0.0592
lp5[22]	28	2	3	0.0075
nos11[26]	47	4	2	0.0364
nos5[26]	23	4	3	0.0113
spft[25]	99	36	1	0.886

Note: above data obtained on SUN-sparc10 station

The accuracy of the proposed sensitivity analysis method is also guaranteed. The only approximation employed in the method is the I_{mn} approximation, which has been successfully utilised in the application of frequency analysis and proven to be stable and accurate over wide range of circuit examples. As has been stated above, no analytical basis is available against which the results obtained from SCNAP4 could be compared. Therefore the perturbation method was utilised to verify the sensitivity analysis results of some selected circuit examples. The results of the two methods match fairly well when the circuit parameter of interest was perturbed by a small variation.

2.3 APPLICATIONS OF SENSITIVITY ANALYSIS

A series of sensitivity application examples is now given to illustrate the power and flexibility of the proposed methods.

A typical example of a second order bandpass four phase switched capacitor filter is used and given in Fig. (2.1) [21]

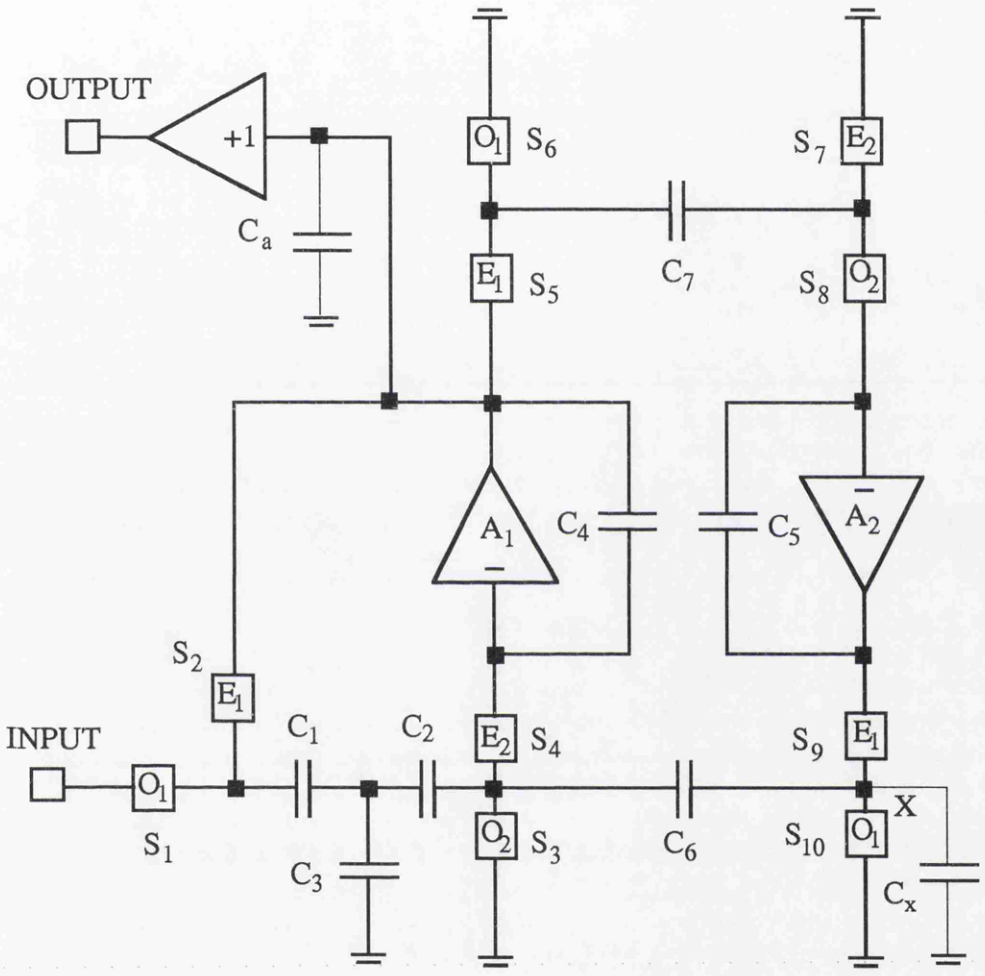


Fig. (2.1) Second order bandpass switched capacitor filter

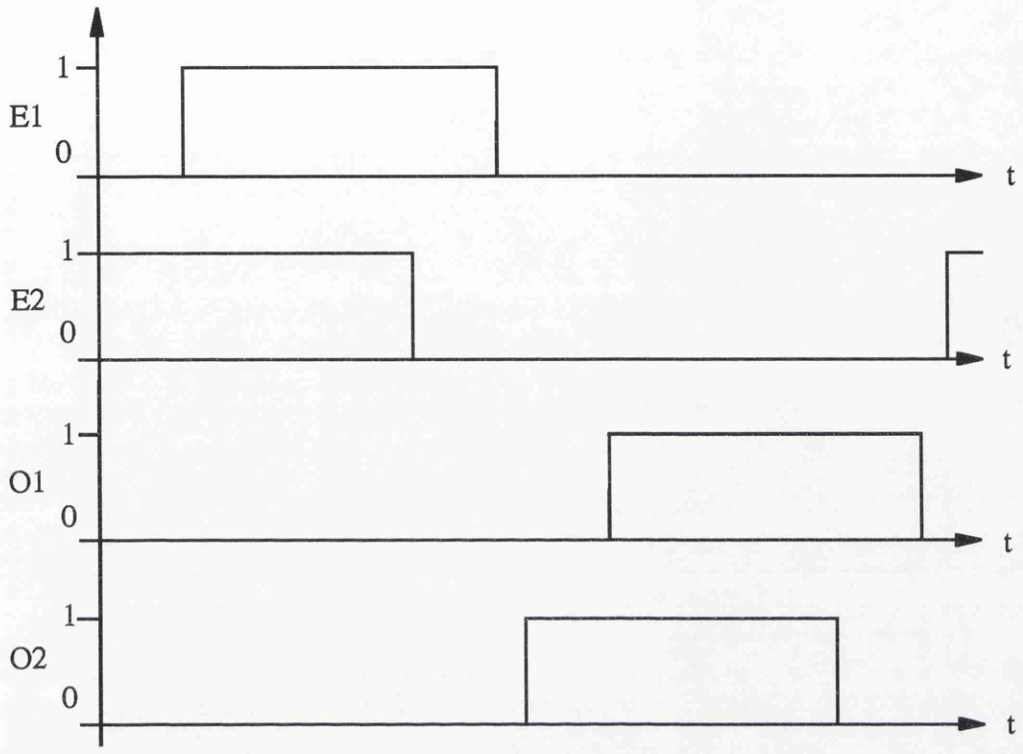


Fig. (2.2) Clock waveforms for second order bandpass SC filter

```
second-order sc bandpass filter fc=400kHz

s1 1 2 o1
s2 2 6 e1
s3 4 0 o2
s4 4 5 e2
s5 6 7 e1
s6 7 0 o1
s7 8 0 e2
s8 8 9 o2
s9 10 11 e1
s10 11 0 o1

c1 2 3 0.1pf
c2 3 4 0.1pf
c3 3 0 0.70206pf
c4 5 6 0.70206pf
c5 9 10 0.70206pf
c6 4 11 0.17668pf
c7 7 8 0.17668pf
cx 11 0 0pf

.subckt opamp 1 2 6 (gain,unit)
* gain = 10k, gb = 50M, bw = 5k
 ra 1 0 500Meg
 rb 2 0 500Meg
 rin 1 2 1Meg
 e1 1 2 3 0 gain
 g2 4 0 5 0 unit
* GB=50Meg
 r1 3 4 3.183098861837906e-5
 c1 4 0 1.0
 rout 5 0 100
.ends opamp

x1 0 9 10 opamp (10k,1)
x2 0 5 6  opamp (10k,1)

vin 1 0 ac 1 0

.phase e1 pwl 0ns 0v 10ns 1v 47.5ns 0v 100ns 0v
.phase e2 pwl 0ns 1v 37.5ns 0v 100ns 0v
.phase o1 pwl 0ns 0v 60ns 1v 97.5ns 0v 100ns 0v
.phase o2 pwl 0ns 0v 50ns 1v 87.5ns 0v 100ns 0v

.option groupd tcsens tcdsen
.option ron = 7.15k

.freq 360kHz 440kHz lin 100
.sens name= e1_x1, s4, c1, c4, c6, rin_x1, rout_x1, cx
.plot ac vdb(6)
.end
```

Fig. (2.3) SCNAP4 data file for second order bandpass SC filter

2.3.1. Linear nonidealities of opamp

In SC circuits, a number of linear nonidealities of the opamps are relevant, such as finite gain bandwidth product and input and output impedances.

(a) Finite gain and bandwidth product

It has long been recognised that the GB product of opamp is an important factor which restricts SC circuits for high frequency application [27-29]. Unfortunately, to study the GB product effects, the only feasible way was to analyse the network repeatedly with various GB products. However, this requires extensive computation costs. An effective method is now presented for evaluating the GB product sensitivity of the network. To facilitate description, a simple single-pole macromodel of opamp shown in Fig. (2.4) has been used.

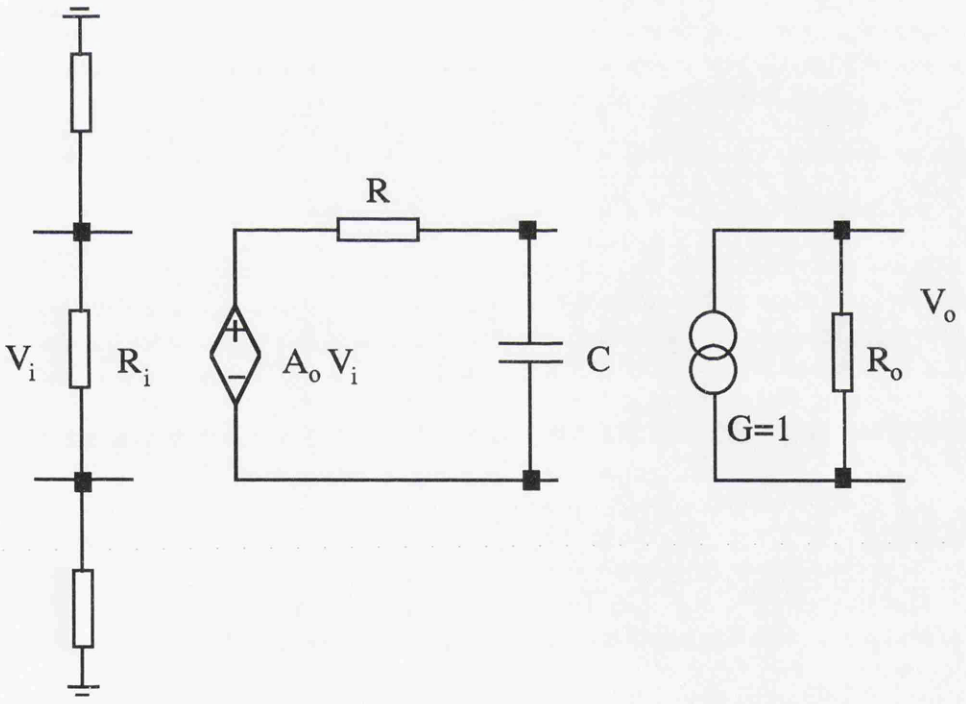


Fig. (2.4) macromodel of opamp

The transfer function of this macromodel is

$$H(f) = \frac{A_0}{1 + j\frac{f}{BW}} \tag{2.36}$$

where $BW = \dfrac{1}{2\pi RC}$

A_0 is the DC gain of opamp, BW denotes the bandwidth and f represents the frequency in Hz. Obviously the gain-bandwidth product can be written as

$$GB = A_0\, BW = \frac{A_0}{2\pi RC} \tag{2.37}$$

Notice that, GB is a function of a number of circuit parameters and to determine its sensitivity, some simple derivations need to be done. Examining (2.37) reveals that it provides a direct link between GB product and other practical network parameters (R, C and A_0 in this case), thus mathematically there are alternative solutions. According to the definition, the sensitivity of the system response with respect to GB can be written as

$$S_{GB}^{\phi} \equiv \frac{GB}{\phi}\frac{\partial \phi}{\partial GB} \tag{2.38}$$

Notice that for $\partial\phi/\partial GB$ the following relation exists

$$\frac{\partial \phi}{\partial GB} = \frac{\partial \phi}{\partial A_0}\frac{\partial A_0}{\partial GB} = \frac{\partial \phi}{\partial R}\frac{\partial R}{\partial GB} = \frac{\partial \phi}{\partial C}\frac{\partial C}{\partial GB} \tag{2.39}$$

If (2.37) and (2.39) are substituted into (2.38), then the result is

$$S_{GB}^{\phi} = \frac{A_0}{\phi}\frac{\partial \phi}{\partial A_0} \tag{2.40a}$$

or $$S_{GB}^{\phi} = -\frac{R}{\phi}\frac{\partial \phi}{\partial R} \tag{2.40b}$$

or $$S_{GB}^{\phi} = -\frac{C}{\phi}\frac{\partial \phi}{\partial C} \tag{2.40c}$$

Hence, the computation of GB sensitivity is now equivalent to calculating the sensitivity with respect to any of these three circuit parameters. A similar strategy can be applied to more elaborate macromodels of the opamp.

Throughout the section, a few data lines for each specific simulation request will be shown to demonstrate the simplicity and flexibility of writing SCNAP4 data file for various simulation purposes. Detailed syntax of the data file can be found in [19]

```
.freq 360kHz 440kHz lin 100
.sens name = e1_x1, c1, c4, c6
```

For the second order bandpass SC filter, the magnitude and phase responses are shown in Fig. (2.5). Sensitivities of some normal circuit elements are displayed in Fig. (2.6). The sensitivity with respect to GB is shown in Fig.(2.7).

(b) Input and output impedance

By utilising the macromodel illustrated above, the influence of input or output impedance of opamp on the circuit performance can be easily investigated by evaluating their sensitivities.

```
.sens name = rin_x1, rout_x1
```

Fig.(2.8) shows the influence of input and output impedance on circuit response. The value of input and output impedance are 1Meg ohm and 100 ohm respectively.

2.3.2. Switch resistance

Although the effect of switch resistance is a very important factor which limits SC circuits for high frequency application, the formal sensitivity investigation of such a parameter was not feasible [30-32] before the full nonideal switched linear network analysis methods were proposed. A special feature of switch resistance is that its value changes with different clock phases. A modified definition of sensitivity to switch resistance, based on the fact that switch-on and switch-off resistances only give contributions in their own activated clock phases [33], is proposed,

$$S_{on}^{\phi} = \sum_{i=1}^{N} \frac{R_{on}}{\phi} \frac{\partial \phi}{\partial R_s} \tag{2.41a}$$

$$S_{off}^{\phi} = \sum_{i=1}^{N} \frac{R_{off}}{\phi} \frac{\partial \phi}{\partial R_s} \tag{2.41b}$$

where R_s represents switch resistances and R_{on}, R_{off} are the values of switch-on and switch-off resistance, respectively. N is the number of relevant time slots. The sensitivity of switch resistance is decomposed into two different contributions. During each clock period, switch-on resistances only have their influence when switches are closed. Similarly, switch-off resistances will give their contribution when switches are opened. Generally, the switch resistance has much less influence on circuit performance compared with other circuit elements, such as capacitance or the gain of

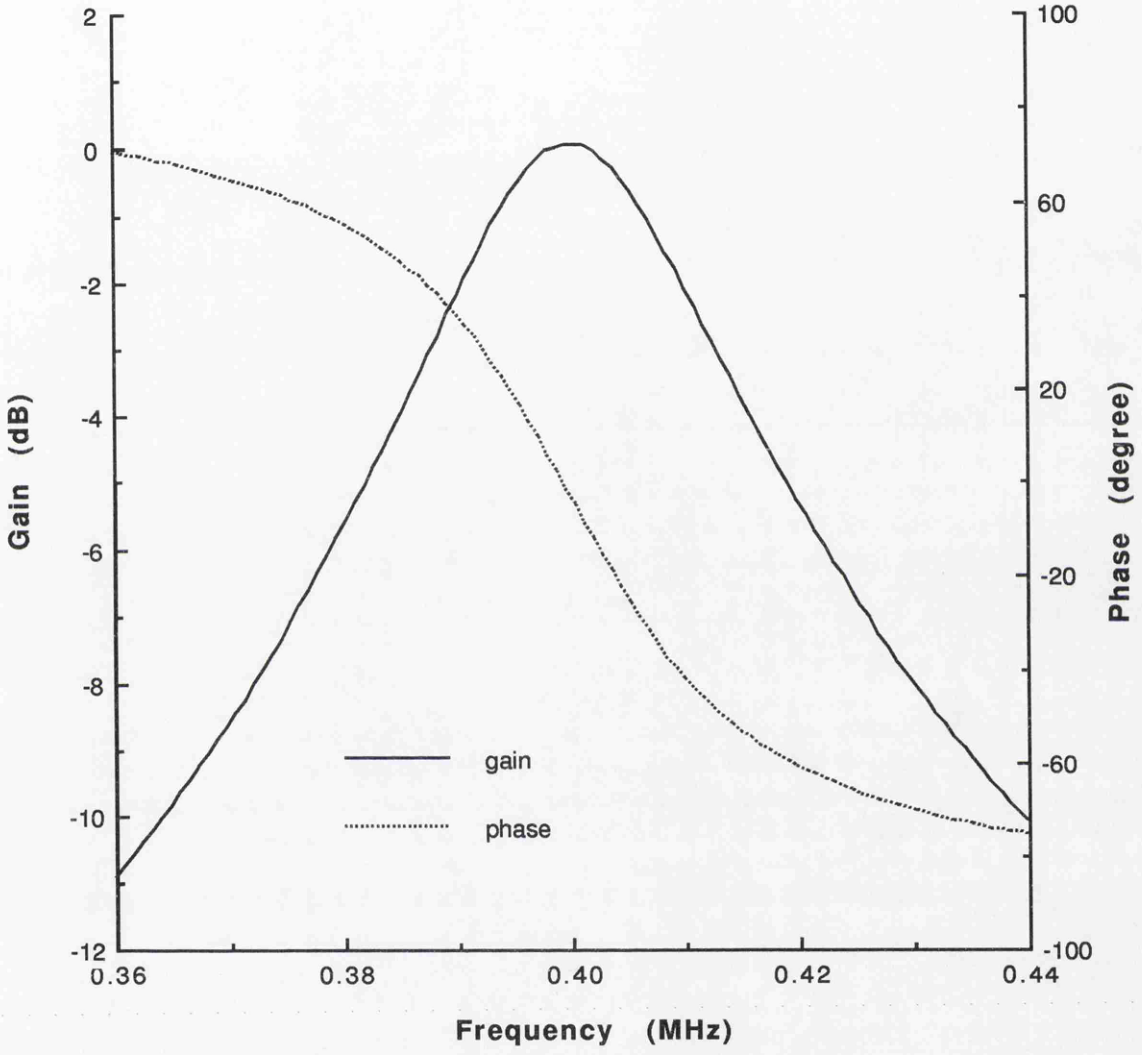


Fig. (2.5) Frequency response of 2nd order bandpass SCF

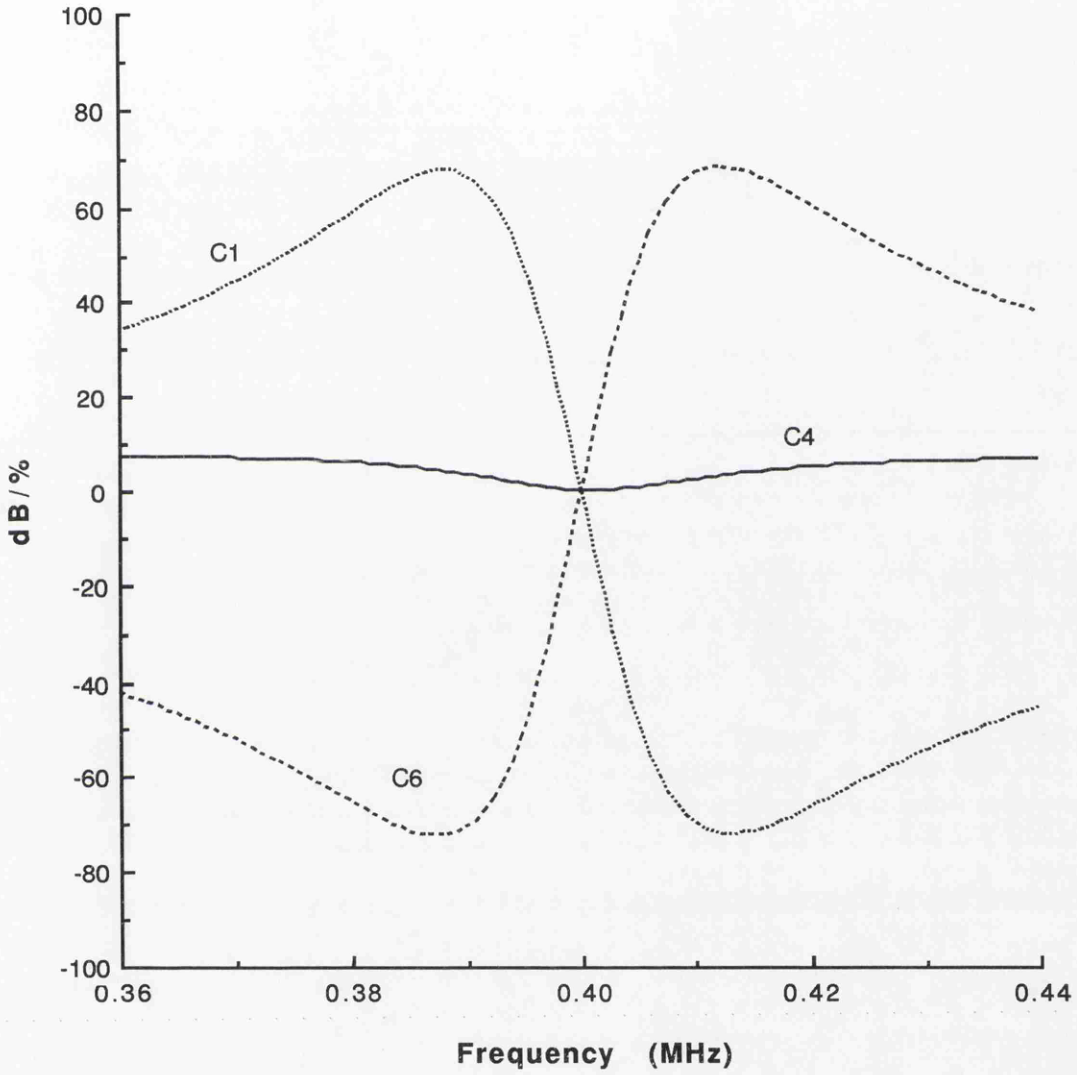


Fig. (2.6) Sensitivities w.r.t. C1, C4, C6

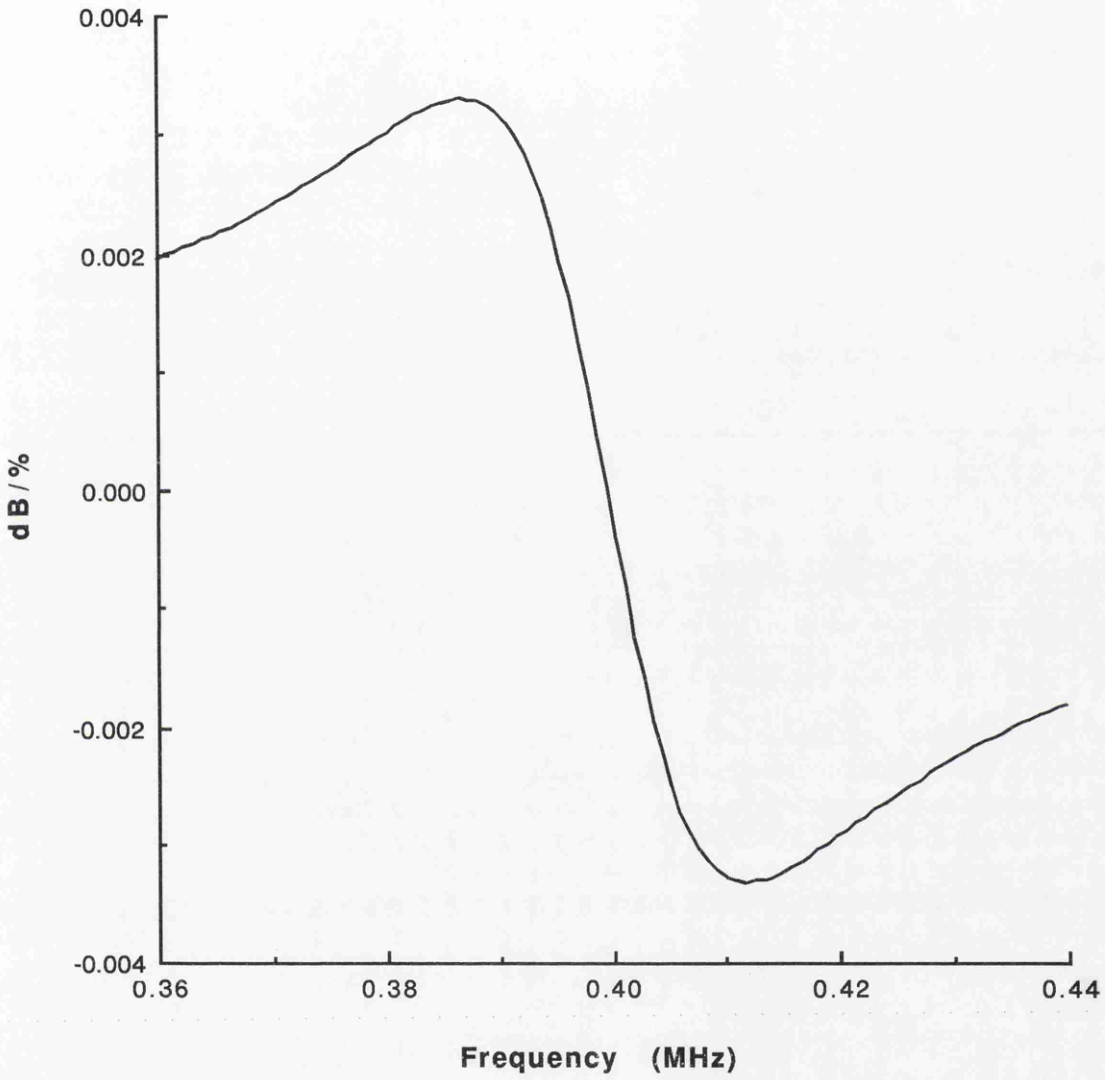


Fig. (2.7) Sensitivity with respect to GB

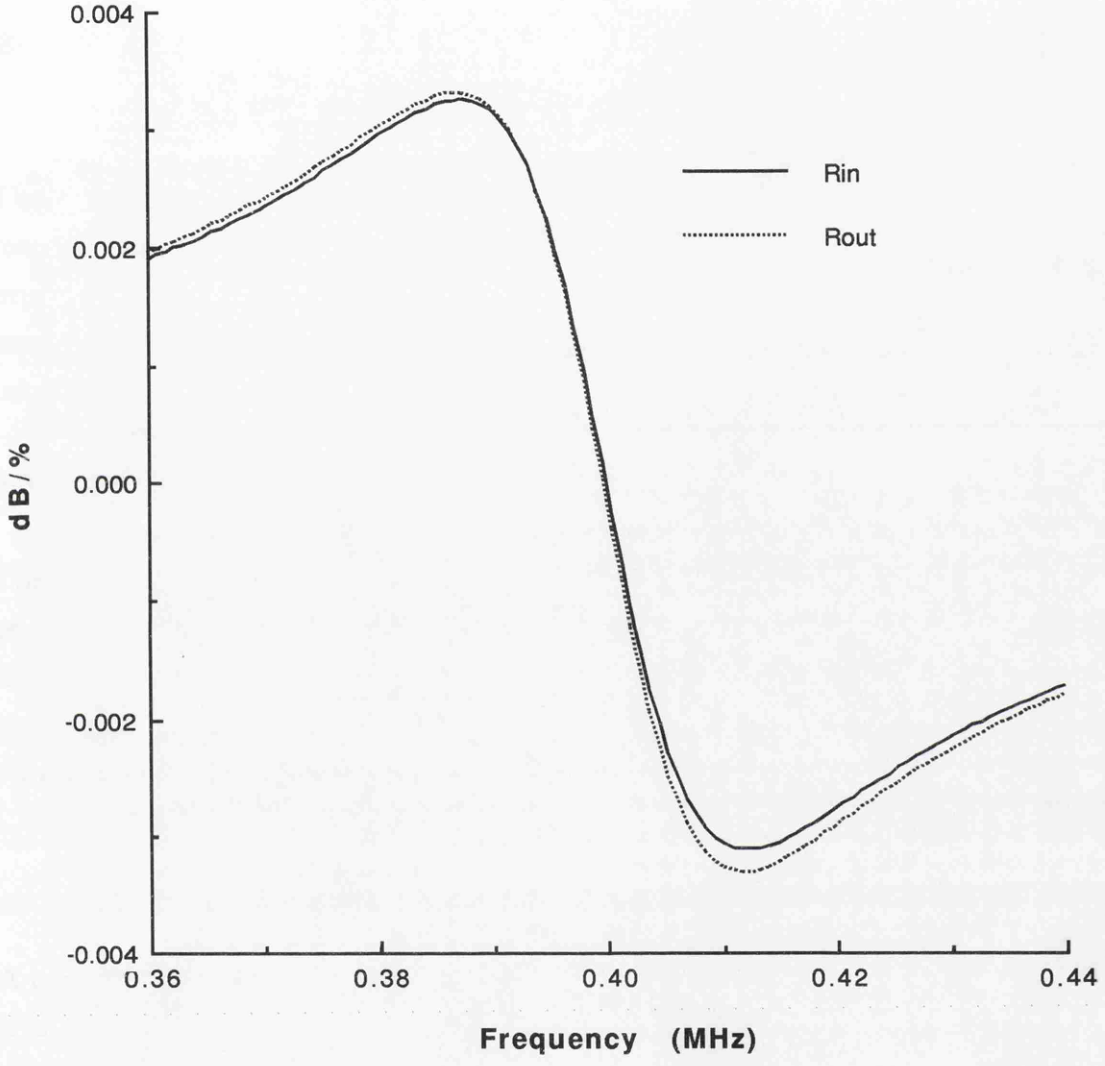


Fig. (2.8) Sensitivities w.r.t. Rin and Rout in subcircuit X1

opamps. However, for some high frequency SC applications, the improper value of switch resistance can totally distort the circuit response. Under these circumstances, sensitivity analysis shows that the circuit performance is very sensitive to switch resistance. This can be interpreted as the incomplete charge transfer in a clock phase due to significant time constants. Hence, the influence of switch resistance cannot be ignored and should be considered seriously.

```
.sens name = s4
```

The default switch-on and switch-off resistance are set to 7.15k and 100Meg ohm, respectively. From Fig. (2.9), it can be observed that the switch-on resistance has much greater influence than switch-off resistance for high frequency application. Its sensitivity magnitude is larger than that of GB product (see Fig.(2.7)) and should be taken into account during the filter design process.

2.3.3. Nodal parasitics

The nodal parasitics are usually identified as stray capacitance associated with some or all of the nodes in a circuit. They are presumed to have zero nominal value in the ideal situation. In practice, nodal parasitics do have some small values which might have a severe influence on network behaviour. By calculating sensitivity to nodal parasitics, their influence can be accurately predicted.

```
cx 11 0 0pf
.sens name = cx
```

A pseudo capacitor with zero value is inserted between node X and the ground. Since it is not a normalised sensitivity, the unit of the sensitivity of the nodal parasitic is given in dB per pico farad. The sensitivity of the nodal parasitic is shown Fig.(2.10).

2.3.4 Group delay

In modern digital communications and signal processing systems, the group delay should be considered to avoid inter-symbol interference. For minimum signal distortion, the group delay should be constant over the filter passband or within specified tolerance. Therefore, efficient computation of the group delay of a designed filter is an indispensable verification step to filter design.

For a system described by (2.16), taking the window function into account, the overall frequency response of the system is given by

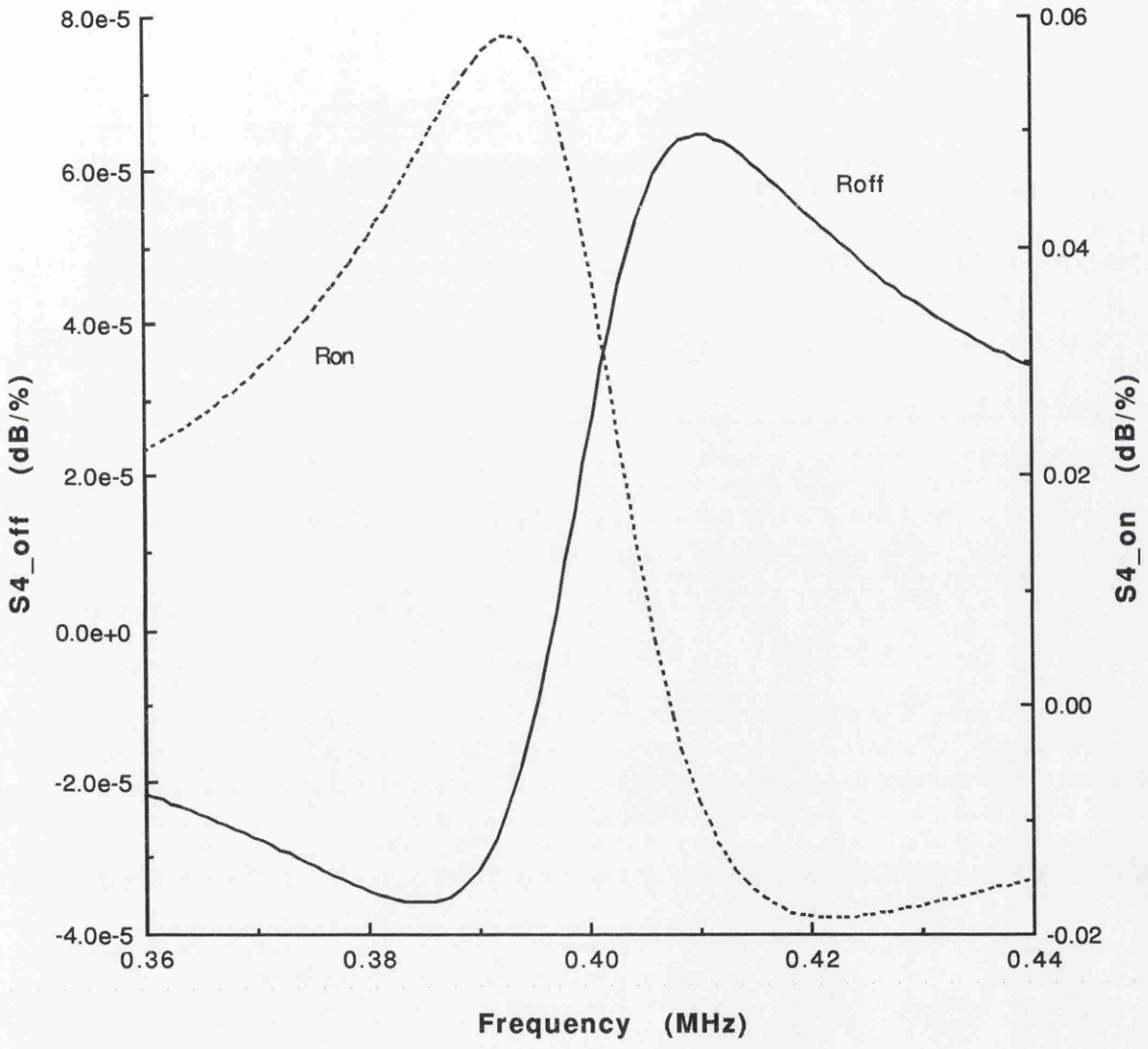


Fig. (2.9) Sensitivities w.r.t. switch resistances of S4

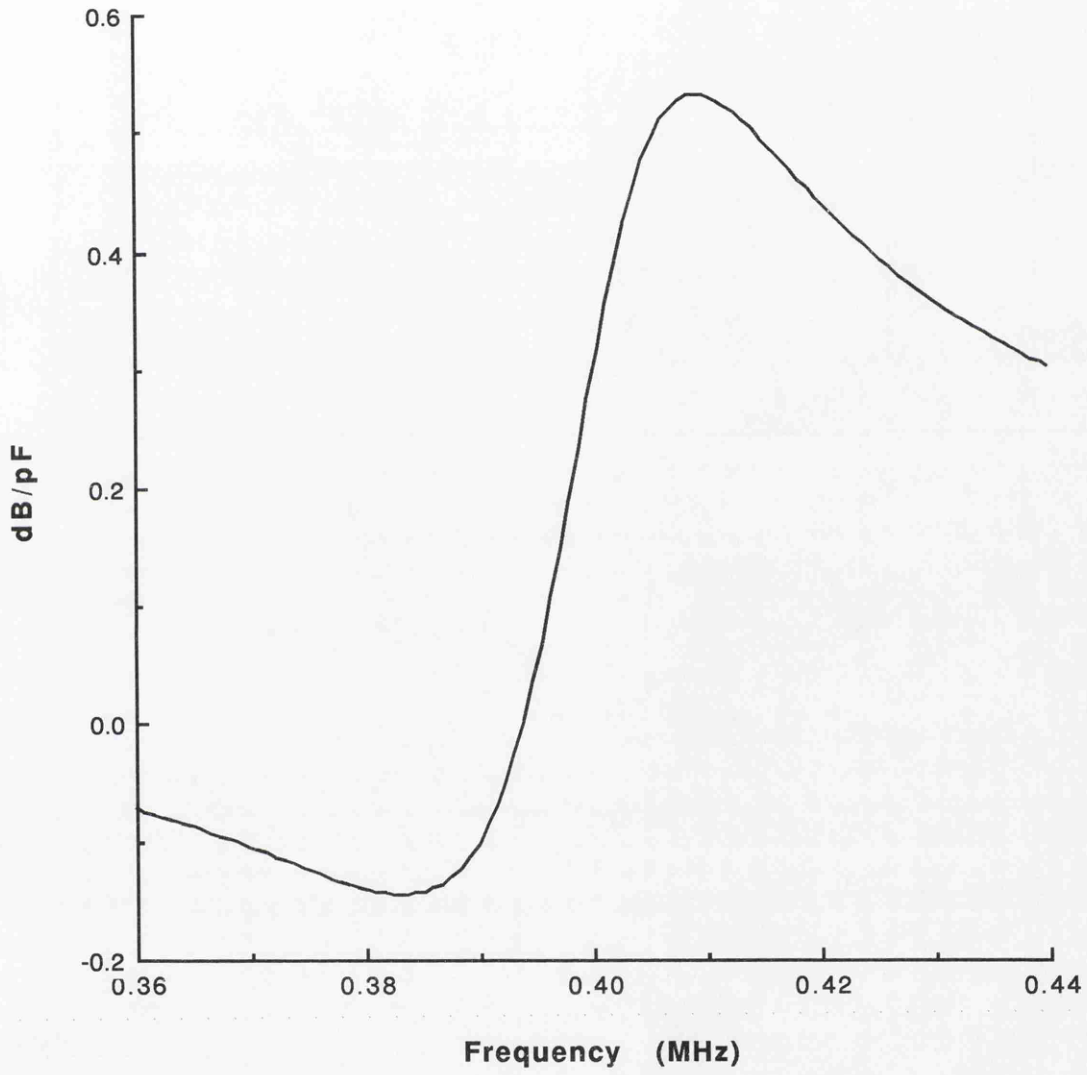


Fig. (2.10) Sensitivity w.r.t. stray capacitance of node X

$$\phi = \sum_{k=1}^{N} d^t D_k V_k \left(e^{j\omega_0 T} \right) \tag{2.42}$$

where

$$\begin{cases} D_k = \dfrac{e^{-j\omega\sigma_{k-1}} - e^{-j\omega\sigma_k}}{j\omega T} & k \neq N \\ D_N = e^{j\omega_0 T} \dfrac{e^{-j\omega\sigma_{N-1}} - e^{-j\omega\sigma_N}}{j\omega T} & k = N \end{cases} \tag{2.43}$$

Differentiate (2.42) with respect to frequency ω gives

$$\frac{\partial \phi}{\partial \omega} = \sum_{k=1}^{N} d^t \frac{\partial D_k}{\partial \omega} V_k + \sum_{k=1}^{N} d^t D_k \frac{\partial V_k}{\partial \omega} \tag{2.44}$$

Define a new vector as

$$\tilde{d} = \begin{pmatrix} D_1 \\ D_2 \\ \vdots \\ D_N \end{pmatrix} d \tag{2.45}$$

Substitute (2.45) into (2.44) to give

$$\frac{\partial \phi}{\partial \omega} = \sum_{k=1}^{N} d^t \frac{\partial D_k}{\partial \omega} V_k + \tilde{d}^t \frac{\partial X}{\partial \omega} \tag{2.46}$$

Replace the second term of (2.46) by referring to (2.15), results in

$$\frac{\partial \phi}{\partial \omega} = \sum_{k=1}^{N} d^t \frac{\partial D_k}{\partial \omega} V_k + X_a^t \frac{\partial M}{\partial \omega} X - X_a^t \frac{\partial W}{\partial \omega} \tag{2.47}$$

where

$$\frac{\partial D_k}{\partial \omega} = -\frac{D_k}{\omega} - \frac{\sigma_{k-1} e^{-j\omega\sigma_{k-1}} - \sigma_k e^{-j\omega\sigma_k}}{\omega T} \tag{2.48}$$

$$\frac{\partial D_N}{\partial \omega} = -\frac{D_N}{\omega} + \frac{e^{-j\omega T}}{\omega} \tag{2.49}$$

$$\frac{\partial M}{\partial \omega} = j\begin{bmatrix} 0 & & \\ & 0 & \\ & & \ddots \\ & & & zTI \end{bmatrix} \tag{2.50}$$

$$\frac{\partial W}{\partial \omega} = j\begin{bmatrix} \sigma_1 B_1 e^{j\omega\sigma_1} \\ \sigma_2 B_2 e^{j\omega\sigma_2} \\ \vdots \\ \sigma_N B_N e^{j\omega\sigma_N} \end{bmatrix} \tag{2.51}$$

The above theoretical derivation reveals the possibility of an efficient computer implementation for group delay calculation. Since p_k and B_k are frequency independent, $\partial p_k / \partial \omega$ and $\partial B_k / \partial \omega$ are equal to zero. Hence for group delay, the pre-processing is not required. In addition, (2.50) indicates that only one entry remains in the matrix so that no storage is required for $\partial M/\partial \omega$. The computer algorithm presented in section (2.2.5) can be easily updated to involve group delay evaluation with merely a few extra lines,

```
/* frequency loop */
for ( freq = fstart; freq <= fstop; freq = freq + fstep) {
    X = M^-1 W;
    X_a^t = -d~^t M^-1;
    φ = d~^t X;
    ∂φ/∂μ = X_a^t ∂M/∂μ X - X_a^t ∂W/∂μ;
    S_μ^φ = (μ/φ)(∂φ/∂μ);  S_μ^φ = μ ∂φ/∂μ;
    ∂φ/∂ω = Σ_{k=1}^{N} d^t ∂D_k/∂ω V_k + X_a^t ∂M/∂ω X - X_a^t ∂W/∂ω
    τ = -Im( (1/φ)(∂φ/∂ω) )
}
```

Notice that the computation of group delay does not need an extra system solution. It can be accomplished along with the normal sensitivity analysis.

```
.sens
.option groupd
```

In Fig. (2.11), the group delay of the second order bandpass switched capacitor filter is illustrated.

2.3.5 Sum sensitivity

Another useful application is to calculate the sum sensitivity of the network. The sum sensitivity can be defined as

$$S^{\phi}_{sum} = \sqrt{\left(S^{\phi}_{1}\right)^2 + \left(S^{\phi}_{2}\right)^2 + \cdots + \left(S^{\phi}_{\rho}\right)^2} \tag{2.52}$$

where S^{ϕ}_{i} represents the sensitivity with respect to the *i*th network parameter.

During the filter design process, it is difficult to generalise on which network should be chosen for a given problem since each prototype has different properties of dynamic range, major component spread and sensitivity to component errors. The sum sensitivity of the network is found particularly useful in the problem of network comparison. Since large or small component values are prone to fabrication errors and usually indicate poor network sensitivity. Therefore, comparing sensitivity can assist designers to choose the best network structure for their purposes.

```
.sens
.option tcsens tcdsen
```

Fig. (2.12) shows the sum of transfer function sensitivities with respect to all capacitances and the sum of group delay sensitivities with respect to all capacitances of the circuit is illustrated in Fig.(2.13).

2.3.6 Various types of circuits

A number of facilities for sensitivity analysis have been demonstrated with a switched capacitor circuit. Actually, the proposed methods are general and able to handle any type of switched linear networks. In addition, alternative methods are also employed in SCNAP4 to analyse continuous-time circuits. Different type of circuits will be illustrated and corresponding analysis results are given.

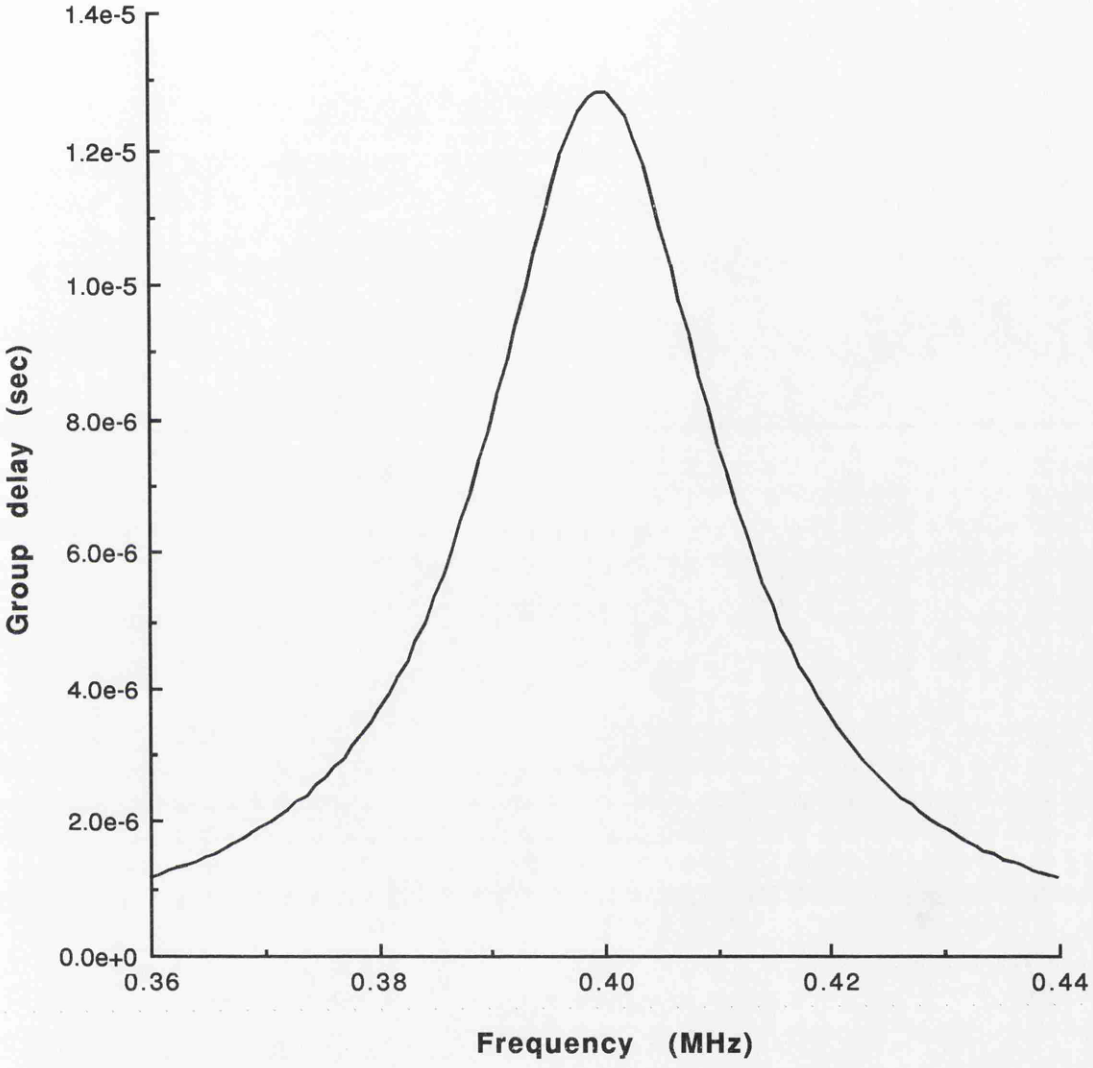


Fig. (2.11) Group delay

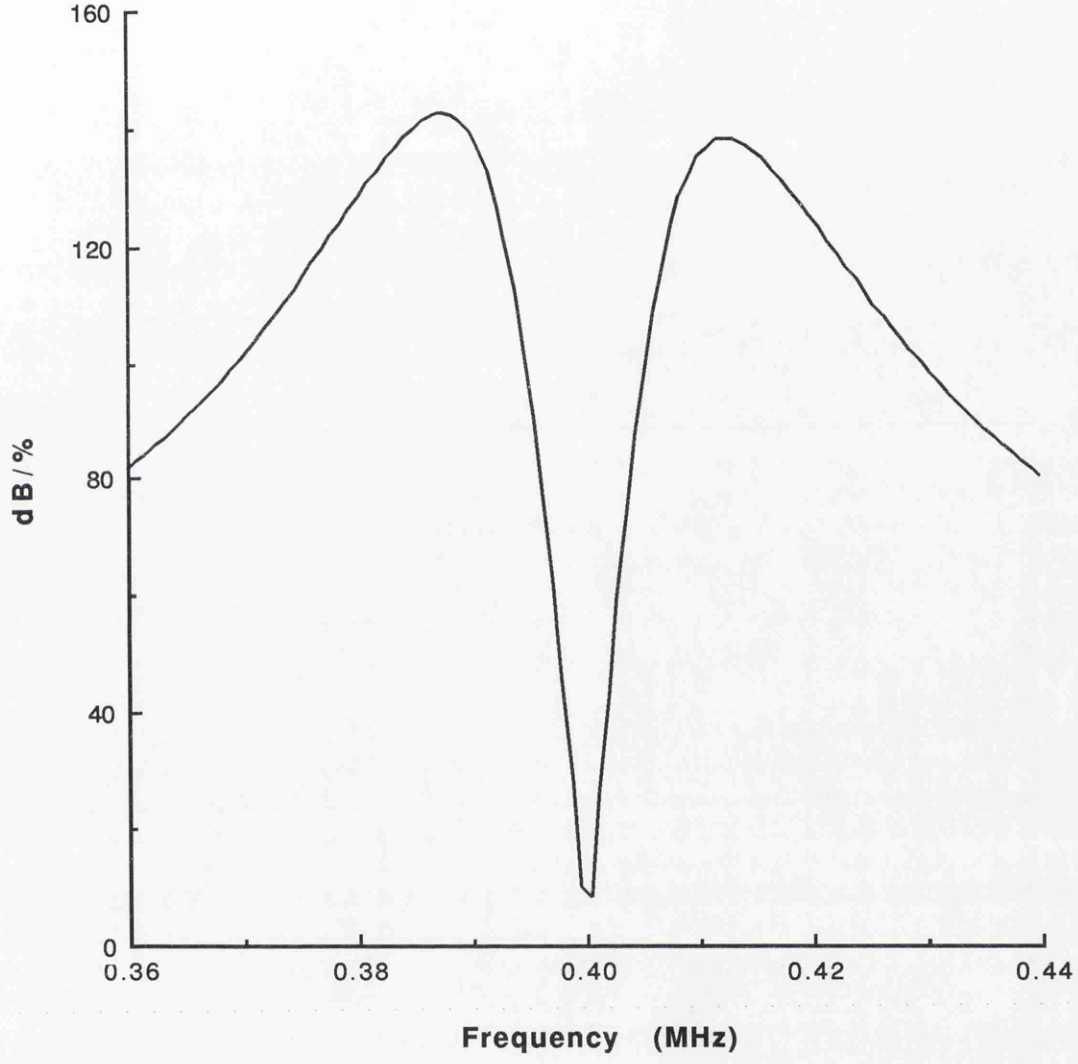


Fig. (2.12) Sum sensitivities w.r.t. all capacitances

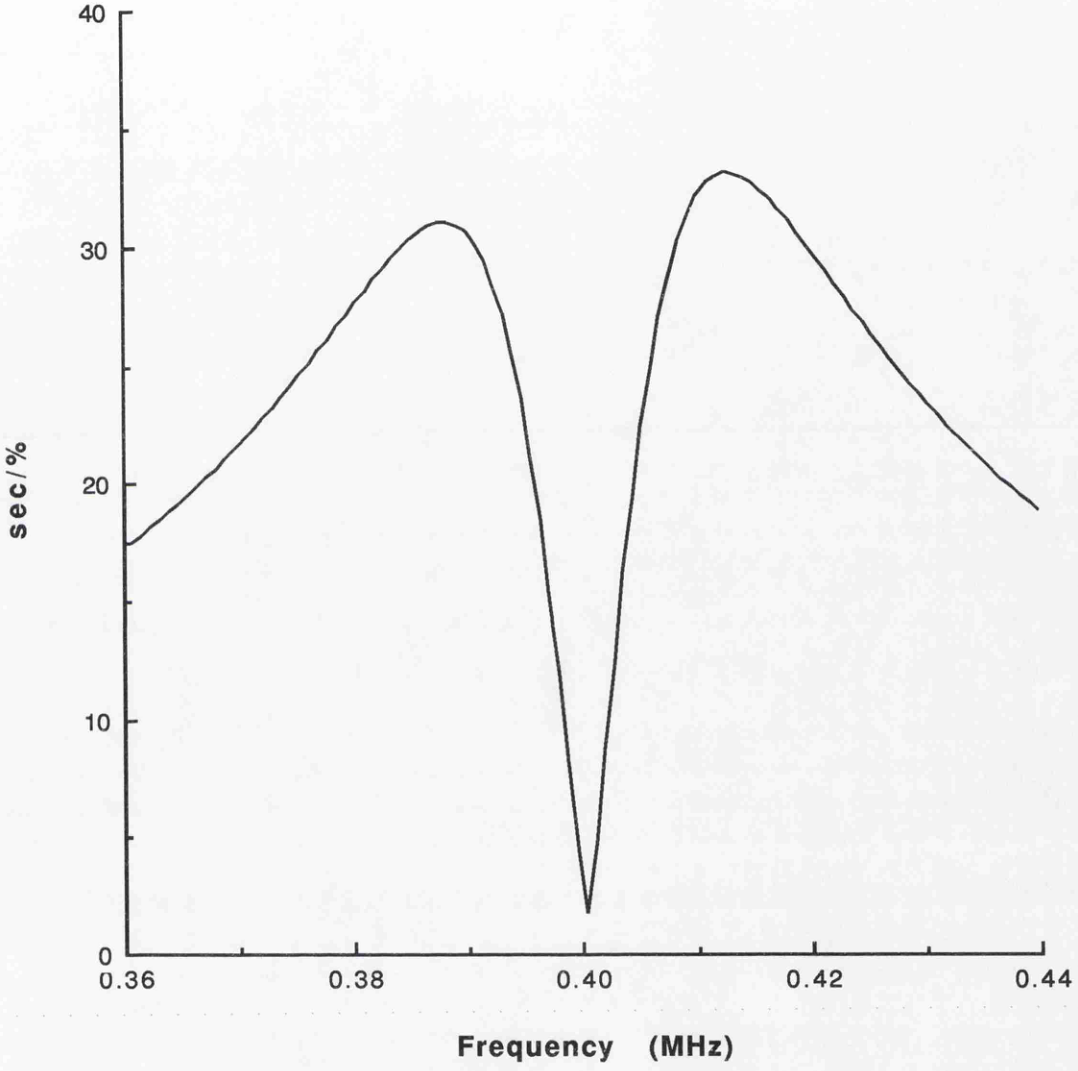


Fig. (2.13) Group delay sensitivity w.r.t. all capacitances

(a) Switched current (SI) circuit

The switched current technique has attracted much attention nowadays because it is fully compatible with standard CMOS technology and inherently applicable to high frequency operation. A fifth-order elliptic lowpass SI filter designed by XFILT[34] is shown in Fig. (2.14). The sum sensitivities with respect to all transconductances is given in Fig. (2.16)

(b) Passive prototype circuits

Prototype circuits are essential to filter synthesis. In the past, they can only be analysed by general purposed simulator, such as SPICE which is not adequate for switched linear circuits. Many special purposed SC simulators are not able to handle these simple prototype circuits. Separate simulators are required in the filter synthesis system. Now all the simulation tasks in a filter synthesis system can be taken by one simulator: SCNAP4. Fig. (2.15) shows the prototype of the fifth order elliptic ladder video filter and simulation results is given in Fig. (2.17).

(c) Transconductor - C circuits

Continuous-time circuits based on operational amplifiers and capacitors are still very popular these days due to the requirements of low noise and high frequency operation. A fifth-order video filter is shown in Fig. (2.18) and its group delay is illustrated in Fig. (2.19).

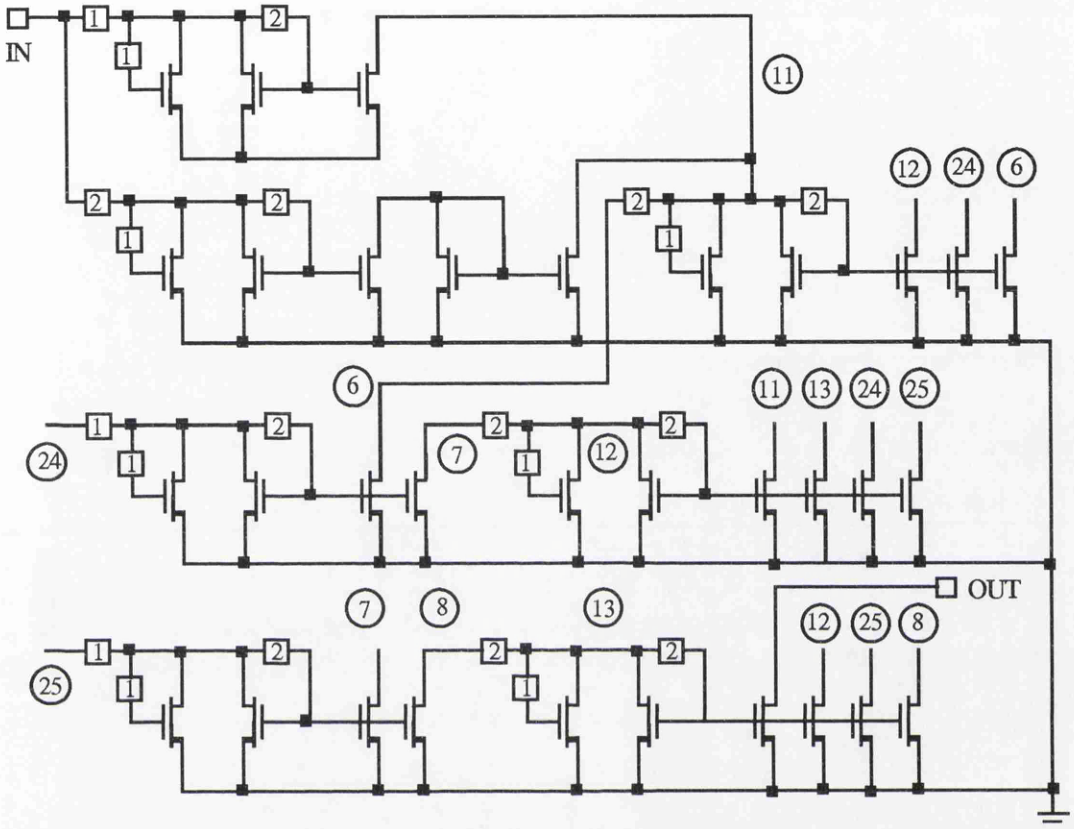


Fig. (2.14) 5th-order lowpass SI filter

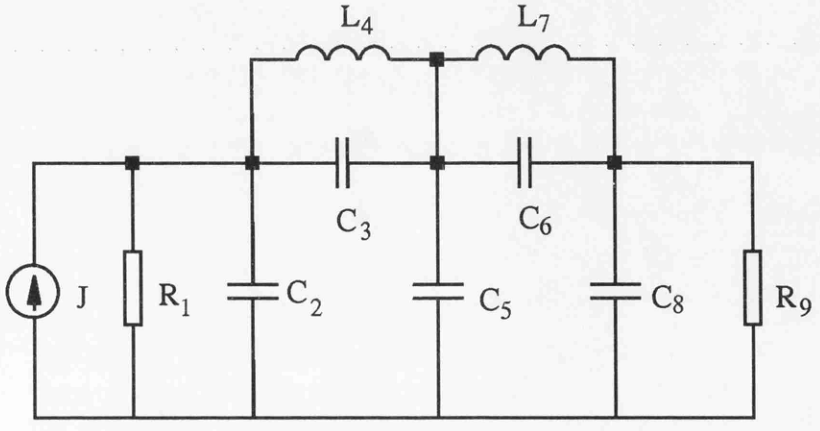


Fig. (2.15) 5th-order elliptic ladder video filter prototype

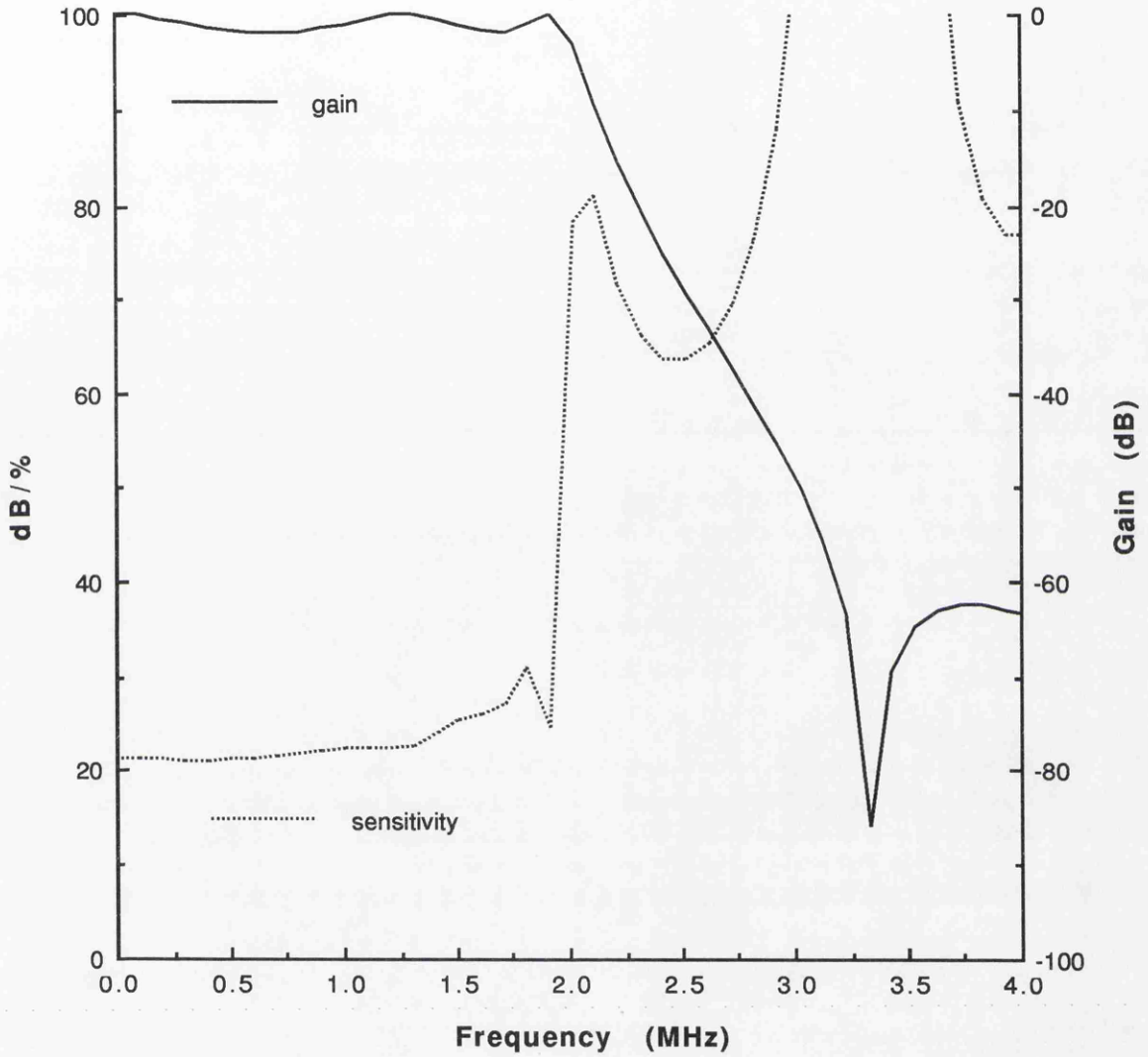


Fig. (2.16) Sensitivities w.r.t. all transconductances

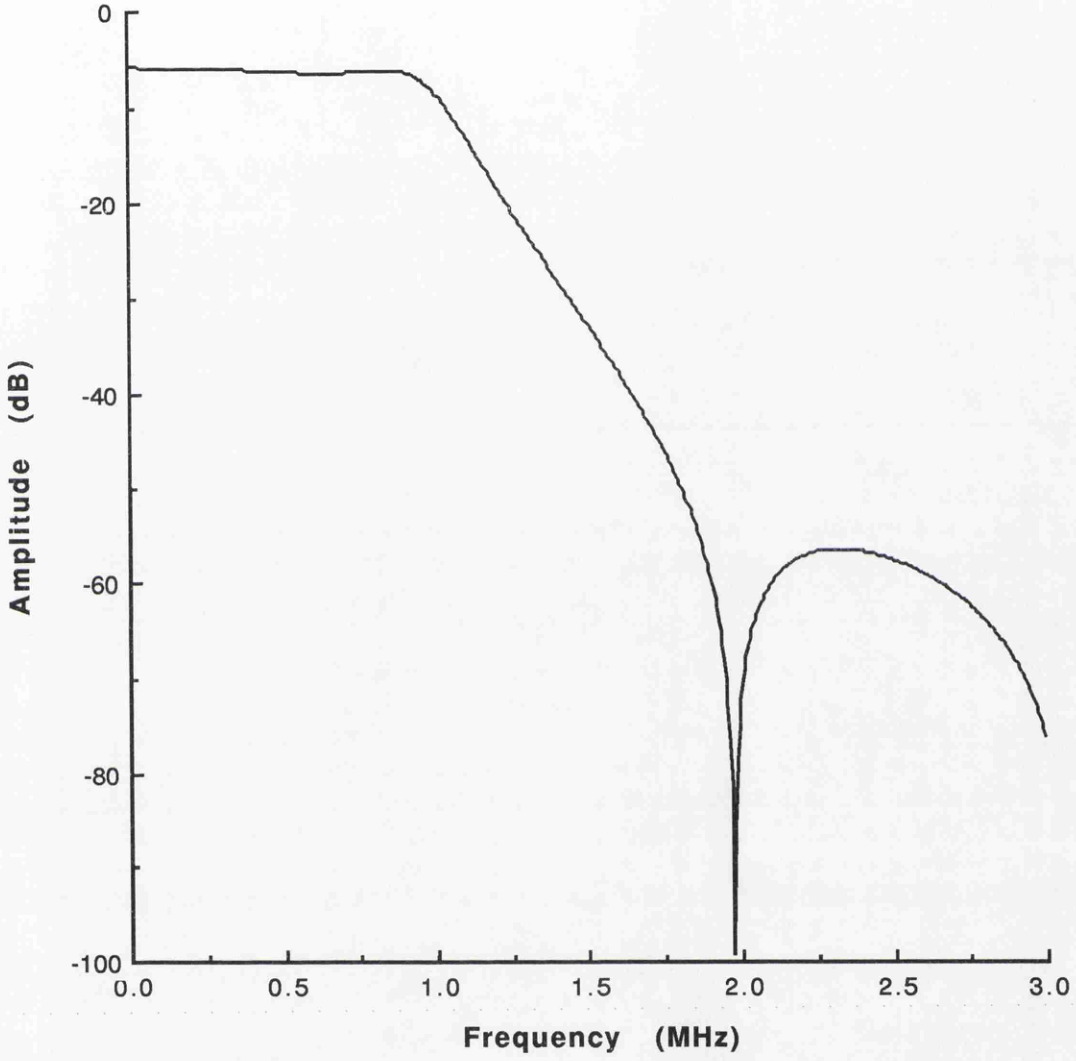


Fig. (2.17) Frequency response of the prototype circuit

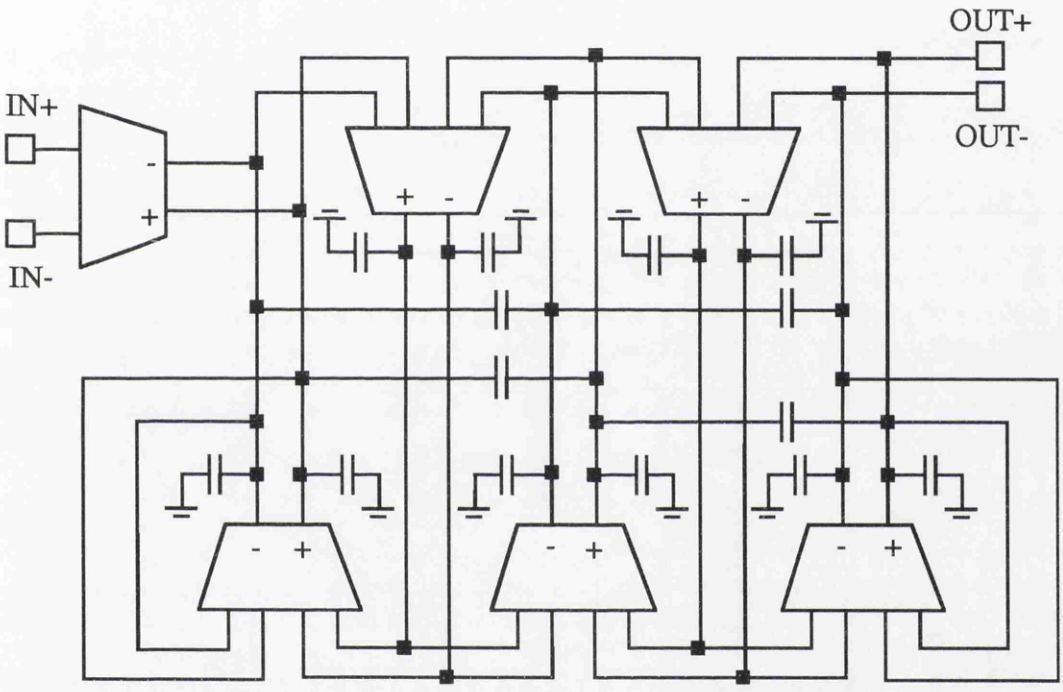


Fig. (2.18) 5th-order transconductor-C video filter

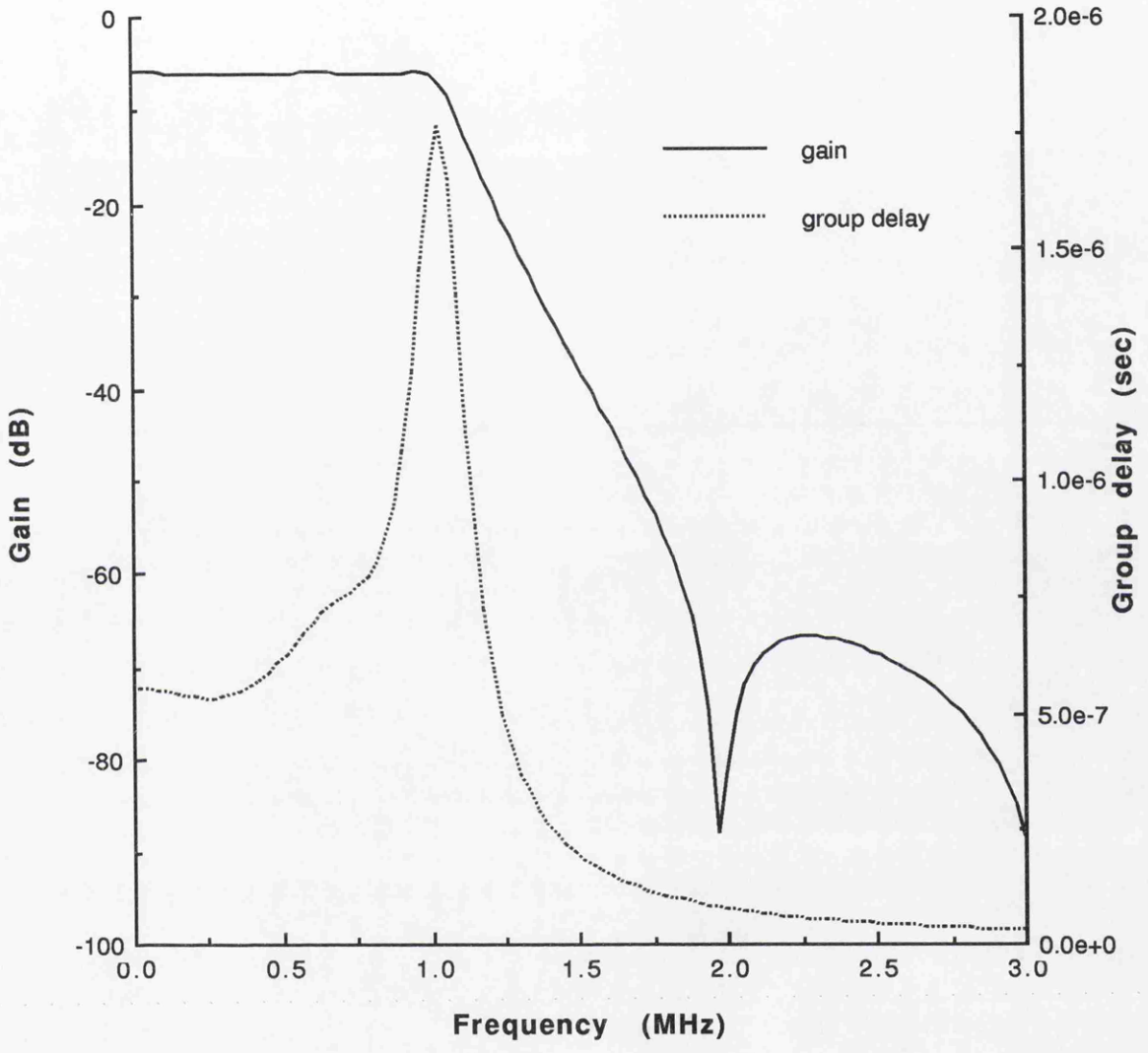


Fig. (2.19) Frequency response and group delay

2.4. SUMMARY

The Chapter began by reviewing the computational techniques for sensitivity analysis. It was seen that, despite the simplicity, the perturbation method provides neither efficient nor accurate solutions. To overcome this inadequacy, a sensitivity network approach was suggested. Although the accuracy can be assured, it is still not very efficient, since for each parameter of interest, the system equation must be solved. By contrast, the adjoint network technique provides a perfect solution to the above drawbacks. Only one extra system solution is required irrespective of the number of parameters. However, it is not directly applicable to general nonideal switched linear networks. A method for generating derivatives is proposed and the computer implementation shows that the algorithm can be conveniently combined with a frequency analysis procedure, since it draws on the same numerical approximation for the inverse Laplace transform. A number of examples illustrate the various applications of the sensitivity analysis. The most important nonidealities, such as GB product of the opamp, input and output impedances of the opamp, switch resistances and nodal parasitics are fully examined by using sensitivity analysis method. Group delay and sum sensitivity evaluation demonstrate other useful applications of sensitivity analysis. Numerical results confirm the efficiency and validity of the method.

REFERENCES

[1] J. Vlach, K. Singhal, and M. Vlach, "Computer oriented formulation of equations and analysis of switched capacitor networks," IEEE Trans. Circuits Syst., Vol. CAS-31, Sept. 1984, pp.753-765

[2] J. Vandewalle, H. De Man and J. Rabaey, "Time, frequency and z-domain modified nodal analysis of switched capacitor networks," IEEE Trans. CAS, Vol. CAS-28, No.3, Mar. 1981, pp.186-195

[3] A. D. Meakin, J. I. Sewell and L. B. Wolovitz, "Techniques for improving the efficiency of analysis software for large switched-capacitor networks," Proc. 28th Midwest Symposium on Circuits and Systems, Aug. 1985, pp.390-393

[4] M. L. Liu, Y. L. Kuo and C. F. Lee, "A tutorial on computer aided analysis of switched capacitor circuits," Proc. IEEE, Vol. 71, No.8, Aug. 1983, pp.987-1005

[5] M. Vlach, J. Vlach, K. Singhal and R. Chadha, "WATSCAD user's manual," Faculty of Eng., Inst. Computer Res. Univ. of Waterloo, Waterloo, Ont., Canada, July 1984

[6] J. B. Huges, N. C. Bird, I. C. MacBeth, "Switched currents-a new technique for analog sampled-data signal processing," Proc. of IEEE ISCAS, Portland, May 1989, pp.1584-1587

[7] J. B. Hughes, I. C. Macbeth, D. M. Pattullo, "Second generation switched-current signal processing," Proc. of IEEE ISCAS, New Orleans, May 1990, pp.2805-2808

[8] P. M. Sinn and G. W. Roberts, "A comparison of first and second generation switched-current cells," Proc. of IEEE ISCAS, May 1994, pp.5.301-5.304

[9] S. C. Fang, Y. P. Tsividis and O. Wing, "SWITCAP, a switched capacitor network analysis program," IEEE Circuits and Systems Magazine, Dec. 1983, pp.4-9 and pp.42-46.

[10] J. Rabaey, J. Vandewalle, H. De Man, "On frequency domain analysis of switched capacitor networks including all parasitics," Proc. of IEEE ISCAS, Chicago, Apr. 1981, pp.868-871

[11] A. Opal and J. Vlach, "Analysis and sensitivity of periodically switched linear networks," IEEE Trans. CAS, Vol.36, No.4, Apr. 1989, pp. 522-532

[12] L. B. Wolovitz and J. I. Sewell, "General analysis of large linear switched capacitor networks," IEE Proc. Vol.135, pt.G, No.3, Jun. 1988, pp.119-124

[13] C. B. Moler and C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix," SIAM Review, Vol.20, No.4, Oct. 1978, pp.801-836

[14] R. C. Ward, "Numerical computation of the matrix exponential with accuracy estimate," SIAM Journal of Numerical Analysis, Vol.14, No.4, Sept. 1977, pp.600-610

[15] V. Zakian, "Properties of the Imn and Jmn approximants and applications to numerical inversion of Laplace transforms and initial value problems," Journal Maths. and Appl., Vol.50, 1975, pp.191-222

[16] K. Singhal and J. Vlach, "Computation of the time domain response by numerical inversion of the Laplace transform," Journal of the Franklin Institute, Vol.299, No.2, Feb. 1975, pp.109-126

[17] V. Zakian and M. J. Edwards, "Tabulation of constants for full grade Imn approximants," Mathematics of Computation, Vol.32, No.142, Apr. 1978, pp.519-531

[18] Z. Q. Shang and J. I. Sewell, "Efficient sensitivity analysis for large non-ideal switched capacitor networks," Proc. of IEEE ISCAS, Chicago, May 1993, pp.1405-1407

[19] Z. Q. Shang and J. I. Sewell, "SCNAP4 users's guide (version 1.6)," Dept. of Electronics and Electrical Engineering, University of Glasgow, Aug. 1994

[20] J. B. Hughes, private communication, Philips Semiconductor Ltd., 1994

[21] S. J. Harrold, I. A. W. Vance and D. G. Haigh, "Second order switched-capacitor bandpass filter implemented in GaAs," Electronics letters, Vol.21, No.11, 23rd May 1985, pp.494-496

[22] D. G. Haigh, B. Singh and J. E. Franca, "Filters for the state of art microelectronics fabrication," Annual Report, Imperial College, London, Sept. 1982

[23] P. Li, Internal report, Dept. of Electronics and Electrical Engineering, University of Glasgow, 1989

[24] A. Fettweiss, D. Herbst, B. Hoefflinger, J. Pandel and R. Schweer, "MOS switched-capacitor filters using voltage inverter switches," IEEE Trans. CAS, Vol. CAS-27, No.7, 1980, pp.527-538

[25] J. E. Franca, "Switched-capacitor systems for narrow bandpass filtering," Ph.D. Thesis, Imperial College of Science and Technology, London, 1985

[26] J. A. Nossek and G. C. Temes, "Switched-capacitor filter design using bilinear element modelling," IEEE Trans. CAS, Vol. CAS-27, No.6, 1980, pp.488-491

[27] G. C. Temes, "Finite amplifier gain and bandwidth effects in switched-capacitor filters," IEEE J. Solid State Circuits, Vol. SC-15, Jun. 1980, pp.358-361

[28] K. Martin and A. S. Sedra, "Effects of op amp finite gain and bandwidth on the performance of switched-capacitor filters," IEEE Trans Circuits Syst., Vol. CAS-28, Aug. 1981, pp.822-829

[29] R. L. Geiger and E. Sanchez-Sinencio, "Operational amplifier gain-bandwidth product effects on the performance of switched capacitor networks," IEEE Trans. Circuits Syst., Vol. CAS-29, Feb. 1982, pp.96-106

[30] "SWAP 2.2 Tutorial," Laboratorium ESAT, Katholieke Universiteit Leuven, Sept. 1984, Belgium

[31] "Users's guide for SWITCAP (version 5)," Columbia University, New York, Aug. 1987

[32] L. B. Wolowitz and J. I. Sewell, "SCNAP Software - switched capacitor analysis programs," Dept. of Electronics and Electrical Engineering, University of Glasgow, 1987

[33] Z. Q. Shang, "Advanced simulation methods for periodically switched linear networks," Internal report, Dept. Electronics and Electrical Eng., University of Glasgow, 1993

[34] Y. Lu, R. K. Henderson, J. I. Sewell, "XFILT: an X-window based modern filter and equaliser design system," Proc. of EECTD, Davos, 1993, pp.305-310

[35] L. B. Wolovitz, "Computer aided analysis of periodically switched linear networks," PhD Thesis, University of Glasgow, Aug. 1987

CHAPTER 3

NOISE ANALYSIS

3.1 INTRODUCTION

Since the early years of last decade, noise analysis for SC circuits has been discussed frequently [1-7]. However most of the presented techniques are only adequate for small circuits. General and efficient methods were seldom mentioned. Recently, switched-current circuits have gained much attention. Naturally, their noise characteristics are of great concern to designers. Generally, two types of noise are considered in SC circuits, thermal noise from switches and flicker noise from operational amplifiers. For switched-current circuits, since the MOS transistors operate in saturation regions, both thermal and flicker noise exist.

Unlike the traditional analogue case (switch free), the folding of wideband white noise from high frequency bands to baseband, caused by the sampling nature of switched-circuits, leads to excessive computational costs. Traditionally, in order to calculate the whole noise spectral density, transfer functions from each noise source to output have to be evaluated. Furthermore, as the noise bandwidths are normally orders of magnitude higher than the sampling frequency, a very large number of system solutions are needed to cover the wide frequency range for taking folding effects into account. Therefore, without efficient and general methods, noise analysis is either very time consuming or the results tend to be unreliable.

The method developed in this Chapter is based on the adjoint network technique which had been successfully applied to nonideal sensitivity analysis of switched linear networks[8]. For noise analysis, only the adjoint system solution is required and the original system solution is no longer necessary. The I_{mn} approximation for extended state transition matrix and Hessenberg technique for solving the adjoint system as well as the pre-processing of a large amount of frequency independent material reduce computation costs dramatically and hence a highly efficient method results. By using a spectral analysis technique, the fold-back effects can be accurately evaluated. In the following sections, these aspects will be discussed in detail.

3.2 FUNDAMENTALS

Noise is referred to the unwanted random fluctuation in voltage or current which tend to interfere the desired signal. Since noise basically belongs to random process, principles of probability and statistics have to be employed to describe and calculate the noise behaviour of a electronic system. Two useful measures which describe the noise in an average power basis are frequently used throughout the Chapter, and hence are introduced as follows,

(a) Mean-square value

If f(t) is noise voltage or current, the mean-square value of f(t) can be defined as

$$\overline{f^2(t)} = \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} |f(t)|^2 \, dt$$

where T is the time interval under consideration. The mean-square value $\overline{f^2(t)}$ gives the time average power of f(t).

(b) Power spectral density

If $S_f(\omega)$ denotes the power spectral density of the signal f(t), the power P in f(t) is

$$P = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_f(\omega) d\omega$$

The power spectral density $S_f(\omega)$ describes the distribution of power of f(t) in frequency domain.

3.2.1 Sources of noise

For both switched capacitor and switched current circuits, a real switch is realised as a MOS transistor with a certain finite switch-on or switch-off resistance which usually exhibits thermal noise. In switched capacitor circuits, operational amplifiers are commonly used. Since they are built up with active components, mostly MOS transistors, several physical noise sources will be introduced, such as thermal noise, shot noise and flicker noise. MOS transistors in switched current circuits act in a very similar way as those in operational amplifier. Hence, the presence of thermal noise, shot noise and flicker noise is also apparent.

Thermal noise is due to the random thermal motion of free electrons in a conducting medium. The noise spectral density is independent of frequency and can be referred to as white noise.

Shot noise is generated by the gate leakage current and is present in pn junction due to the random passage of each carrier across the junction resulting in a large number of random independent current pulses. It is also independent of frequency and can be treat as another source of white noise. However, it is usually very small and can be neglected.

Flicker noise is dependent on contamination and crystal imperfections, which are factors that can vary randomly for different transistors, or integrated circuits. However, experiments show that the noise spectral density has 1/f frequency dependence. It is evident that the flicker noise is most significant at low frequencies.

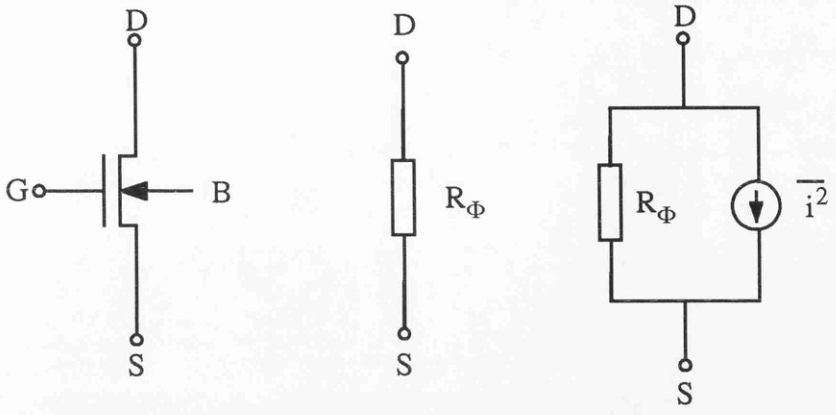
Since the thermal noise and shot noise are considered as white noise in electronics circuits, the under sampling of these broad band noise sources will complicate the noise analysis of a sampling system. It will be described later.

3.2.2 Noise models

In the preceding section, the various physical sources of noise in switched linear networks were described. For computer application, it is necessary to build up mathematical models for these noise sources [9-12]. The most commonly used components in switched linear networks are switches, operational amplifiers and switched current memory cells. Detailed description of their noise models will be presented as follows.

(a) Switches

Switches are implemented by MOS transistors. The resistance of the channel is modulated by the gate source voltage. Such a resistance characteristic exhibits thermal noise and usually it is the major noise source in MOS transistors. Thus, the noise model of switches can be derived in a straightforward manner. As shown in Fig. (3.1), it is a parallel connected noiseless resistor R_Φ with a noise current source, where R_Φ could be either switch-on resistance R_{on} or switch-off resistance R_{off}.



MOS transistor Switch model Noise model

Fig. (3.1) Noise model of the MOS transistor switch

The noise current amplitude is equal to

$$\overline{i^2} = 4k\theta G_\Phi \Delta f$$

where k is Boltzman's constant, θ denotes the temperature in degrees Kelvin, G_Φ represents the switch conductance, and Δf is the bandwidth under consideration.

(b) Operational amplifiers

For a complex circuit building block, it is convenient and helpful to represent the noise contribution as an equivalent input noise source generator which produces the same output noise as the circuit under consideration. The scheme has another advantage that it allows direct comparison between the input signal and the equivalent input noise. To generate the equivalent input noise source of an operational amplifier, it usually employs modelling all transistors with their equivalent noise sources and then calculating the total equivalent input noise [13].

The noise model for a MOS operational amplifier is illustrated in Fig. (3.2).

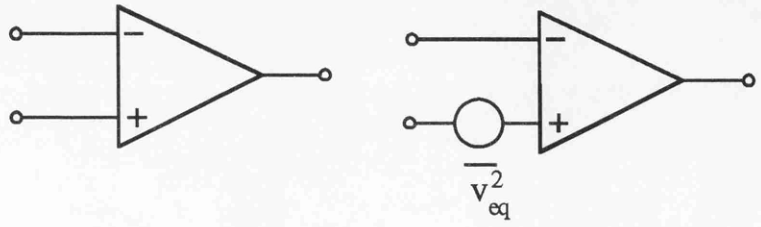


Fig. (3.2) Noise model for a MOS operational amplifier

It has been found that the equivalent input noise source consists of both thermal noise and flicker noise. The exact expression of the spectral density for the equivalent input noise is complicated to derive and is configuration dependent. However, empirical data shows that it has the property shown in Fig. (3.3). At the low frequency, it is dominated by the flicker noise, but at higher frequencies the thermal noise takes over.

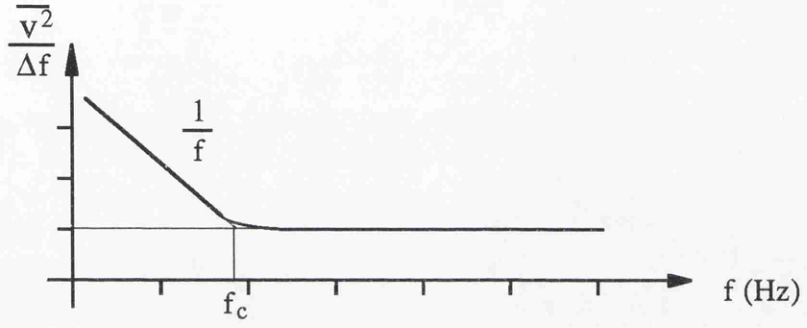


Fig. (3.3) Typical spectral density
of equivalent MOS opamp input noise source

(c) Memory cells

The switched current memory cell is a basic circuit building block in switched current networks. It is important to understand the noise performance of the memory cell. Consider a simplified memory cell shown in Fig. (3.4).

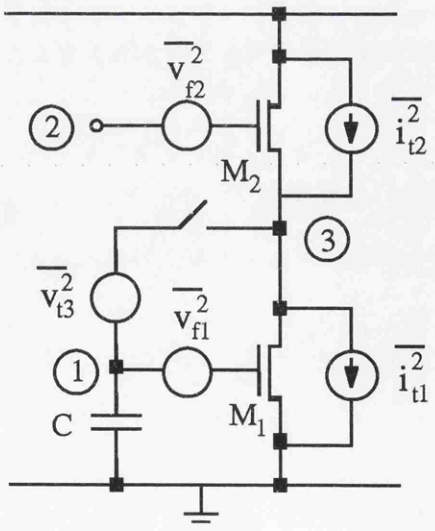


Fig. (3.4) Memory cell with noise sources

The memory transistors M_1 and current source transistor M_2 operate in saturation region, hence generate both flicker noise and thermal noise.

The flicker noise is modelled as a gate-referred noise which generates an equivalent drain noise current. The amplitude of flicker noise current is given by

$$\overline{i_f^2} = g_m^2 \frac{K_f}{WLf} \Delta f$$

where K_f is a process dependent constant, g_m represents the device transconductance at the operating point, W and L are the channel width and length of the transistor.

The thermal noise is modelled as a drain-referred noise current with the mean-square value

$$\overline{i_t^2} = 4k\theta\left(\frac{2}{3}g_m\right)\xi\Delta f$$

where ξ is a process dependent constant.

The thermal noise generated from the switch can usually be neglected because the noise voltage v_{t3} has little impact on the v_{gs} of M_1 due to the attenuation of v_{t3} by the g_o/g_m ratio of M_1.

Since noise currents are uncorrelated, they may be combined to give an equivalent noise current of the memory cell as

$$\overline{i_{eq}^2} = \overline{i_{t1}^2} + \overline{i_{f1}^2} + \overline{i_{t2}^2} + \overline{i_{f2}^2}$$

By modelling the MOS transistor with a macro-model which consists of linear components [14], the small signal equivalent circuit of the memory cell is shown in Fig. (3.5).

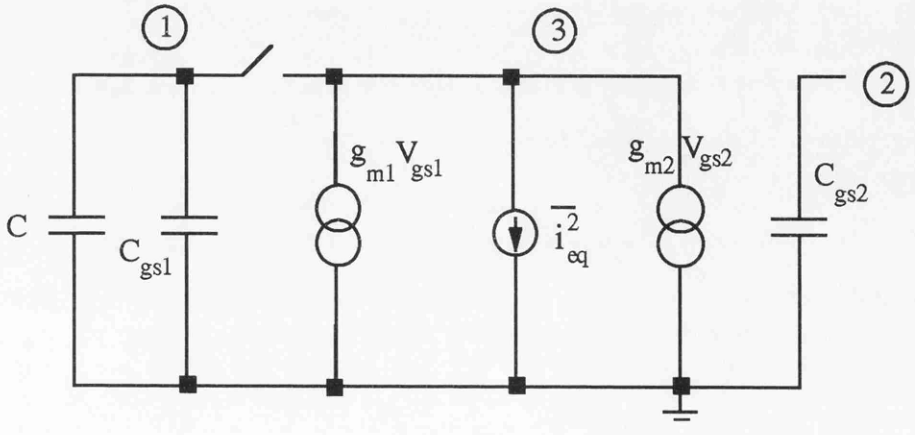


Fig. (3.5) Equivalent circuit of memory cell with noise sources

Based on the memory cell, the noise models of integrators can be built easily. It should be stated that the flicker noise can be removed by the correlated double sampling process [15], consequently this will lead to the increase of the white noise level.

3.2.3 Sampling theory

Consider a band-limited signal f(t) which was sampled by a unit impulse train. Denote the sampled signal by $f_s(t)$ and the unit impulse train by $\delta_T(t)$, then

$$f_s(t) = f(t)\delta_T(t) \tag{3.1}$$

where

$$\delta_T(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \tag{3.2}$$

The spectral density of $f_s(t)$ is

$$F_s(\omega) = \omega_s \sum_{n=-\infty}^{\infty} F(\omega - n\omega_s) \tag{3.3}$$

and

$$\omega_s = \frac{2\pi}{T}$$

Equation (3.3) reveals that the spectral density of the sampled signal $f_s(t)$ consists of replicas of the original spectral density about multiples of the sampling frequency, as shown in Fig. (3.6)

Apparently in Fig. (3.6), the signal bandwidth B is less than one-half the sampling frequency ω_s. In this case, the original can be retrieved by using an ideal low-pass filter. However, if the signal bandwidth is larger than $\omega_s/2$, frequency components originally located above Nyquist frequency will appear below that point. If the frequency range of interest is from 0 to $\omega_s/2$, the shaded area as shown in Fig. (3.7), is known as aliasing effect.

Since the white noise bandwidth is usually much higher then the sampling frequency, The aliasing of white noise from high frequency bands into the base band occurs and therefore makes the noise analysis of the sampling system complicated.

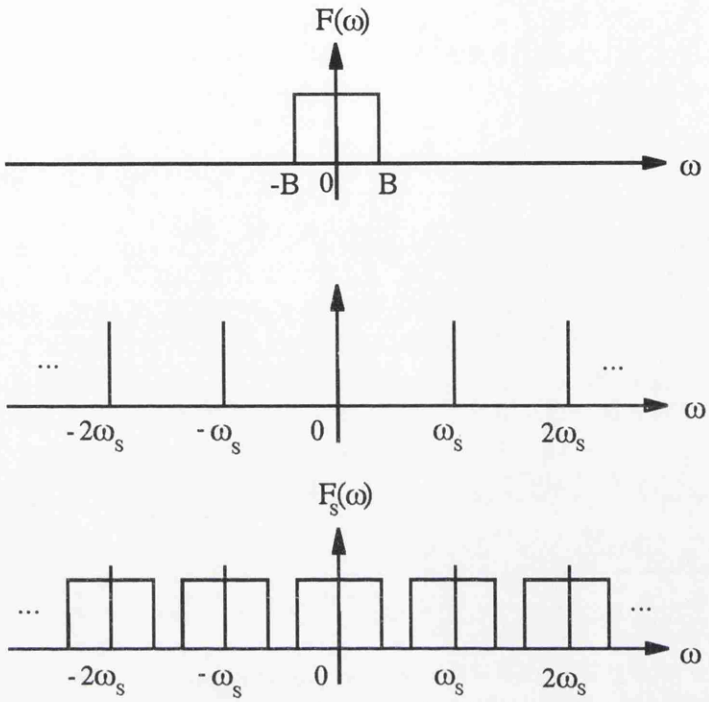


Fig.(3.6) Idealised sampling

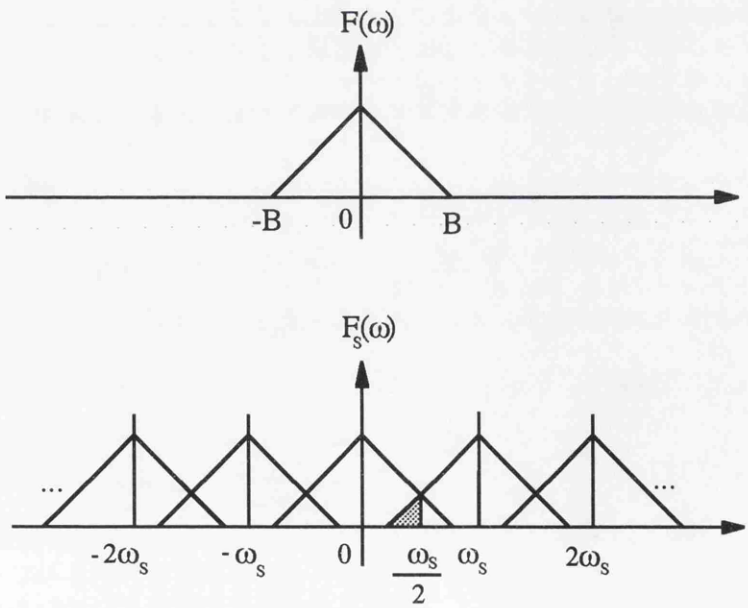


Fig. (3.7) Aliasing effects

3.2.4 Spectrum Analysis

To investigate the aliasing effect, it is essential to calculate the frequency components in the output due to the input signal at other frequencies.

Consider a general switched linear network which can be characterised by

$$\begin{bmatrix} I & & & -P_1 \\ -P_2 & I & & \\ & \ddots & \ddots & \vdots \\ & & -P_N & zI \end{bmatrix} \begin{bmatrix} V_1(z) \\ V_2(z) \\ \vdots \\ V_N(z) \end{bmatrix} = \begin{bmatrix} \Sigma W_1(z) \\ \Sigma W_2(z) \\ \vdots \\ z\Sigma W_N(z) \end{bmatrix} \tag{3.4}$$

Notations of (3.4) can be found in Chapter 2.

The system equation (3.4) is in z domain, to obtain the solution at a particular input frequency ω_0, substitute

$$z = e^{j\omega_0 T} \tag{3.5}$$

Apply Poisson's formula to the right hand side of (3.4) gives

$$\begin{aligned} \sum W_k(e^{j\omega_0 T}) &= \sum B_k w(nT + t_k) e^{-jn\omega T} \\ &= \frac{1}{T} \sum B_k W(\omega - n\omega_s) e^{j(\omega - n\omega_s)t_k} \end{aligned} \tag{3.6}$$

In reality, the input signal is usually of the form

$$w(t) = \cos(\omega_0 t) \tag{3.7}$$

The Fourier transform of (3.7) is

$$W(\omega) = \pi\left[\delta(\omega - \omega_0) + \delta(\omega + \omega_0)\right] \tag{3.8}$$

Obviously,

$$W(\omega - n\omega_s) = \pi\left[\delta(\omega - \omega_0 - n\omega_s) + \delta(\omega + \omega_0 - n\omega_s)\right] \tag{3.9}$$

Substituting (3.9) into (3.6) results in

$$\sum W(e^{j\omega_0 T}) = \begin{cases} \frac{\pi}{T} B_k e^{j\omega_0 t_k} & \omega = n\omega_s + \omega_0 \\ \frac{\pi}{T} B_k e^{-j\omega_0 t_k} & \omega = n\omega_s - \omega_0 \end{cases} \tag{3.10}$$

Band Definition:

A frequency band n is defined as the frequency interval from $(n-1)\omega_s/2$ *to* $n\omega_s/2$.

From (3.10), the following relationship between input frequency and output frequency can be derived

$$\omega_0 = \begin{cases} \omega + \frac{\omega_s}{2}(1 - OB - IB) & \lambda \text{ is odd} \\ \omega + \frac{\omega_s}{2}(IB - OB) & \lambda \text{ is even} \end{cases} \tag{3.11}$$

where OB is the output frequency band, IB denotes the input frequency band, λ represents the number of bands between the IB and OB.

By taking the window function into account, the solution of the spectrum analysis is

$$\phi(\omega) = \sum_{k=1}^{N} D_k(\omega) V_k(\omega) \tag{3.12}$$

Here, $V_k(\omega)$ is obtained from the solution of (3.4) with the right hand side defined by (3.10) and (3.11).

Any frequency component in the output can be calculated no matter where the input frequencies are located. Therefore, the aliasing effect can be studied exactly and further application to the noise calculations is feasible.

3.2.5. Multi-independent sources

In frequency analysis, usually only one independent excitation is applied to the network and the output response of the network due to the input excitation is calculated. However, if the network has multi-independent sources, for example, various types of noise sources, the algorithms developed in [16] are no longer valid and further generalisation is required.

Consider the same system as described by (3.4). Let the *j*th input excitation be approximated by *m*th order polynomials

$$w_k^j(t) \approx \sum_{i=0}^{m} \alpha_{i,k}^j t^i d_j \tag{3.13}$$

where $\alpha_{i,k}^{j}$ are the coefficients of the polynomial approximations of $w_k^j(t)$ in the kth time slot. d_j is a constant vector which expresses the topology of the contributing source to the right hand side excitation vector.

Assuming there are overall q independent sources, the circuit equation in time domain can be written as

$$C_k \dot{x}_k(t) + G_k x_k(t) = \sum_{j=1}^{q} \sum_{i=0}^{m} \alpha_{i,k}^{j} t^i d_j \tag{3.14}$$

Taking the Laplace transform,

$$X(s) = \left[sC_k + G_k\right]^{-1} \left\{ C_k x_0 + \sum_{j=1}^{q} \sum_{i=0}^{m} \frac{\alpha_{i,k}^{j} i!}{s^{i+1}} d_j \right\} \tag{3.15}$$

Taking the inverse Laplace transform,

$$x_k(t) = P_k(t) x_{k-1}(nT + \sigma_k) + \sum_{j=1}^{q} \sum_{i=0}^{m} \alpha_{i,k}^{j} B_{i,k}^{j}(t) \tag{3.16}$$

where the definition of P_k is unchanged, and the $B_{i,k}^{q}$ are defined as

$$B_{i,k}^{j} = \pounds^{-1} \left\{ \frac{i!}{s^{i+1}} \left[sC_k + G_k\right]^{-1} d_j \right\} \qquad (j = 1, 2, \ldots, q) \tag{3.17}$$

Correspondingly, the right hand side of (3.4) should be changed into

$$\sum W_k(z) = \sum_{j=1}^{q} \sum_{n=0}^{\infty} \left[\sum_{i=0}^{m} \alpha_{i,k}^{j} B_{i,k}^{j} \right] z^{-n} \qquad (j = 1, 2, \ldots, q) \tag{3.18}$$

Note that, contributions from each independent source can now be exactly computed from (3.18).

3.3 NOISE ANALYSIS METHODS

3.3.1 Assumptions

Noise is a purely random signal, the instantaneous value of the waveform cannot be predicted at any time. The only information available for use in circuit calculations concerns the mean square value of the signal. There are two fundamental assumptions

about the input process which should be made during the derivation of the noise analysis method for periodically switched networks,

(1) The input is a stationary stochastic process;

(2) No correlation between the input noise sources.

It has been found that for the noise sources under consideration, these assumptions are practically reasonable.

3.3.2. Classical method

For periodically switched networks, due to the inherent time variance property, the output noise in general is a nonstationary process even when input is a stationary process. However it is periodically stationary in the wide sense [17-18] which makes the calculation of an average spectral density possible. In other words, if all the transfer functions from the noise sources to the output are generated, the overall output noise spectral density of the network can be determined from

$$S_o(\omega) = \sum_{i=1}^{\mu} S_i(\omega)\left|H_i(\omega)\right|^2 \tag{3.19}$$

where $H_i(\omega)$ denotes the transfer function from the *i*th noise source to the output, $S_i(\omega)$ is the input noise power spectral density and μ represents the number of noise sources in the network.

To obtain $H_i(\omega)$, the classical method proceeds by solving the network equations with each noise source present independently and then repeating until all the noise sources are processed. This method, though simple and straightforward, leads to high computation costs since there are very likely to be a large number of noise sources in a practical circuit. For a sampled system, as has been discussed in previous sections, the folding of wide band white noise from the higher frequency bands into the base band, greatly increases the number of network solutions required to take the aliasing contribution into account. Hence, this precludes the possibility of using the classical method for large nonideal switched linear networks.

3.3.3 Adjoint network methods

For a general sampled system, the classical method requires $\mu\beta$ network solutions per frequency point to obtain all the transfer functions for noise calculation. Here β is the

number of bands to be considered. The task of improving the efficiency of noise analysis can be divided into two different aspects,

(a) reducing the overall number of system solutions;

(b) producing efficient methods for each individual system solution.

The second aspect has already been considered when SCNAP4 [8] was being developed. A solid platform for noise analysis has been built successfully. In the following, attention will be focused on the first issue.

For general linear switched networks, consider the *i*th noise source, the z domain system equations are similar to (3.4), and given here for the convenience of description,

$$\begin{bmatrix} I & & & -P_1 \\ -P_2 & I & & \\ & \ddots & \ddots & \vdots \\ & & -P_N & zI \end{bmatrix} \begin{bmatrix} V_1^i(z) \\ V_2^i(z) \\ \vdots \\ V_N^i(z) \end{bmatrix} = \begin{bmatrix} \sum W_1^i(z) \\ \sum W_2^i(z) \\ \vdots \\ z\sum W_N^i(z) \end{bmatrix} \tag{3.20a}$$

or in a compact form

$$MX_i = W_i \tag{3.20b}$$

The transfer function from the *i*th noise source to the output can be written as

$$H_i(\omega) = \tilde{d}^t X_i \tag{3.21}$$

with $\tilde{d}$ defined in expression (2.45).

Define the adjoint system as

$$M^t X_a = -\tilde{d} \tag{3.22}$$

Substituting (3.20b), (3.22) into (3.21) gives

$$H_i(\omega) = -X_a^t W_i \qquad \text{for} \quad i = 1,2,\cdots,\mu \tag{3.23}$$

Equation (3.23) reveals that once the adjoint system solution X_a is obtained, transfer functions of all noise sources can be calculated by merely few multiplications and subtractions since the excitation vector W_i is usually very sparse. The computational cost for evaluating (3.23) is trivial compared to the solution of equation (3.22). The

spectral analysis technique presented in section 3.2.4 should be employed here in order to account for the folding effects. The total number of system solutions reduces from $\mu\beta$ to β. The reduction in computation is significant.

Having all the transfer functions, the total noise power spectral density at the output is then calculated by superposition

$$S_T(\omega) = \sqrt{\sum_{IB=1}^{\beta} \sum_{i=1}^{\mu} \left|H_i(\omega,\omega_0)\right|^2 S_i(\omega_0)} \tag{3.24}$$

where $S_i(\omega_0)$ is the *i*th input noise power spectral density.

Note that the folding effects from high frequencies to a specific band are evaluated over a user defined noise bandwidth. This is based on the fact that the noise is normally bandlimited by op-amp frequency response and time constant effects from switch resistances and circuit capacitances (actual or parasitic).

3.3.4 Computer implementation

The noise analysis method presented above has been implemented in the program SCNAP4 [19]. Two types of noise source (voltage and current) are available, they can be either flicker noise or white noise. The foldover wideband white noise from high frequency bands to baseband can be calculated. The method may be summarised by the following computer algorithm:

```
for (freq_out = fstart; freq_out <= fstop; freq_out += fstep) {
    S_T = 0;
    for (IB = OB; IB <= β; IB++) {
        if( λ == even) {
            freq_in = freq_out + f_s (IB - OB) / 2;
        } else {
            freq_in = freq_out + f_s (1 - OB - IB) / 2;
        }
        X_a^t = -d̃^t M^{-1};
        for (i = 1; i <= μ; i ++) {
            H_i = -X_a^t W_i;
            S_T = S_T + H_i^2 S_i;
        }
    }
    S_T = sqrt (S_T);
}
```

It is evident that the only computation step of high cost is the solution of the adjoint system. However, the adjoint system can be solved by using the Hessenberg approach which has already been proved very effective in frequency domain analysis as well as in the sensitivity analysis of switched linear networks. Hence a highly efficient method can be expected.

The efficiency of the proposed method and its computer implementation can be demonstrated by noise analysis statistics listed in Table (3.1)

Table (3.1) Statistics of noise analysis

Circuit name	Signal type	Circuit size	Number of slots	Points per band	Number of bands	Noise analysis (sec)
bp6(SC)	S/H	48	2	100	1	3.76
bp6(SC)	S/H	48	2	100	50	187.69
bp6(SC)	S/H	48	2	100	100	382.25
nos5(SC)	S/H	28	4	100	1	2.62
nos5(SC)	CONT	28	4	100	1	12.13
nos5(SC)	S/H	28	4	100	10	26.81
swc6(SI)	S/H	37	2	100	1	2.41
elp3(SI)	S/H	38	2	106	1	2.92
elp3(SI)	S/H	38	2	106	6	17.26
elp3(SI)	CONT	38	2	106	1	12.14
elp3(SI)	CONT	38	2	106	6	77.55

Note: Above data obtained on SUN-Sparc ELC station

It can be seen from Table (3.1) that even for circuits of reasonable size, SCNAP4 is able to give noise analysis results in fairly short time. Powerful computer, such as SUN-Sparc20 which is three times faster than ELC station, is more appropriate for such calculation. Thus, noise analysis of large switched systems is feasible and the application to some degree of noise optimisation is a realistic possibility. However, since no other switched-network analysis software with noise calculation facility is available, it is not possible to compare their run time efficiencies with SCNAP4. But, as has been stated in Chapter 2, the superior efficiency of SCNAP4 observed during the comparison with other packages in frequency analysis should also translate into the noise computation.

A note of caution is appropriate, since noise in switched-networks is a complicated physical phenomena, the validity of the simulated noise contribution depends very much on the accuracy of the noise models and appropriate values of related parameters. A number of circuits were analysed by SCNAP4, numerical results were checked either experimentally or from the literature where results were compared with measurements. Good agreement confirms the validity of the method.

3.3.5 APPLICATIONS

Noise analysis of several typical circuits of different types will be given now to illustrate the performance of the proposed methods.

(a) Switched capacitor circuits

In switched capacitor circuits, the time constant produced by the combination of capacitor and switch-on resistance has to be much smaller than the sampling period, otherwise the complete charge transfer cannot be ensured. Similarly, the bandwidth of the operational amplifier should be much larger than sampling frequency. As a result, the fold back of the white noise from higher frequency bands to base band is common.

A sixth order bandpass filter is selected from [5] and shown in Fig. (3.8a). The op-amp noise components are shown in Fig. (3.8c) and given to be $80nV/\sqrt{Hz}$ with corner frequency 1kHz. The switch on resistance is set to 1Ω and the gain bandwidth product of opamp is 50M Hz. The frequency response is shown in Fig. (3.9a) and the output noise spectrum density is illustrated in Fig. (3.9b). A close agreement between simulation and results from [5] can be noted. The folding effects are also shown in Fig. (3.9b) where it can be seen that circuit time constant and amplifier bandwidth ultimately limit the extent of noise fold back as the number of bands considered is increased. It should also be observed that 1/f noise components are present, but have little effect because of the band pass filter characteristic. The run time statistics can be found in Table (3.1).

(b) Switched current circuits

The noise behaviour of switched-current circuits has received considerable attention since the emergence of switched-current techniques [20-21]. A third-order elliptic ladder filter is considered and shown in Fig. (3.10). The noise bandwidth is about 120MHz. If the reference noise is $1nA/\sqrt{Hz}$, then the circuit response and noise behaviour calculated by SCNAP4 are illustrated in Fig. (3.11a) and Fig. (3.11b) respectively. The same circuit was also analysed by SCALP2 (a frequency domain analysis package developed by Philips Semiconductor Ltd.). The difference between the two simulation results is trivial. The complete noise analysis only takes 17.26 seconds on SUN-Sparc ELC station. It is recommended to use the noise macromodels for switched current integrators rather than for each individual transistors. This would simplify the noise analysis of the switched current filters significantly.

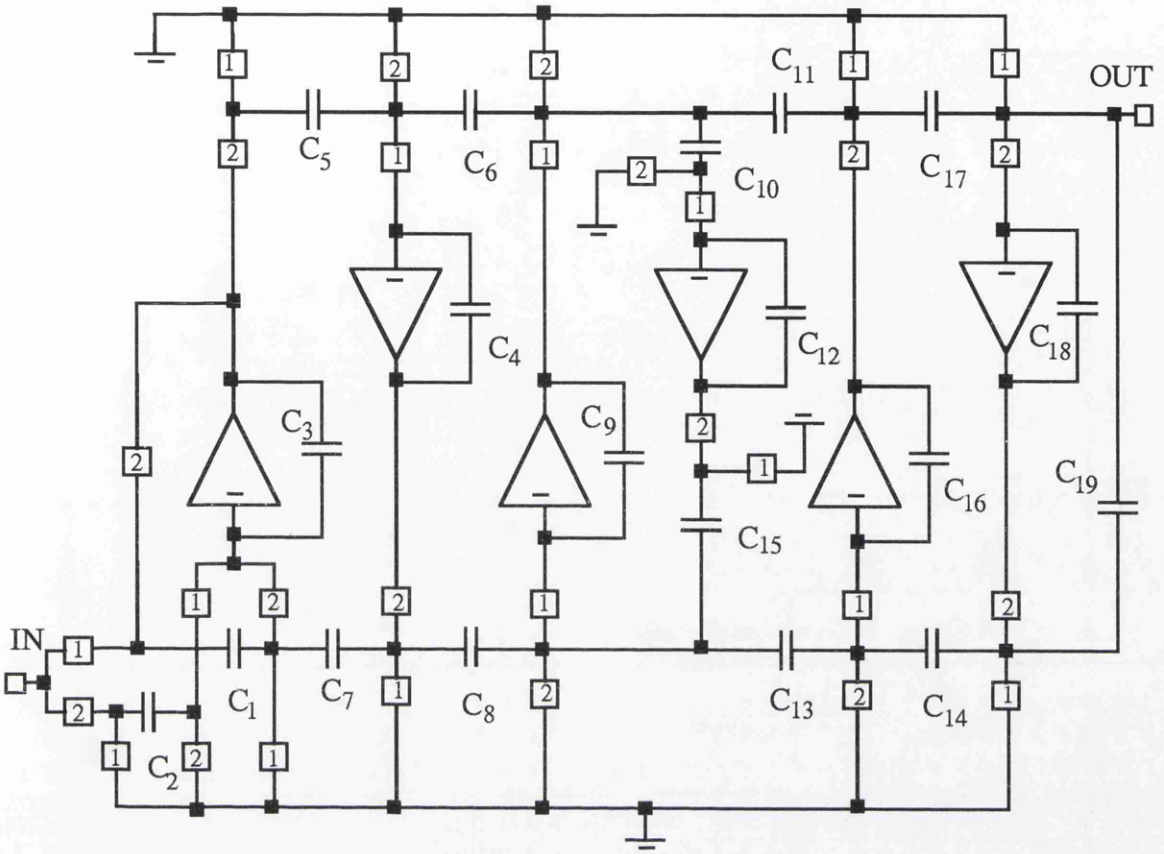


Fig. (3.8a) A 5th-order bandpass SC filter: circuit diagram

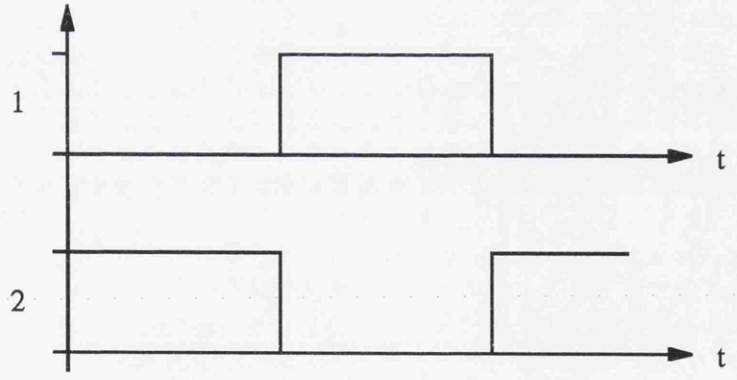


Fig. (3.8b) clock waveforms

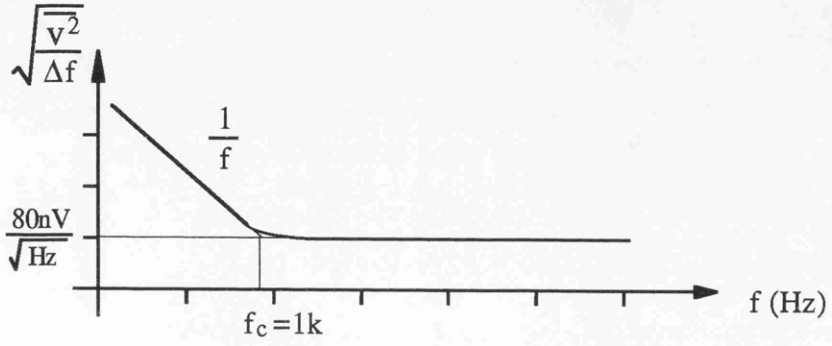


Fig. (3.8c) Noise spectral density of opamp

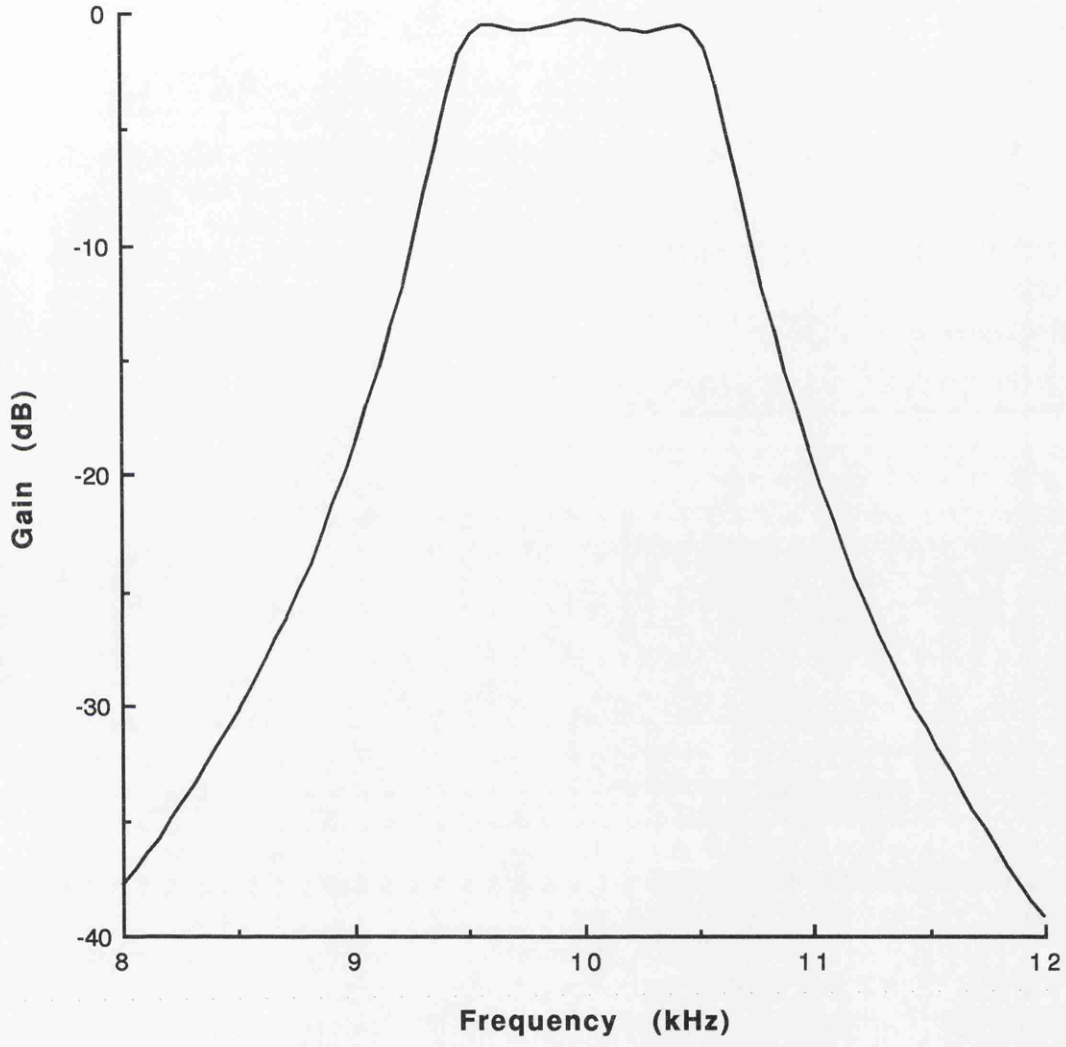


Fig. (3.9a) Frequency response of a 6th-order bandpass SC filter

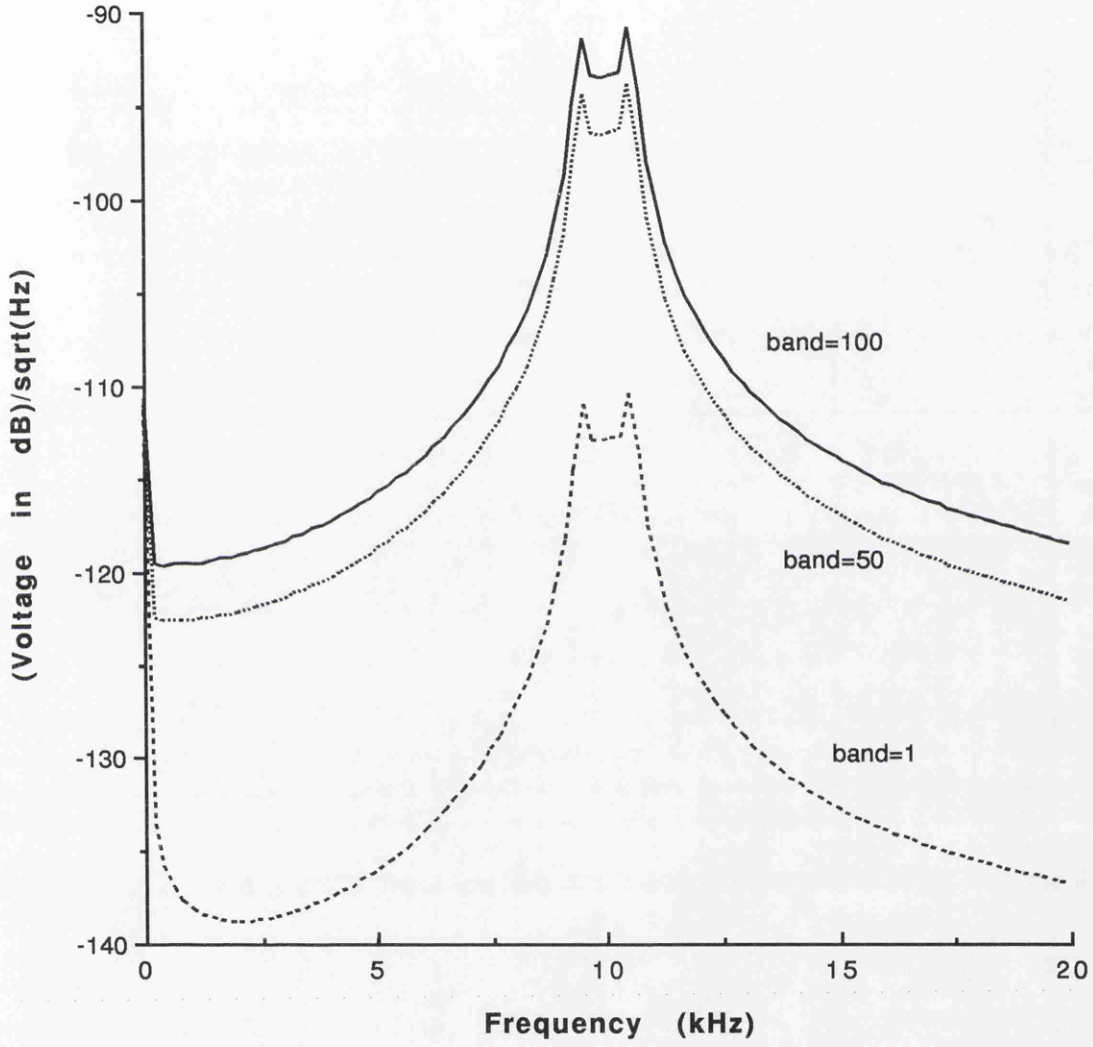


Fig. (3.9b) Noise behaviour of 6th-order bandpass SC filter with folding back effects

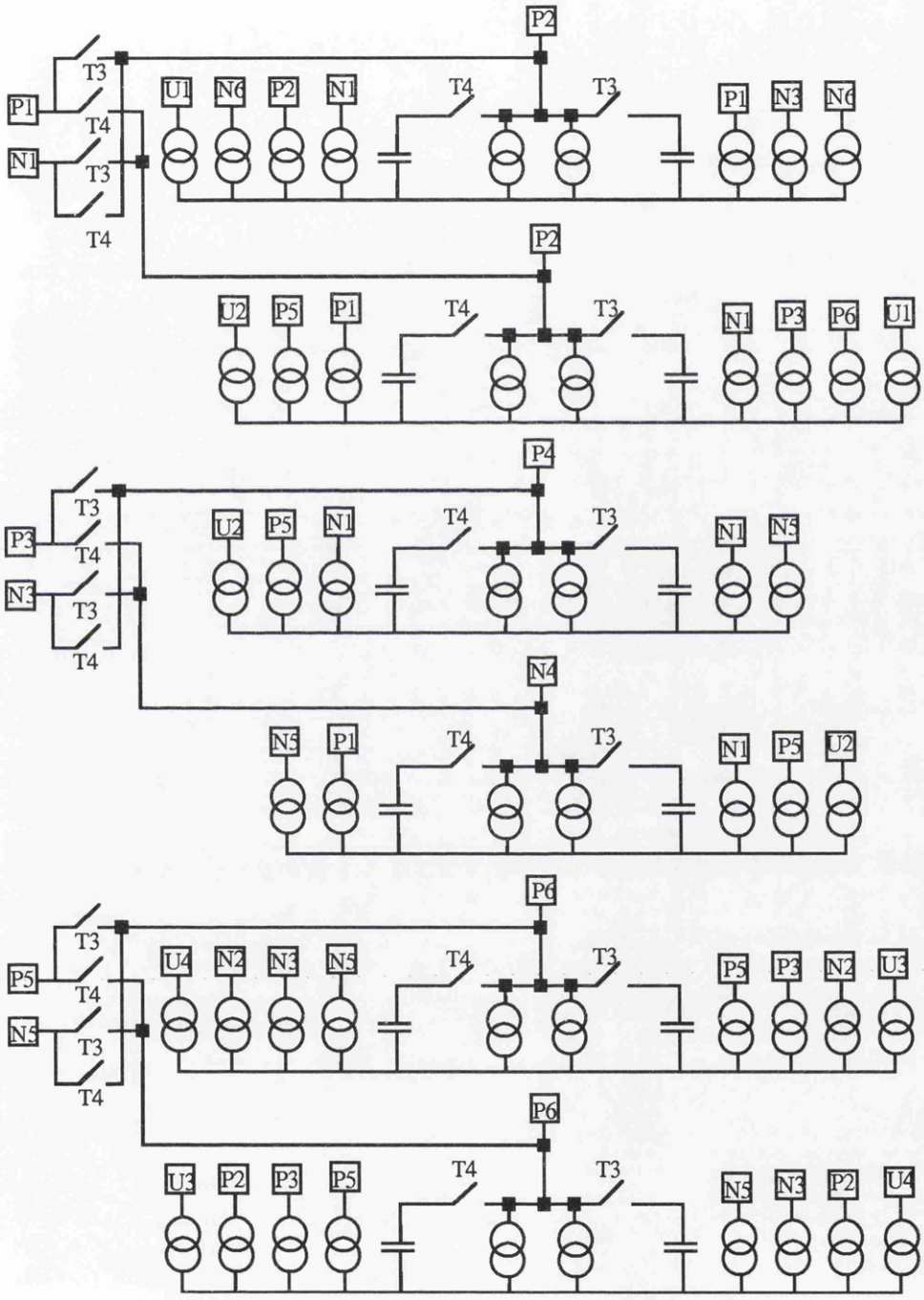


Fig. (3.10) A 3rd-order elliptic ladder SI filter

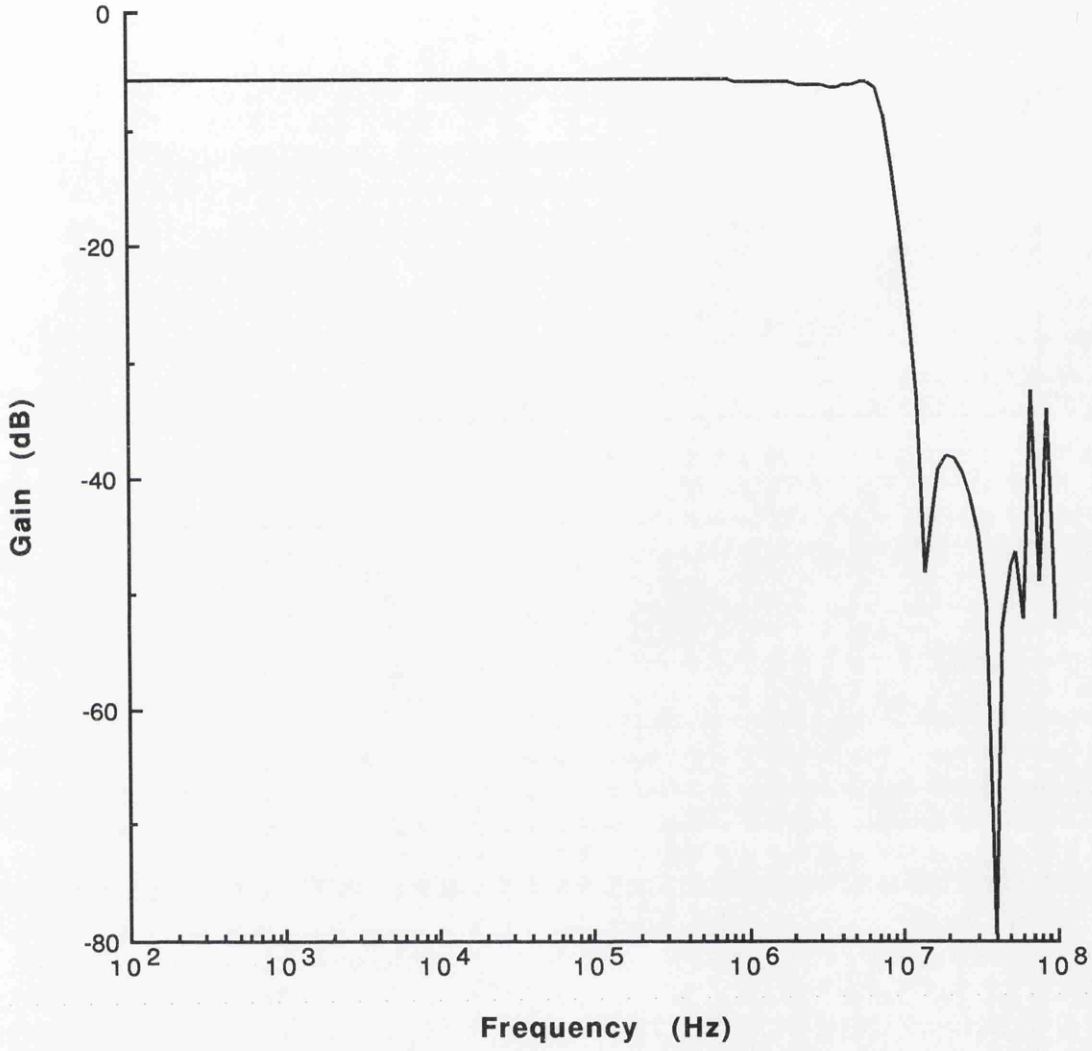


Fig. (3.11a) Frequency response of 3rd-order elliptic ladder SI filter

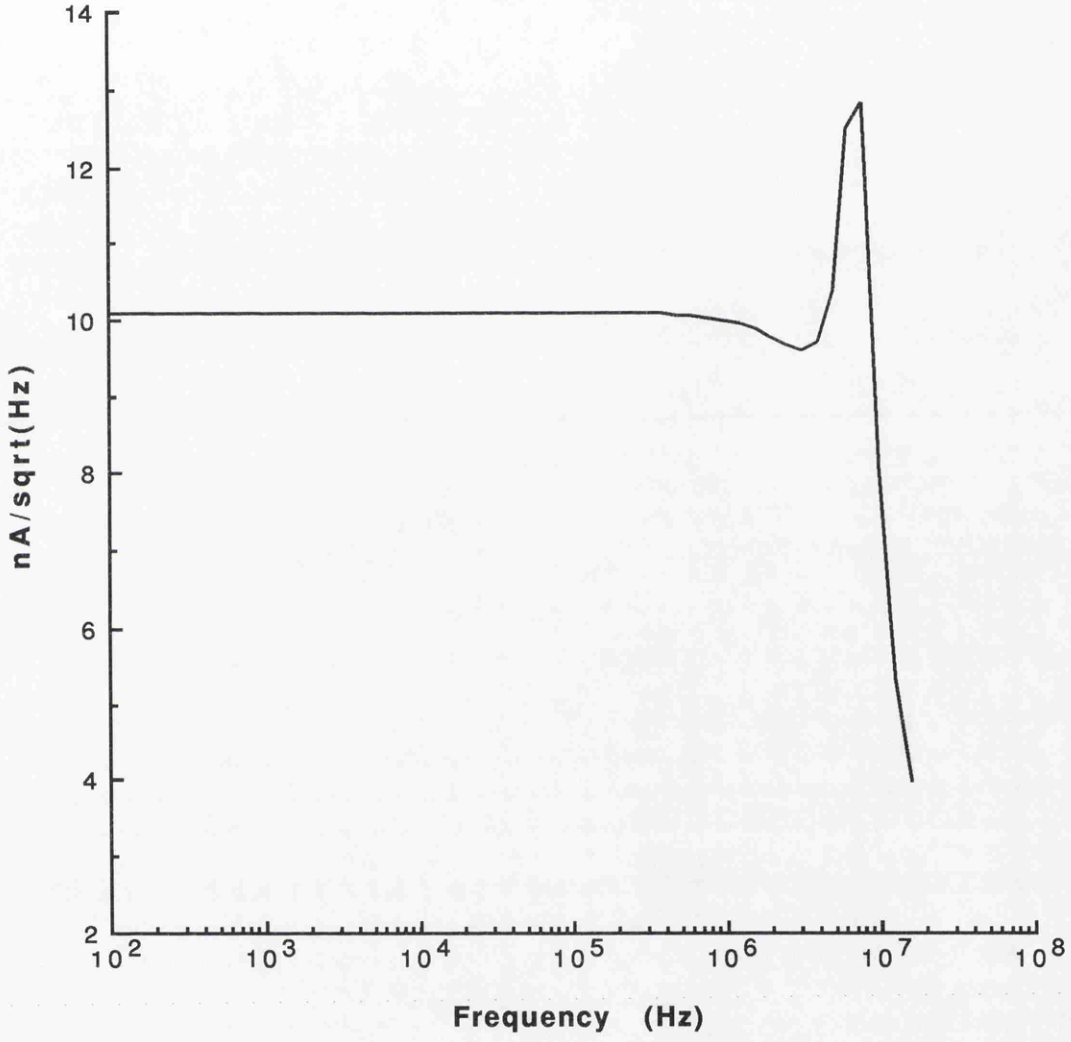


Fig. (3.11b) Noise behaviour of a 3rd-order elliptic ladder SI filter

(c) Continuous-time circuits

Noise analysis for continuous-time circuits is also possible with SCNAP4. A fifth-order elliptic lowpass transconductor-Capacitor video filter [23] illustrated in Fig. (3.12) was examined. The corner frequency is set to 100kHz and the power spectral density of the white noise is given to be $400\text{nV}/\sqrt{\text{Hz}}$. The noise simulation result is shown in Fig. (3.13) and compared well with measurements. It can be observed that the 1/f noise is apparent in lowpass filter. For this cricuit, only 0.41 seconds were required for full noise analysis. It should be pointed out that the determination of the noise source values is essential to the noise analysis. Since there are no such values available from designers, the values of the noise sources are adjusted in computations until a reasonalbe fit between the simulation result and measurement is obtained. Then the resultant noise macromodels are checked with designers for the agreement of the realistic noise source values.

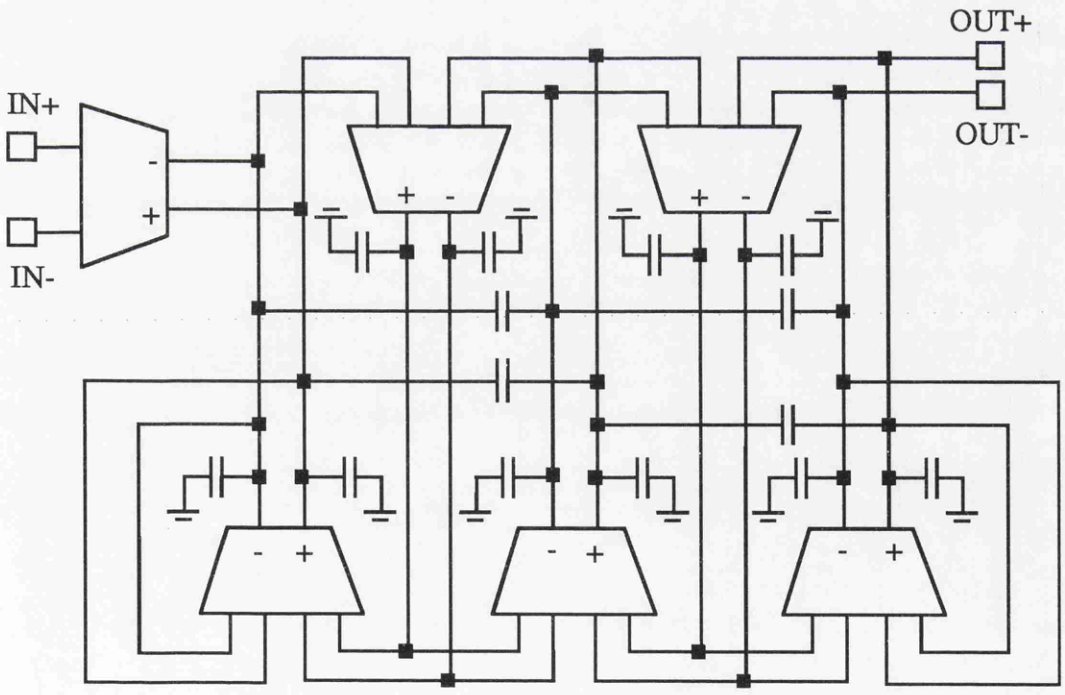


Fig. (3.12) Fully differential 5th-order transconductor-C video filter

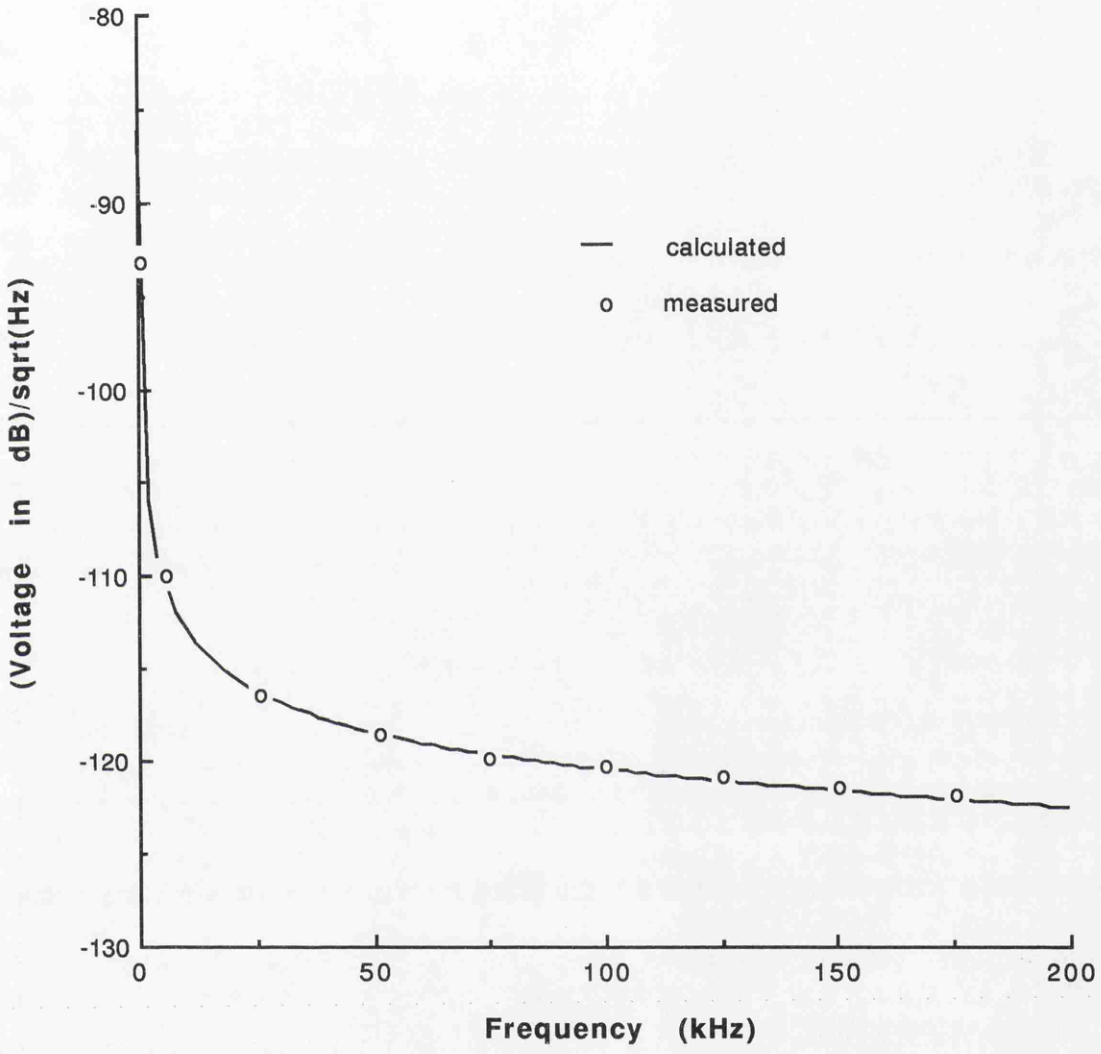


Fig. (3.13) Noise behaviour of 5th-order transconductor-C filter

3.4 SUMMARY

A fully general and efficient noise analysis method is presented for both switched capacitor and switched-current circuits. Noise sources in these circuits are discussed and various noise models are given. Complete computer noise analysis procedures are described. The fold-back effects and nonideal effects such as parasitic time constant and amplifier bandwidth have been taken into account. The thermal noise and flicker noise generated by MOS transistors are considered. By utilising the adjoint network technique, only one system solution is needed for noise analysis. In addition, numerical inversion of the Laplace transformation, the Hessenberg technique, extensive sparse matrix routines and interpretable code generation have been used to improve the efficiency of noise analysis. The method has been implemented in SCNAP4 and numerical results compare well with practical results measured from integrated circuits. Suggestions of using accurate noise models are given. The speed of analysis on a relatively modest work station indicates that noise analysis of large switched systems is feasible.

REFERENCES

[1] J. H. Fischer, "Noise sources and calculation techniques for switched capacitor filters," IEEE J. of Solid-State Circuits, vol. SC-17, No.4, Aug. 1982, pp.742-752

[2] C. A. Gobet and A. Knob, "Noise analysis of switched capacitor networks", IEEE Trans. on CAS, vol. CAS-30, No.1, Jan. 1983, pp.37-43

[3] J. Rabaey, J. Vandewalle and H. De Man, "A general and efficient noise analysis technique for switched capacitor filters", Proc. of IEEE ISCAS, Newport Beach, May. 1983, pp.570-573

[4] S. W. Li, F. W. Stephenson and J. Velazquez-Ramos, "Noise analysis of switched capacitor filters," Proc. of IEEE ISCAS, Montreal, May 1984, pp.1312-1315

[5] C. K. Pun, J.I.Sewell and A. G. Hall, "Noise analysis for switched capacitor networks in symbolic form," Proc. of 29th Midwest symposium on circuits and systems, Louisville, Aug. 1985, pp.807-810

[6] J. Gotte and C. Gobet, "Exact noise analysis of SC circuits and an approximate computer implementation," IEEE Trans. on CAS, vol. CAS-36, No.4, Apr. 1989, pp.508-521

[7] L. Toth and K. Suyama, "Exact noise analysis of 'ideal' SC networks", Proc. of IEEE ISCAS, New Orleans, May 1990, pp.1585-1588

[8] Z. Q. Shang and J. I. Sewell, "Efficient sensitivity analysis for large non-ideal switched capacitor networks," Proc. of IEEE ISCAS, Chicago, May 1993, pp.1405-1407

[9] D. G. Peterson, "Noise performance of transistors," IRE Trans. Electron Devices, vol. ED-4, May 1962, pp. 296-303

[10] A. Van der Ziel, *Fluctuation Phenomena in Semiconductors,* London: Butterworth, 1959

[11] A. G. Jordan and N. A. Jordan, "Theory of noise in metal-oxide semiconductor devices," IEEE Trans. Electron Devices, vol. ED-12, Mar. 1965, pp. 148-156

[12] A. Van der Ziel, "Thermal noise in field-effect transistors," Proc. IRE, vol. 50, Aug. 1962, pp.1808-1812

[13] J. C. Bertalls, "Low frequency noise considerations for MOS amplifier design," IEEE Journal of Solid-State Circuits, vol. SC-14, Aug. 1979, pp.773-776

[14] J. A. Barby, "Switched-current filters models for frequency analysis in the continuous-time domain," Proc. of IEEE ISCAS, Chicago, May 1993, pp.1427-1430

[15] C. Toumazou, J. B. Hughes and N. C. Battersby, *SWITCHED-CURRENTS an analogue technique for digital technology*, Peter Peregrinus Ltd. 1993, (pp.121-122)

[16] L. B. Wolovitz and J. I. Sewell, "General analysis of large linear switched capacitor networks," IEE Proc. vol. 135, pt.G, no. 3, Jun. 1988, pp.119-124

[17] A. Papoulis, *Probability, Random Variables and Stochastic Process,* New York: McGraw-Hill, 1965

[18] M. L. Liou and Y. L. Kuo, "Exact analysis of switched capacitor circuits with arbitrary inputs," IEEE Trans. on Circuits and Syst., vol. CAS-26, no. 4, Apr. 1979, pp. 213-223

[19] Z. Q. Shang and J. I. Sewell, "Efficient noise analysis methods for large non-ideal SC and SI circuits," Proc. IEEE ISCAS, London, Jun. 1994, pp.5.565-5.568

[20] J. B. Hughes, "Analogue techniques for large scale integrated circuits," Ph.D Thesis, University of Southampton, March 1992

[21] G. Wegmann, "Design and analysis techniques for dynamic current mirrors," Thesis No. 890, Ecole Polythechnique Federale de Lausanne, 1990

[22] S. J. Daubert and D. Vallancourt," Operation and analysis of current copier circuits," Proc. IEE Pt. G, vol. 137, April 1990, pp.109-115

[23] Y. Lu, N. P. J. Greer, and J. I. Sewell, "A transconductor-capacitor video filter and equaliser design," Proc. of IEEE ISCAS, Chicago, May 1993, pp.986-989

CHAPTER 4

SYMBOLIC ANALYSIS

4.1 INTRODUCTION

The symbolic analysis of linear analogue circuits has attracted the attention of many researchers over a considerable period of time. Many algorithms and techniques have been developed. Among them, the matrix and determinant method and the signal-flow graph method appear to be favoured in terms of flexibility and efficiency [1-15]. All approaches suffer from restrictions inherent to the problem, the escalation of computer time and memory requirements with increase in circuit size. Two typical solutions proposed are expression approximation[16] and hierarchical decomposition [17], some improvements have resulted. However, despite all the effort, these methods still experience great difficulty with medium to large networks. Whilst attention has been focused on continuous-time analogue circuits, some work has been carried out on the symbolic analysis of ideal SC networks[18] and only exploratory investigations have been reported into the semi-symbolic analysis of nonideal SC networks[19] and noise performance [20].

It has been realised that the only practical scheme for large networks of any type is a semi-symbolic one, when polynomials in s or z or both, with purely numeric coefficients, are generated. It is well known that generation of polynomials by interpolation is a very efficient technique. However, it has only been applied to continuous-time and ideal SC networks. The application of symbolic analysis to large nonideal switched linear networks still remains an open research area.

This Chapter is concerned with efficient symbolic analysis methods for large nonideal switched linear networks. Some general symbolic analysis methods are reviewed and general comparison of the different approaches is given. The application of polynomial interpolation method for large nonideal switched linear networks is then presented. The accuracy issues of the interpolation method are also discussed. The extension to noise computation in switched linear networks is considered. Further investigation on the comparison of the computation costs between symbolic expression evaluation and numerical matrix solution has been made and results are given.

4.2 GENERAL SYMBOLIC ANALYSIS METHODS

Symbolic analysis of linear networks involves yielding the network transfer functions in the form

$$H(\mathbf{x},\mathbf{p}) = \frac{N(\mathbf{x},\mathbf{p})}{D(\mathbf{x},\mathbf{p})}$$

where $N(\mathbf{x},\mathbf{p})$ and $D(\mathbf{x},\mathbf{p})$ are polynomials in complex frequency variable x (x could be s for continuous-time and z for discrete-time circuits or a combination of both) and the symbolic network variables **p**. Depending on the problem to be solved, **p** could be either all the parameters in a network or only part of them. In an extreme case, **p** entries are represented by their numeric values and the frequency variable (s or z) is the only symbol.

The objective of the following sections is to seek an appropriate symbolic analysis method for large nonideal switched linear networks. A brief survey of the general symbolic analysis methods is presented next.

4.2.1 Matrix -based methods

By performing a series of matrix manipulations on the matrix of the symbolic network equations, the determinant can be generated symbolically. The matrix manipulation methods depend on the problem to be solved. If the whole response vector is required, elimination methods are applied. However this is a rare demand, in most cases, only one output variable is of interest. Then the Cramer's rule can be applied. Several algorithms can be used to calculate a symbolic determinant [21]. For example, a determinant can be calculated by means of a recursive expansion formula [22] which is free of additive and multiplicative cancellations for general matrix entries.

The Laplace expansion of the determinant of a $n \times n$ matrix along an arbitrary row i is given by

$$\det(M) = \sum_{j=1}^{n} m_{ij}(-1)^{i+j} \det(M_{ij})$$

where M is the matrix of the symbolic network equations, $\det(M_{ij})$ is the minor obtained by removing row i and column j of the original matrix M. m_{ij} is the entry of the ith row and the jth column.

It is evident that the determinant can be recursively expended along multiple rows or columns simultaneously.

Computer programs based on these methods have been developed[1-2]. Various applications have been reported and shown some success over a certain range of circuits. However, the matrix and determinant methods require that the network equations are fully expressed with symbolic coefficients.

4.2.2 Graph-based methods

(a) Signal-flow graph method

The so called signal-flow graph is a weighted, directed graph with variables as nodes and the circuit elements as branch weights [23]. The transfer function from an input variable x_i to an output variable x_j can be generated from the structure of the signal-flow graph by means of Mason's rule [24]

$$\frac{x_j}{x_i} = \frac{\sum P_k \Delta_k}{\Delta}$$

where

$\Delta = 1 - \Sigma$(1st order loop weights)
$+ \Sigma$(2nd order loop weights)
$- \Sigma$(3rd order loop weights)
$+ \cdots$

P_k = weight of the kth path from the input node x_i to the output node x_j

$\Delta_k = \Sigma$(terms in Δ without constituent loops touching P_k)

Here the summation is carried out over all paths from x_i to x_j. A path (or loop) weight means the product of all branch weights along the path (or loop). Detailed algorithms can be found in [25].

It also can be noticed that to construct the signal-flow graph of the circuit, all branch weights are explicitly related to the circuit elements according to corresponding circuit equations.

(b) Tree-enumeration method

The circuit equations are expressed in the form of the indefinite admittance matrix (IAM) from which the property that the sum of all entries in each column and row is zero, can be utilised. The IAM equations are then represented by a directed, weighted graph. Having this representation, the determinant of the system matrix is equal to the sum of the branch-weight products of all directed trees in the directed graph of the circuit. Therefore, the computation of a network function implies the enumeration of all direct trees in the directed graph. Detailed theory of the tree enumeration method can be found in [25].

However, the method is inherently only applicable to RLC-g_m networks. In addition, it suffers from term cancellations so that the size of the circuit is also limited.

4.2.3 Parameter-extraction method

This method takes the advantages of combining numerical and topological techniques and efficiently takes care of the case where a relatively small number of the network elements are represented by symbols. The circuit equations have to be formulated in the form of indefinite admittance matrix (IAM). The process [6] of symbolic parameter extraction from the matrix determinant can be accomplished by splitting up the determinant into two new determinants which exclude the symbolic parameter of interest. The strategy can be carried out recursively when more symbolic parameters are required to be extracted.

Generally, if only few parameters are of interest, it is possible to handle relatively large circuit. As more and more elements left as symbols, the process will have the same network size limitations as experienced by matrix-based or topological oriented methods. Further more, the method strongly depends on the mode of matrix formulation and parameter to be extracted must be expressed explicitly in the indefinite admittance matrix.

4.2.4 Interpolation method

It is well known that a polynomial f(x) of degree n can be uniquely defined by the n+1 distinct values of $f(x_i)$ and can be recovered by interpolating these values. The process is called polynomial interpolation. It was first extended to calculate symbolic network functions in [8]. For the interpolation method, the frequency variable is the

only symbol left in the network functions. By using the FFT algorithm, it is expected that much large circuit can be handled than by the alternative methods above.

4.2.5 Overview

The symbolic analysis methods above are now examined for suitability for the challenge of large nonideal switched linear networks. The matrix and graph-based methods are very appropriate for full symbolic analysis, where all network elements are represented by symbols. Both the determinant and signal - flow graph methods are generally more applicable and efficient due to their cancellation-free property. The tree-enumeration method is found not applicable to general networks. Besides, term cancellations make the method inferior. The parameter-extraction method takes another strategy based on the fact that only a subset network parameters are of interest to designers, hence improvement in efficiency results. However, all the methods described above require that the network parameters should have an explicit relation with network equations. Unfortunately, the prescribed requirement cannot be satisfied in nonideal switched linear networks where the system equations are constructed from parameters created through approximation techniques. An interpolation method, on the other hand, requires only the computation of the sampled response to recover the network functions and is applicable to any type of circuit equations. As the frequency is the unique symbol in a network function, it provides less insight into the network behaviour than fully symbolic analysis.

4.3 INTERPOLATION METHODS

Consider a *m*th order polynomial

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_m x^m \tag{4.1}$$

subject to m+1 distinct samples of the variable x, x_i will enable the coefficient a_i to be determined from

$$\begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_m) \end{bmatrix} = \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^m \\ 1 & x_1 & x_1^2 & \cdots & x_1^m \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & x_m & x_m^2 & \cdots & x_m^m \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix} \qquad (4.2a)$$

$$\mathbf{F} \qquad\qquad \sigma \qquad\qquad \mathbf{A}$$

or $$\mathbf{A} = \sigma^{-1}\mathbf{F} \qquad (4.2b)$$

where **A** is the vector of unknown coefficients, **F** denotes the vector of sampled responses and σ represents the (m+1)×(m+1) Vandermonde matrix of samples.

The unknown coefficients can be obtained by solving equation (4.2). There are many techniques that can be used to compute this solution and there generally come under the heading of polynomial interpolation.

4.3.1 Accuracy issues

It has been experienced that when m increases, the error introduced into the coefficients generated by interpolation becomes significant. Examination of equation (4.2) reveals that the error could come from two sources, the evaluation of **F**, and in the formation of σ^{-1}. In the following, techniques for improving the accuracy of the coefficients of network polynomials produced by the interpolation method will be discussed in detail.

(a) Partitioned polynomial interpolation

The objective is to reduce the dimension of the individual matrices involved in the determination of the coefficients such that the anticipated error incurred during the process can also be reduced.

Divide equation (4.2a) and the coefficients into p sets of equal number h, where h is equal to $(m+1)/p$. Referring to [27], (4.2a) yields

$$\begin{bmatrix} \mathbf{A}_0 \\ \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_{p-1} \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{D}_0^1 & \cdots & \mathbf{D}_0^{p-1} \\ 1 & \mathbf{D}_1^1 & \cdots & \mathbf{D}_1^{p-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \mathbf{D}_{p-1}^1 & \cdots & \mathbf{D}_{p-1}^{p-1} \end{bmatrix}^{-1} \begin{bmatrix} \sigma_0^{-1}\mathbf{F}_0 \\ \sigma_1^{-1}\mathbf{F}_1 \\ \vdots \\ \sigma_{p-1}^{-1}\mathbf{F}_{p-1} \end{bmatrix} \qquad (4.3)$$

$$\mathbf{A} \qquad\qquad \mathbf{D}^{-1} \qquad\qquad \Sigma^{-1}\mathbf{F}$$

where $\mathbf{A}_i$ and $\mathbf{F}_i$ are subset vectors of coefficients and responses, σ_i represents the Vandermonde matrix related to $\mathbf{A}_i$ and $\mathbf{F}_i$. $\mathbf{D}_i$ is a diagonal matrix in which entries are subset samples.

The inversion of **D** matrix presents few problems since its order is equal to the number of partitions which is usually quite small, and as it is in the form of Vandermonde matrix, a Traub[26] technique can be used. Observe that $\sigma_i^{-1}\mathbf{F}_i$ are similar to that in equation (4.2b) but with much smaller size. The inversion of a large matrix in (4.2b) now becomes a process of inverting a series of relatively small matrices.

(b) Selection of sample points

In the computer implementation of interpolation method, the selection of sample points also controls the accuracy of inversion of the Vandermonde matrix. As the order of the polynomials increases, the choice of sensible sample points is crucial. Originally, numerical interpolation has been carried out by selecting sample points on the real axis, so that only real arithmetic is employed throughout. The scheme, though efficient, suffers from numerical instability in terms of ill-condition matrices[28]. The error incurred is found to be quite sensitive to the locations of the sample frequencies on the real axis. The complex conjugate pair interpolation formula was then suggested in [8] to improve both accuracy and efficiency of the method. The strategy was further generalised by selecting sample points uniformly distributed on the unit circle in the complex plane[29-30]. In [18], the transformation of the complex plane to improve accuracy was performed.

Let the points uniformly distributed on the unit circle

$$x_k = e^{j\frac{2k\pi}{m+1}} \qquad (k = 0, 1, ..., m) \tag{4.4}$$

The polynomial (4.1) can now be written as

$$f(x_i) = \sum_{k=0}^{m} a_k e^{j\frac{2\pi ik}{m+1}} \qquad (i = 0, 1, ..., m) \tag{4.5}$$

Notice that, equation (4.5) is identical to the standard form of discrete Fourier transform (DFT). Hence, the coefficients can be obtained from

$$a_i = \frac{1}{m+1}\sum_{k=0}^{m} f(x_k) e^{-j\frac{2\pi ik}{m+1}} \qquad (i = 0, 1, ..., m) \tag{4.6}$$

Instead of matrix inversion, only a discrete Fourier transform is required to generate all coefficients of the polynomial. Since extensive effort has been devoted in providing extremely fast and accurate FFT routines, the polynomial can be recovered accurately and efficiently.

(c) Evaluation of sampled response

Techniques for reducing the error incurred due to the formation of σ^{-1} have been discussed. It has been found that for discrete-time circuits, the proper evaluation of the sampled responses is particularly important. It will be further discussed in the next section.

4.3.2 Computer algorithms

Throughout the Chapter, attention is focused on large nonideal switched linear networks. There can be described by a system of equations in z domain as presented in previous Chapters

$$\underbrace{\begin{bmatrix} \mathbf{I} & & & -\mathbf{P}_1 \\ -\mathbf{P}_2 & \mathbf{I} & & \\ & \ddots & \ddots & \\ & & -\mathbf{P}_N & z\mathbf{I} \end{bmatrix}}_{\mathbf{M}} \underbrace{\begin{bmatrix} \mathbf{V}_1(z) \\ \mathbf{V}_2(z) \\ \vdots \\ \mathbf{V}_N(z) \end{bmatrix}}_{\mathbf{X}} = \underbrace{\begin{bmatrix} \mathbf{W}_1(z) \\ \mathbf{W}_2(z) \\ \vdots \\ \mathbf{W}_N(z) \end{bmatrix}}_{\mathbf{W}} \tag{4.7}$$

where

$$\mathbf{P}_k = \mathbf{p}_k \mathbf{C}_k$$

$\mathbf{C}_k$ is the capacitance matrix of the kth time-slot and $\mathbf{p}_k$ is the extended state transition matrix. The right hand side contains the excitation vectors and appropriate sampling weighting factors, the vector $\mathbf{V}_k$ contains node voltages and branch currents.

Since the $\mathbf{p}_k$ matrices and components of the right hand side vector can only be assembled by numerical approximation techniques, **fully symbolic solutions for the system responses are impossible to obtain**. However, semi-symbolic solutions with the frequency as the only symbol is feasible. As has been discussed in section (4.2.5), a polynomial interpolation scheme provides such a possibility.

The general computer algorithm for the interpolation method can be summarised as follows:

for (i = 0; i < m; i++) { / m is polynomial degree */*

$z_i = e^{j\frac{2\pi i}{m+1}}$;

$\boldsymbol{X} = \boldsymbol{M}^{-1}(z_i)\boldsymbol{W}$;

$D(z_i) = det\boldsymbol{M}(z_i)$;

$N(z_i) = D(z_i)\boldsymbol{d}^t\boldsymbol{X}$;

}

DFT (FFT) to generate the numerator polynomial;

DFT(FFT) to generate the denominator polynomial;

For continuous-time systems, the above procedures are straightforward. However, for a discrete-time system like (4.7), in order to obtain the determinant of the system matrix **M**, further derivations are required.

Apply a block Gaussian elimination to equation (4.7) to give

$$\begin{bmatrix} \mathbf{I} & & & -\mathbf{P}_1 \\ & \mathbf{I} & & -\mathbf{P}_2\mathbf{P}_1 \\ & & \ddots & \vdots \\ & & & z\mathbf{I}-\mathbf{E} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1(z) \\ \mathbf{V}_2(z) \\ \vdots \\ \mathbf{V}_N(z) \end{bmatrix} = \begin{bmatrix} \mathbf{W}_1(z) \\ \sum_{k=1}^{2} \mathbf{g}_k^2 \mathbf{W}_2(z) \\ \vdots \\ \sum_{k=1}^{N} \mathbf{g}_k^N \mathbf{W}_N(z) \end{bmatrix} \tag{4.8}$$

where

$$\mathbf{E} = \mathbf{P}_N\mathbf{P}_{N-1}\cdots\mathbf{P}_1$$

$$\mathbf{g}_k^i = \begin{cases} \mathbf{P}_i\mathbf{P}_{i-1}\cdots\mathbf{P}_{k+1} & k < i \\ \mathbf{I} & k = i \end{cases}$$

The system matrix $\mathbf{M}$ has been reduced to the upper block triangular form. Therefore, the denominator of the transfer function can be determined by

$$\begin{aligned} D(z_i) &= \det \mathbf{M}(z_i) \\ &= \left[\prod_{k=1}^{N-1} \det \mathbf{I}\right] \det(z_i \mathbf{I} - \mathbf{E}) \\ &= \det(z_i \mathbf{I} - \mathbf{E}) \end{aligned}$$

The numerator polynomial then follows in a simple mannar.

However, as has been mentioned early in section (4.3.1), in the case of switched network, some care has to be taken for the evaluation of sampled responses, otherwise considerable error may be introduced. The loss of accuracy may be demonstrated by means of a simple 2-phase clock system.

The system equation can be written as

$$\begin{bmatrix} \mathbf{I} & -\mathbf{P}_1 \\ -\mathbf{P}_2 & z\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{H}_1(z) \\ \mathbf{H}_2(z) \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1 e^{j\omega t_1} \\ \mathbf{B}_2 e^{j\omega t_2} \end{bmatrix} \tag{4.9}$$

solving the matrix equation (4.9) gives

$$\begin{bmatrix} \mathbf{H}_1(z) \\ \mathbf{H}_2(z) \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1 e^{j\omega t_1} + \dfrac{\mathbf{P}_1 \mathbf{B}_2 e^{j\omega t_2} + \mathbf{P}_1 \mathbf{P}_2 \mathbf{B}_1 e^{j\omega t_1}}{z\mathbf{I} - \mathbf{P}_2 \mathbf{P}_1} \\ \dfrac{\mathbf{B}_2 e^{j\omega t_2} + \mathbf{P}_2 \mathbf{B}_1 e^{j\omega t_1}}{z\mathbf{I} - \mathbf{P}_2 \mathbf{P}_1} \end{bmatrix} \tag{4.10}$$

The denominator polynomial is obtained from

$$D(z) = \det(z\mathbf{I} - \mathbf{P}_2 \mathbf{P}_1) \tag{4.11}$$

and the numerators are determined by

$$\begin{pmatrix} N_1(z) \\ N_2(z) \end{pmatrix} = D(z) \begin{pmatrix} H_1(z) \\ H_2(z) \end{pmatrix} \tag{4.12}$$

The numerator and denominator polynomials are then generated by interpolation using a DFT (FFT). Notice that the steps illustrated above exactly follow the general interpolation procedures.

When the frequency response is evaluated from the resultant polynomials and compared with directly computed results, the error is quite considerable, even for small circuits.

For a small SC treble tone control filter shown in Fig. (4.1), the error between a directly computed frequency response and one from interpolated polynomial for one transfer functions $H_2(z)$ is illustrated in Fig. (4.2a).

The poor accuracy is due to fractional powers of z which are introduced by the time-slot weighting factors. If the polynomials are re-defined to account for this, very high orders result and low accuracy follows. There is a simple solution. Consider again the 2-phase system and separately excite each time-slot with unit excitation:

$$\begin{bmatrix} \mathbf{I} & -\mathbf{P}_1 \\ -\mathbf{P}_2 & z\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{H}_{11}(z) \\ \mathbf{H}_{21}(z) \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{0} \end{bmatrix} \tag{4.13a}$$

$$\begin{bmatrix} \mathbf{I} & -\mathbf{P}_1 \\ -\mathbf{P}_2 & z\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{H}_{12}(z) \\ \mathbf{H}_{22}(z) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_2 \end{bmatrix} \tag{4.13b}$$

Solving (4.13a) and (4.13b) for H_{11}, H_{21}, H_{12} and H_{22}, respectively

$$\mathbf{H}_{11}(z) = \mathbf{B}_1 + \frac{\mathbf{P}_1\mathbf{P}_2\mathbf{B}_1}{z\mathbf{I} - \mathbf{P}_2\mathbf{P}_1} \tag{4.14a}$$

$$\mathbf{H}_{21}(z) = \frac{\mathbf{P}_2\mathbf{B}_1}{z\mathbf{I} - \mathbf{P}_2\mathbf{P}_1} \tag{4.14b}$$

$$\mathbf{H}_{12}(z) = \frac{\mathbf{P}_1\mathbf{B}_2}{z\mathbf{I} - \mathbf{P}_2\mathbf{P}_1} \tag{4.14c}$$

$$\mathbf{H}_{22}(z) = \frac{\mathbf{B}_2}{z\mathbf{I} - \mathbf{P}_2\mathbf{P}_1} \tag{4.14d}$$

Referring to (4.10), the following relation can be obtained

$$\begin{pmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{pmatrix} \begin{pmatrix} e^{j\omega t_1} \\ e^{j\omega t_2} \end{pmatrix} \tag{4.15}$$

The error resulting from evaluation of $H_2(z)$ interpolated polynomials is given in Fig.(4.2b) and shows a great improvement in accuracy.

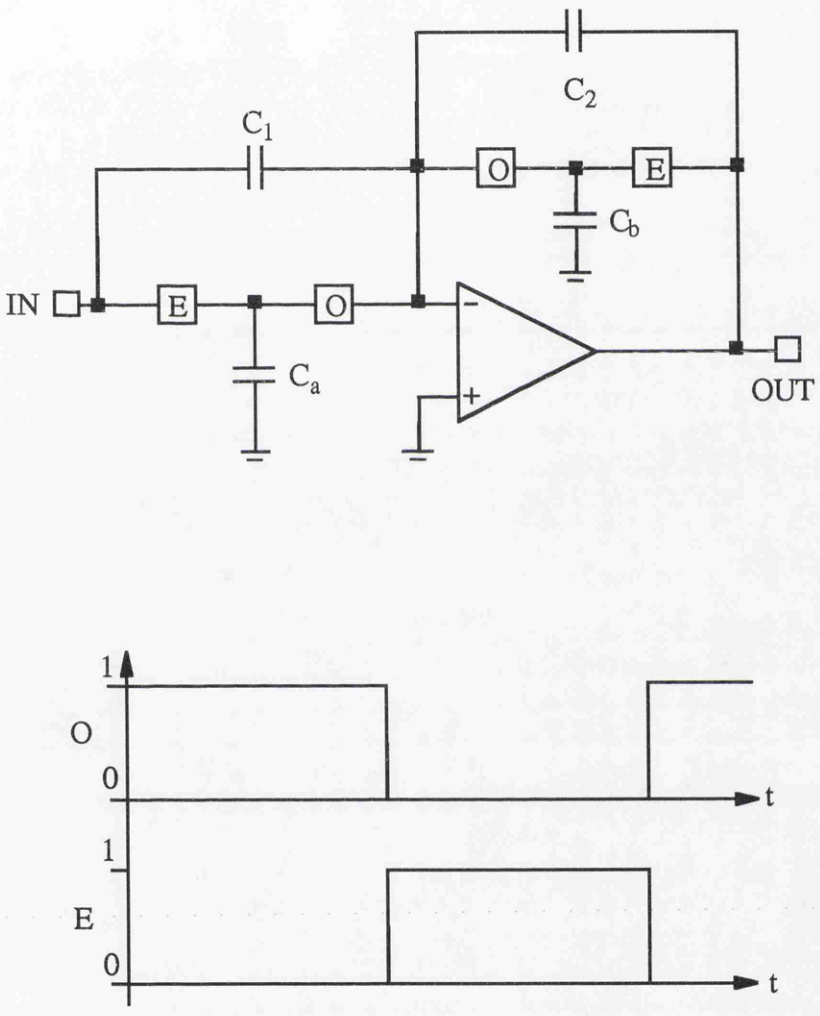


Fig. (4.1) A treble tone control SC filter
circuit diagram and clock waveforms

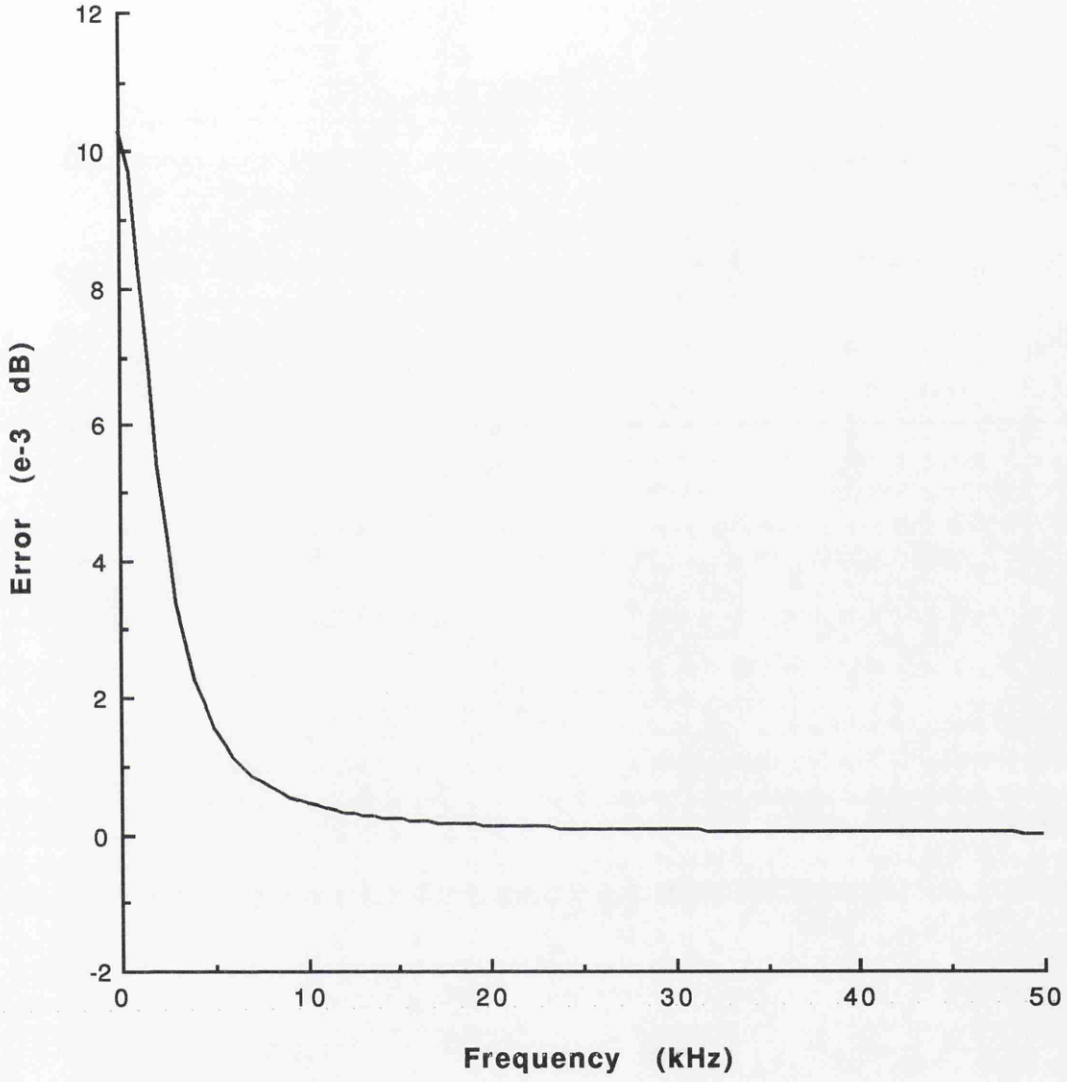


Fig. (4.1a) Comparative error response for a direct interpolation method

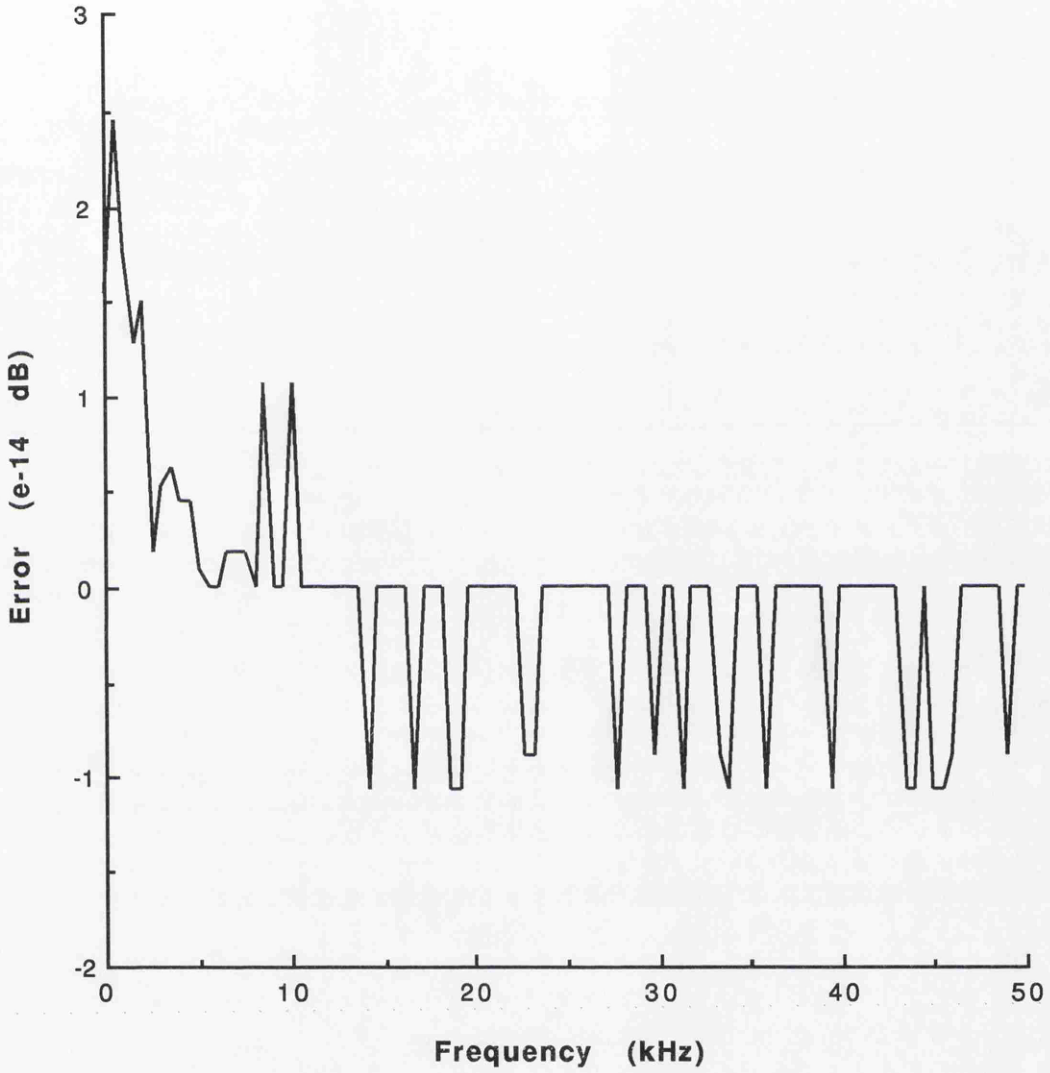


Fig. (4.1b) Comparative error response for a modified interpolation method

4.3.3 APPLICATIONS

The above strategies were implemented on computer. Examples are now given to demonstrate some interesting applications.

(a) Transfer function evaluation

Once the symbolic transfer functions have been generated, the evaluation of such analytical expressions over wide frequency range becomes more efficient than numerical solutions which have to perform a full AC analysis at each frequency point.

A fifth order elliptic lowpass filter [31] will be used to demonstrate the application of symbolic analysis. The network is shown in Fig. (4.3a). Symbolic analysis of the network results in a 4×4 matrix of transfer functions which relate input in each time slot to output in each time slot. Actually, the input is applied to the network only in phase 1 and phase 3, resulting in H_{2k} and H_{4k} (k = 1, 2, 3, 4) with very small but non-zero coefficients. It can be interpreted that due to the finite switch off resistance, even in phase 2 and phase 4, the input signal still has very weak impact on the network.

Part of the matrix of numerators and coefficients of denominator are given in Table (4.1). The circuit response evaluated from the symbolic function and associated error are shown in Fig. (4.3b).

Table (4.1) Coefficients of the numerator and denominator

power in z	Numerator coefficients				Denominator coefficients
	N_{11}	N_{21}	N_{31}	N_{41}	
0	4.715521e-3	2.218036e-9	1.391175e-2	4.325436e-9	-0.321373
1	6.526007e-4	-4.018590e-9	-2.881133e-2	-9.532969e-9	1.751595
2	-5.613027e-3	6.893035e-9	4.323491e-2	1.344378e-9	-4.051438
3	1.713348e-2	-5.384095e-9	-3.067823e-2	-9.045607e-9	5.020159
4	-1.227348e-2	3.272793e-9	1.527681e-2	3.912781e-9	-3.347205
5	8.318768e-3	0	0	0	1.0

For a large network example, a frequency translated switched capacitor filter with 24 clock phases and the system size is 2,688 is selected. Here the system size stands for the dimension of the system matrix. The circuit contains 14 opamps, 61 capacitors, 80 switches, 5 inverters and 4 unity-gain buffers and was designed by XFILT[32].

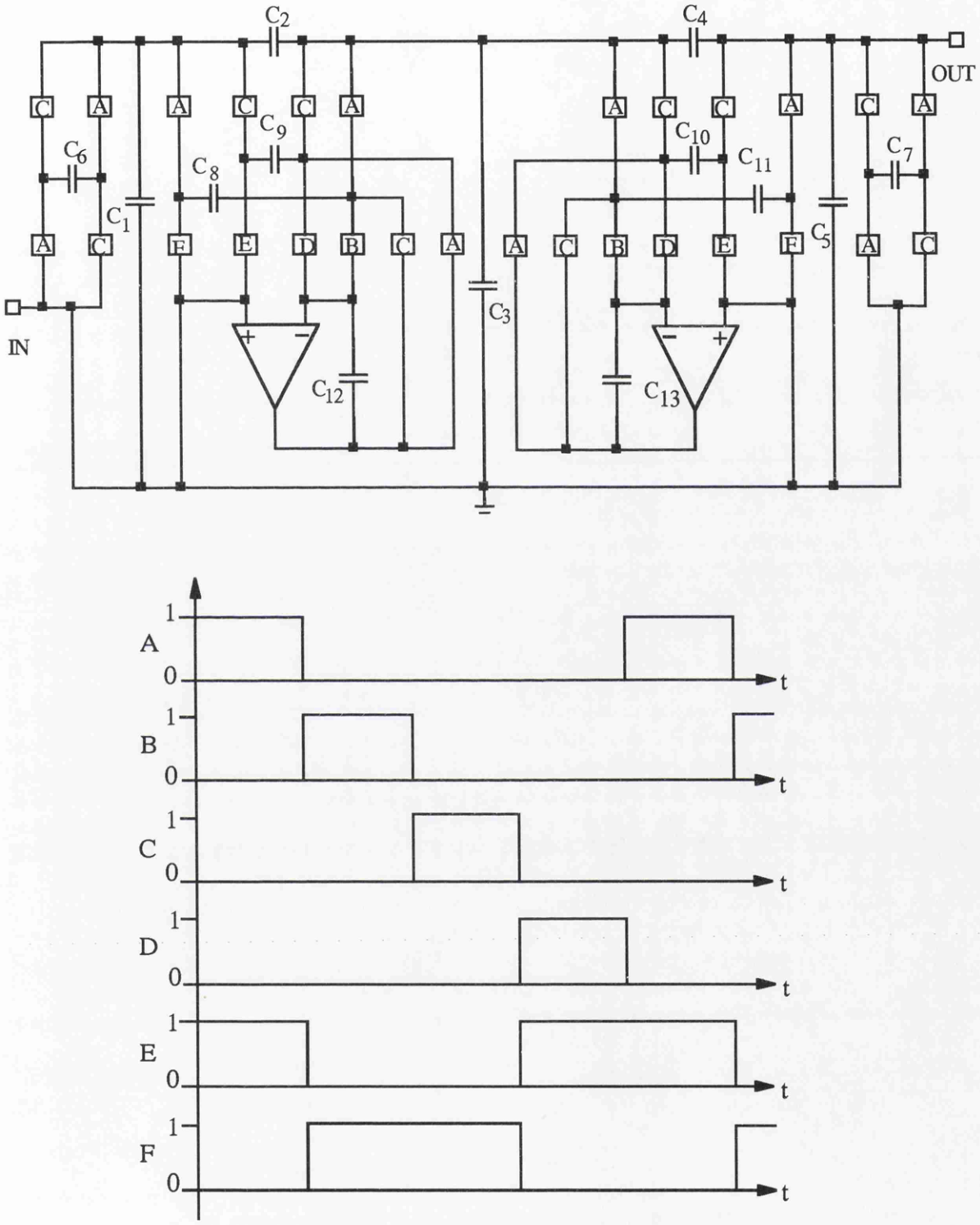


Fig.(4.3a) A 5th order elliptic lowpass filter
circuit diagram and clock waveforms

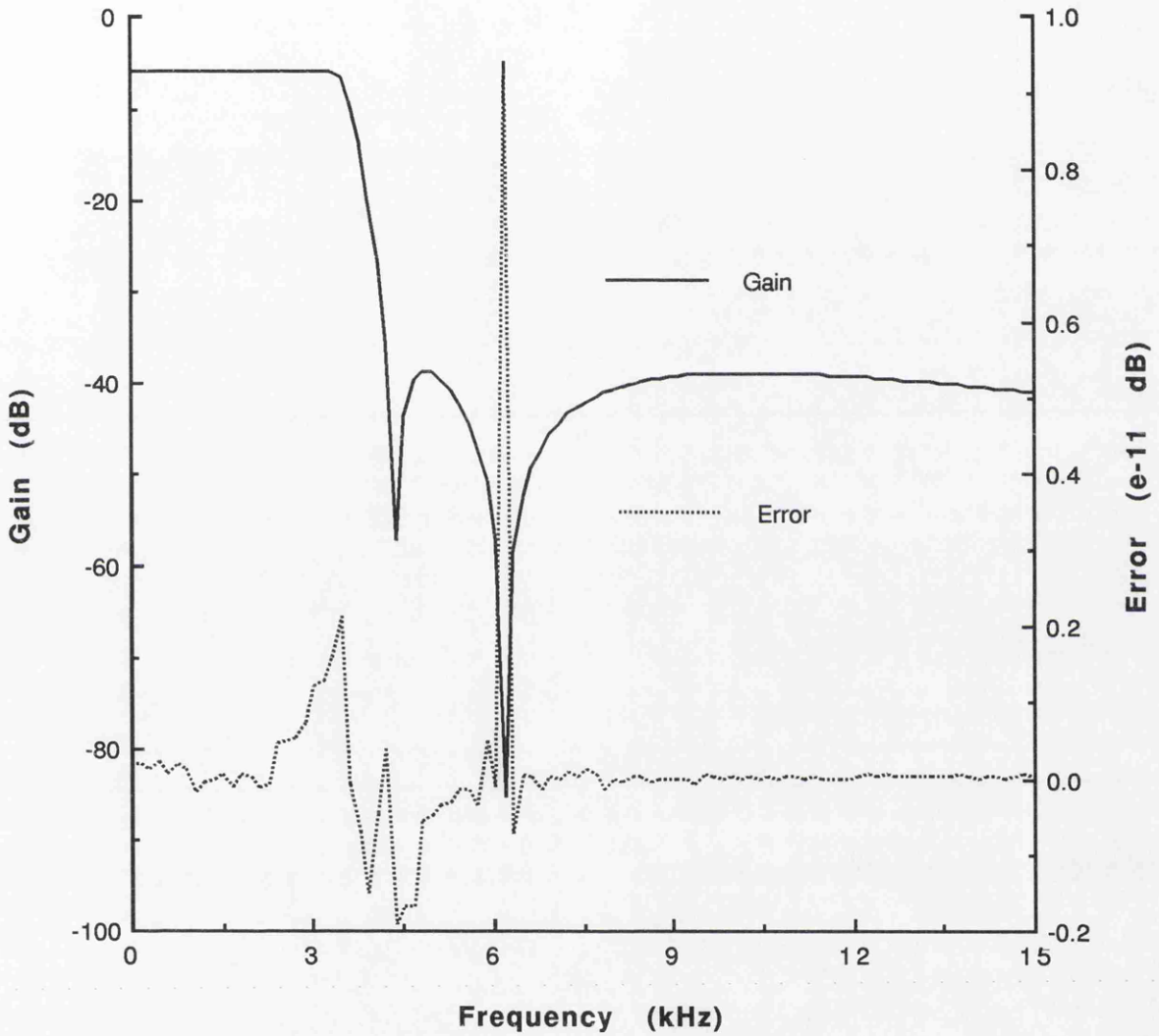


Fig. (4.3b) Frequency response and interpolation error of a 5th order elliptic lowpass filter

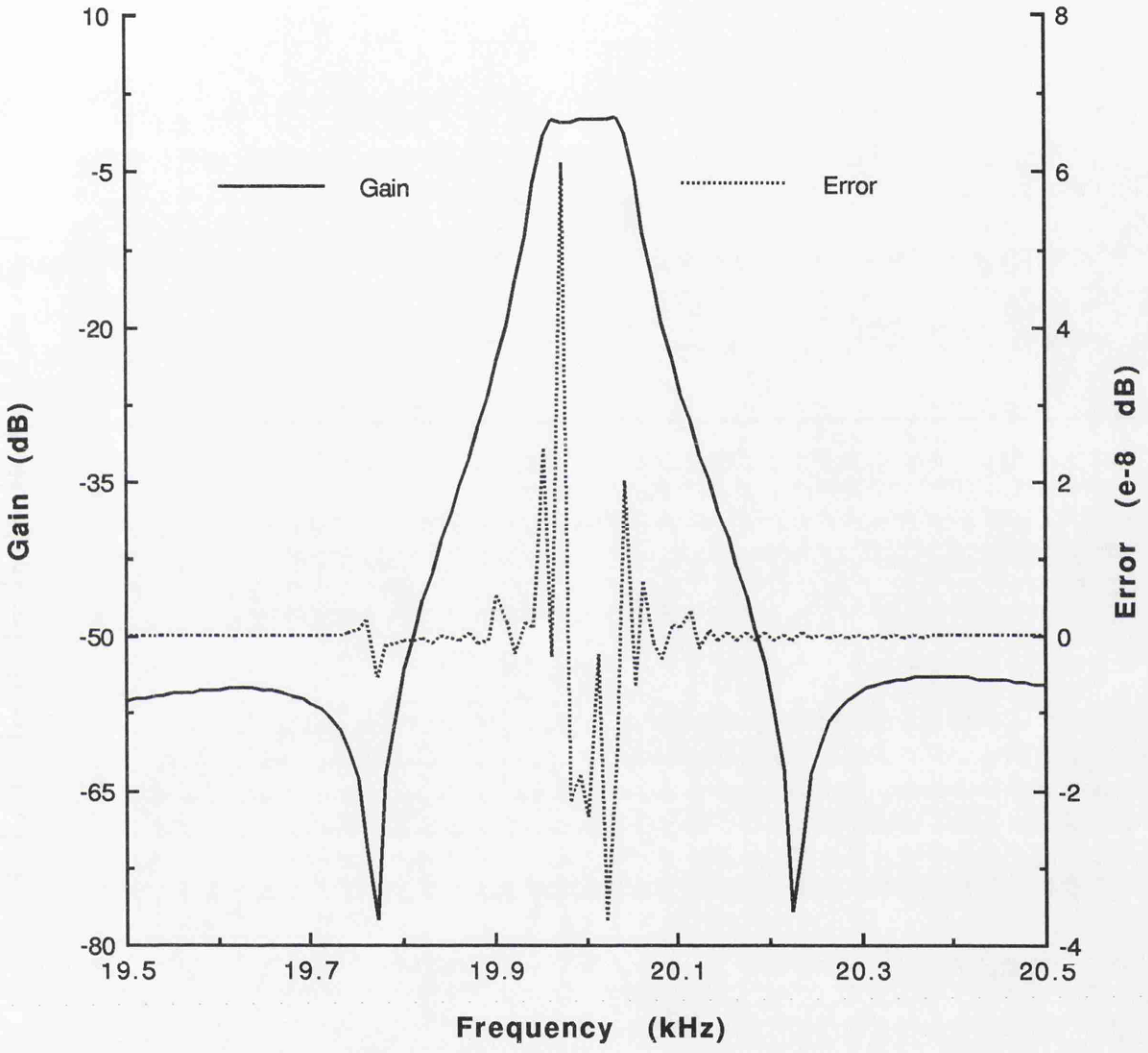


Fig. (4.4) Frequency response and interpolation error of a frequency translated SC filter

The symbolic analysis result and error response are illustrated in Fig. (4.4). It can been seen that even for such a big circuit, the accuracy of symbolic analysis was preserved fairy well. It is therefore not necessary to resort to accuracy improvement routines such as partitioned polynomial interpolation [27] or complex plane transformations [18].

A wide range of SC circuits were tested and the run time statistics are illustrated in Table (4.2).

Table (4.2). Statistics of polynomial interpolation method

circuit name	circuit size (ν)	time slots (N)	polyn. degree (m)	sample time (sec)	DFT time (sec)	symbolic (100 pts) (sec)	SCNAP4 (100 pts) (sec)	peak error (dB)
spft	112	24	20	241.14	16.42	23.61	89.06	1e-8
lp10	60	16	15	24.26	4.46	7.25	17.99	1e-9
nos11	48	4	15	1.17	0.33	0.59	3.47	1e-5
bp8	43	2	11	0.20	0.06	0.15	1.70	1e-9
bp6	36	2	10	0.15	0.04	0.06	1.18	1e-6
nos5	24	4	8	0.27	0.12	0.31	0.88	1e-11
bp2	15	2	5	0.05	0.03	0.05	0.23	1e-10
tf	8	2	4	0.03	0.01	0.06	0.09	1e-14
int	5	2	2	<0.01	0.01	0.04	0.07	1e-14

Note: Above data obtained on SUN-Sparc ELC station

From the table, several deductions can be made:

(1) Potentially all N^2 (N is the number of time slots) transfer functions need to be interpolated, though some are zero in the majority of circuits and interpretable code makes most matrix arithmetic common.

(2) The approach effectively needs N solutions of equation (4.7) at each sample frequency, whereas a numerical simulator such as SCNAP4[33-34] needs one solution for each frequency point required.

(3) The DFT (FFT) process takes much less time than the computation of sampled frequency responses.

(4) Once all transfer functions have been interpolated, repetitive expression evaluation is more efficient than direct matrix solution, though for multirate

circuits the efficiency gains decrease because of the increased number of transfer functions.

(5) The polynomial interpolation methods can preserve accuracy even in the analysis of quite large circuits.

(b) Noise calculation

By utilising transpose techniques, all transfer functions from each noise source to the output can be obtained by solving an adjoint system. Referring to section 3.3.3, the problem can be formulated as

$$\psi_i = \tilde{\mathbf{d}}^t \mathbf{M}^{-1} \mathbf{W}_i \tag{4.16}$$

Introduce $\mathbf{X}^a$ as the vector of transfer functions of an adjoint system, and subject to the following equation

$$\mathbf{M}^t \mathbf{X}^a = -\tilde{\mathbf{d}} \tag{4.17}$$

The transfer function of each noise source is then calculated by,

$$\psi_i = -(\mathbf{X}^a)^t \mathbf{W}_i \tag{4.18}$$

The total noise power spectral density at the output is calculated by superposition

$$S_T(\omega) = \sqrt{\sum_{n=0}^{\beta} \sum_{i=0}^{\mu} \psi_i^* S_i(\omega - n\omega_s) \psi_i} \tag{4.19}$$

where $S_i(\Omega)$ is the *i*th input noise source power spectral density and β is the number of bands to be considered. μ denotes the number of noise sources. * represents the complex conjugate.

For example, the adjoint system of a simple 2-phase switched linear network is

$$\begin{bmatrix} \mathbf{I} & -\mathbf{P}_2^t \\ -\mathbf{P}_1^t & z\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{X}_1^a \\ \mathbf{X}_2^a \end{bmatrix} = -\begin{bmatrix} \mathbf{d}\vartheta_1 \\ \mathbf{d}\vartheta_2 \end{bmatrix} \tag{4.20}$$

where ϑ_k are sin(x)/x factors.

After two adjoint system solutions at one sampling frequency, transfer functions of the adjoint system can be obtained from:

$$\begin{pmatrix} \mathbf{X}_1^a \\ \mathbf{X}_2^a \end{pmatrix} = \begin{pmatrix} \mathbf{X}_{11}^a & \mathbf{X}_{12}^a \\ \mathbf{X}_{21}^a & \mathbf{X}_{22}^a \end{pmatrix} \begin{pmatrix} \vartheta_1 \\ \vartheta_2 \end{pmatrix} \tag{4.21}$$

and finally, transfer function of each noise source is generated by referring to (4.18)

Fig. (4.5) shows the baseband noise response of the treble tone control filter. Symbolic analysis spends 0.14 (sec) CPU time for 100 frequency points whiles the numerical simulator SCNAP4 only takes 0.09 (sec) CPU time for the same number of points. It shows that for switched linear networks, no advantages can be obtained by using a symbolic method for noise evaluation. This is mainly due to the fact that all the noise source transfer functions have to be evaluated to obtain the noise response and no further speed-up techniques can be adopted during this procedure. In the numerical approach, due to the employment of adjoint techniques, only a minor increase computation time is required for processing the noise sources. Further comparison by examining the relative theoretical computational costs will be given next.

4.3.4 COMPUTATION COSTS COMPARISON

There are two basic operations which are extensively used in numerical matrix solution and symbolic expression evaluation. Their definitions are presented as follows:

A flop (floating point operation) is defined to be the time required for a particular computer system to execute the C code

$$A = A + C \times D$$

which involves one floating point multiplication, one floating point addition and possibly a few storage references. It is very typical in matrix solution.

Similarly, a trop (trigonometric function operation) is the time required for execute another C code

$$A = A + B \times cos(C \times D)$$

which implies one trigonometric function operation, two floating point multiplications, one floating point addition and a few storage references. This is widely used when z remains as the only symbol in a symbolic expression.

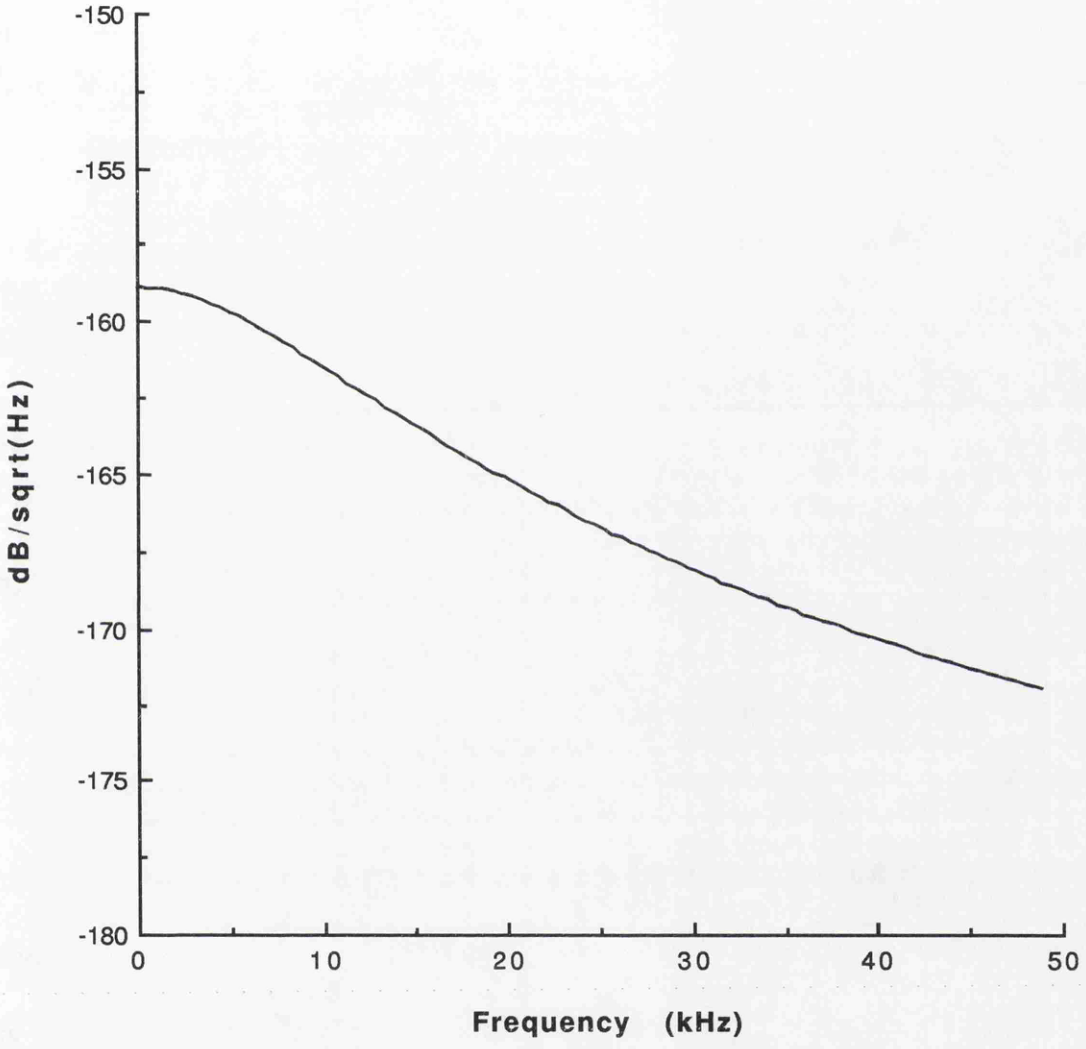


Fig. (4.5) Noise response of a treble tone control filter

(a) Costs of numerical solution

By referring to equation (4.8), the last time-slot matrix equation is

$$(zI - E)V_N(z) = \sum_{k=1}^{N} g_k^N W_k(z) \tag{4.22}$$

which can be very efficiently solved using a Hessenburg approach. The solutions for $V_k(z)$ for k = 1, 2, ..., N-1 are then obtained by block back substitution,

$$V_1(z) = P_1 V_N(z) + W_1(z)$$
$$V_k(z) = P_k V_{k-1}(z) + W_k(z) \qquad k = 2, \cdots, N-1$$

Therefore, solving equation (4.22) takes about $5\nu^2$ flops, where ν is the matrix dimension. The solutions of all other slots require another $2(N-1)\nu^2$ flops and make a total $(2N+3)\nu^2$ flops per frequency point for numerical matrix solution. The theoretical costs are compared to actual run times obtained for a number of typical applications and illustrated in Fig. (4.6).

(b) Costs of expression evaluation

In general, a network transfer function can be expressed as

$$H(z) = \frac{\sum_{k=0}^{m} a_k z^k}{\sum_{k=0}^{m} b_k z^k} \tag{4.23}$$

where a_k and b_k are real coefficients.

By using

$$z^k = e^{j\omega kT} = \cos(\omega kT) + j\sin(\omega kT) \tag{4.24}$$

It is evident that at least 4m trops are required for each transfer function evaluation per frequency point. If the network has N time slots, then the total computational costs is about $4mN^2$ trops. The mN^2 dependence of the expression evaluation is shown in Fig. (4.7).

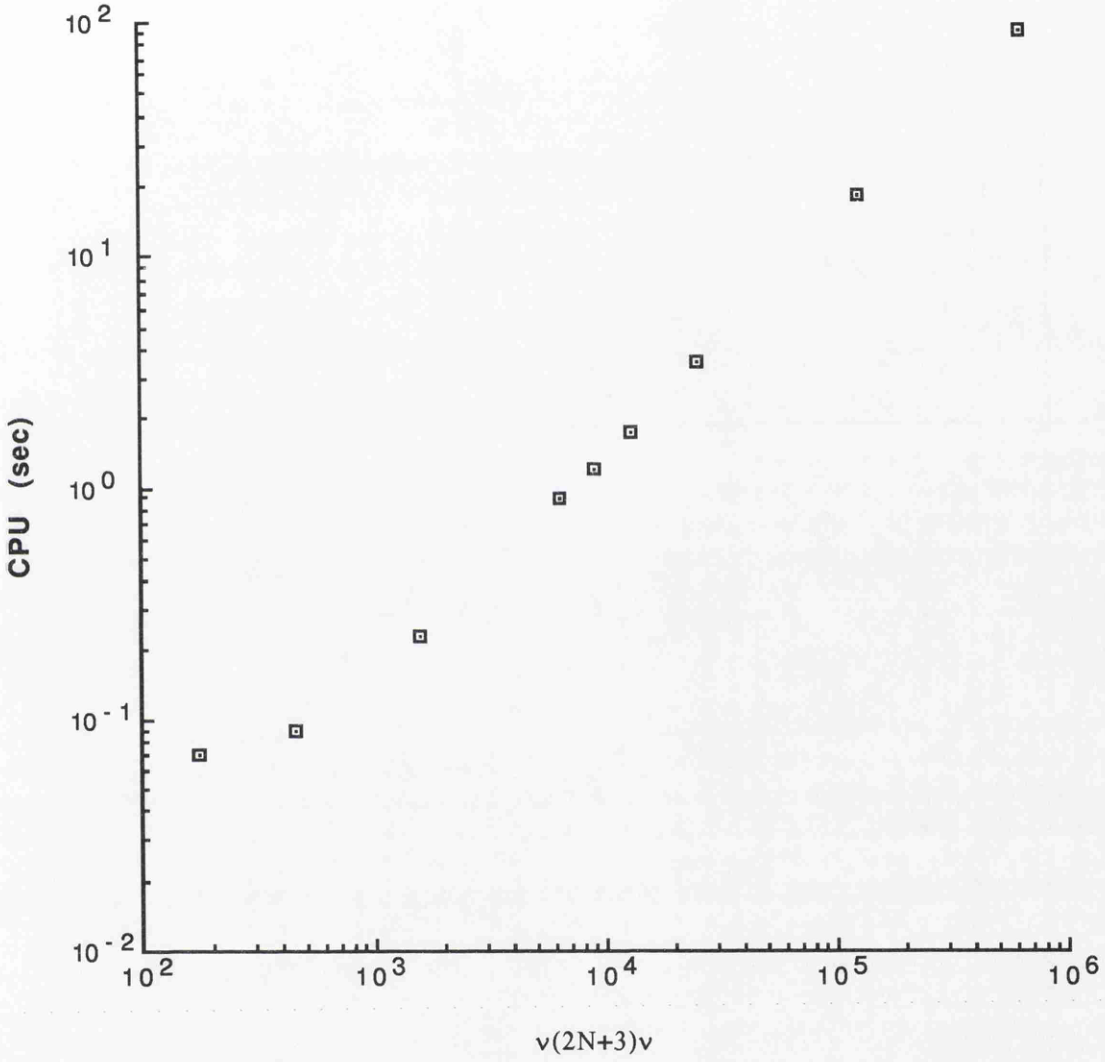


Fig. (4.6) Theoretical cost dependence of numerical matrix solution

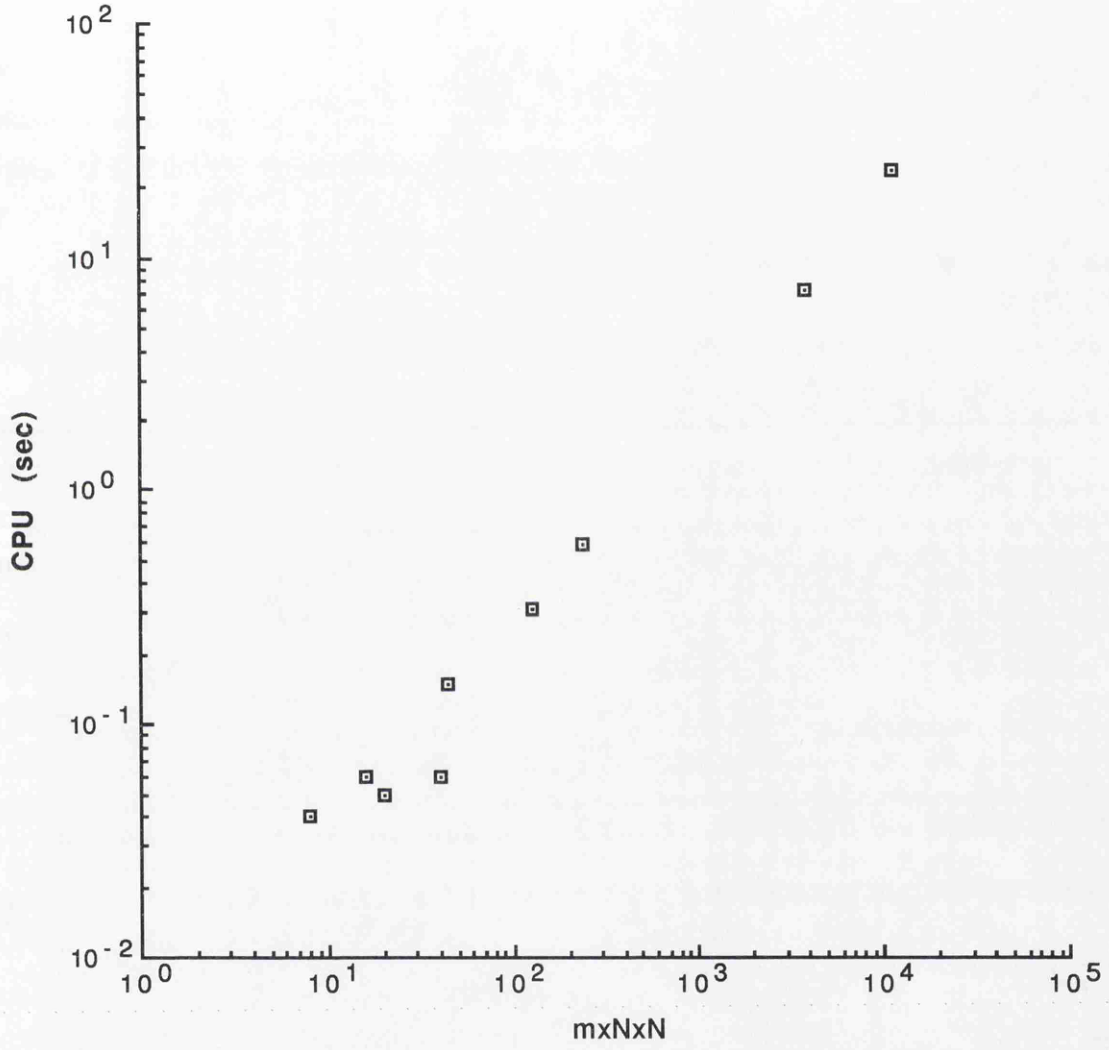


Fig. (4.7) Theoretical cost dependence of expression evaluation

(c) Comparison and conclusions

Define an efficiency parameter η as

$$\eta = \frac{(2N+3)\nu^2}{4\gamma mN^2} \tag{4.25}$$

where γ is a computer system related constant which reflects the relationship between flop and trop in terms of computer execution time. For example, on a SUN-Sparc ELC station, one trop takes about nine times longer than one flop, therefore, $\gamma = 9$.

η can be used to estimate the efficiency of the symbolic expression evaluation against numerical matrix solution.

In the computation of the transfer functions for a number of examples listed in Table (4.2), it can be found, that although η varies for different networks, $\eta > 1$ is always true which means that symbolic expression evaluation is efficient than numerical matrix solution. According to formula (4.25), several deductions can be made:

(1) For 2-phase circuits maximum efficiency of analysis is obtained by using symbolic expression evaluation. As the N increases (multi-rate circuits), the efficiency of the symbolic method will decrease, though the total run-time is dominated by an increase in the interpolation time which is a fixed overhead and the method may still be very attractive when many frequency points are required.

(2) For noise analysis by expression evaluation, a large number of noise sources can be involved and transfer functions for each noise source are required. If there are μ noise sources in a circuit, then the total computation costs will increase up to $4\mu\gamma mN^2$. The example in section 4.3.2 has already shown that the application of symbolic techniques to noise analysis is not very successful.

4.4 SUMMARY

A symbolic analysis method utilising polynomial interpolation suitable for large nonideal switched linear networks has been presented. Other approaches are also examined for the suitability and their inadequacies for the purpose are described. Possible techniques for improving accuracy of the interpolation method are discussed.

A number of application examples show that symbolic expression evaluation is more efficient than numerical matrix solution for computing the network response. Even for large network, accuracy has been satisfactorily retained. However, the application to the noise analysis is not very successful. Explanations are given through the comparison of the theoretical computation costs between the symbolic and numerical methods.

REFERENCES

[1] G. Gielen, H. Walscharts, W. Sansen, "ISAAC: a symbolic simulator for analog integrated circuits," IEEE Journal of Solid-State Circuits, vol. SC-24, no. 6, Dec. 1989, pp. 1587-1597

[2] S. Sedra, M. Degrauwe, W. Fichtner, "A symbolic analysis tool for analog circuit design automation," Proc. IEEE ICCAD, Santa Clara, 1988, pp. 488-491

[3] A. Libertore, S. Manetti, "SAPEC - a personal computer program for the symbolic analysis of electronic circuits," Proc. IEEE ISCAS, Helsinki, May 1988, pp.897-900

[4] A. Konczykowska, M. Bon, "Automated design software for switched-capacitor IC's with symbolic simulator SCYMBAL," Proc. DAC, Anaheim, 1988, pp.362-368

[5] S. -D. Shieu, S. -P. Chan, "Topological formulation of symbolic network functions and sensitivity analysis of active networks," IEEE Trans. Circuits and Syst., vol. CAS-21, no.1, Jan.1974, pp. 39-45,

[6] G. E. Alderson, P. M. Lin, "Computer generation of symbolic network functions - new theory and implementation," IEEE Trans. Circuit Theory, vol. CT-20, no.1, Jan. 1973, pp. 48-56

[7] P. Sannuti, N. N. Puri, "Symbolic network analysis - an algebraic formulation," IEEE Trans. Circuits and Syst., vol. CAS-27, no. 8, Aug. 1980, pp. 679-687

[8] J. K. Fidler, J. I. Sewell, "Symbolic analysis for computer-aided circuits design - the interpolative approach," IEEE Trans. Circuit Theory, vol. CT-20, no. 6, Nov. 1973, pp. 728-741

[9] K. Singhal, J. Vlach, "Symbolic analysis of analog and digital circuit," IEEE Trans. Circuits and Syst., vol. CAS-24, no. 11, Nov. 1977, pp. 598-609

[10] M. M. Hassoun, P. M. Lin, "A new network approach to symbolic simulation of large-scale networks," Proc. IEEE ISCAS, Portland, May 1989, pp. 806-809

[11] M. M. Hassoun, " Hierarchical symbolic analysis of large-scale systems using a Mason's signal flow graph model," Proc. IEEE ISCAS, Singapore, Jun.1991, pp. 802-805

[12] T. Matsumoto, N. Nakayama, K. Tsuji, "Symbolic analysis for large networks and systems including both numerical and symbolic parameters by using hybrid decomposition method," Proc. IEEE ISCAS, Singapore, Jun. 1991, pp. 794-797

[13] G. Gielen, P. Wambacq, W. Sansen, " Symbolic approximation strategies and the symbolic analysis of large and nonlinear circuits," Proc. IEEE ISCAS, Singapore, Jun. 1991, pp. 806-809

[14] S. -M. Chang, J. F. Mackay and G. M. Wierzbr, "Matrix reduction and numerical approximation during computation techniques for symbolic analysis," Proc. IEEE ISCAS, San Diego, May 1992, pp. 1153-1156

[15] G. Gielen, P. Wambacq, W. M. Sansen," Symbolic analysis Methods and applications for analog circuits: A tutorial overview," Proc. IEEE, vol. 82, no. 2, Feb. 1994, pp. 287-304

[16] F. V. Femandez, P. Wambacq, G. Gielen, A. R.- Vasquez, W. M. Sansen, "Symbolic analysis of large analog integrated circuits by approximation during expression generation," Proc. IEEE ISCAS, London, Jun. 1994, pp. 1.25-1.28

[17] S. Jou, M. Perng, C. Su and C. K. Wang, "Hierarchical techniques for symbolic analysis of large electronic circuits," Proc. IEEE ISCAS, London, Jun. 1994, pp. 1.21- 1.24

[18] D. G. Johnson and J. I. Sewell, "Improved z plane polynomial interpolative analysis of switched capacitor networks," IEEE Trans. CAS. vol. CAS-31, no.7, 1984, pp.666-668

[19] C. K. Pun and J. I. Sewell, "Symbolic analysis of ideal and non-ideal switched capacitor networks," Proc. IEEE ISCAS, Kyoto, June 1985, pp. 1165-1168

[20] C. K. Pun, A. G. Hall and J. I. Sewell, "Noise analysis of switched capacitor networks in symbolic form," Proc. 29th Midwest Symp., Lincoln, Na., August 1986, pp. 807-810

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

[22] M. L. Griss, "An efficient sparse minor expansion algorithm," Proc. ACM 76, 1976, pp. 429-434

[23] P. M. Lin, "Computer generation of symbolic network functions - an overview," Proc. of Working Conference on Principles of CAD, North-Holland, 1973, pp.261-282

[24] S. J. Mason, "Feedback theory - some properties of signal flow graphs," Proc. of the IRE, Sept. 1953, pp.1144-1156

[25] L. O. Chua, P. M. Lin, *Computer-aided analysis of electronic circuits: algorithms and computational techniques*, Prentice-Hall, 1975

[26] J. F. Traub, "Associated polynomials and uniform methods for solution of linear problems," SIAM Rev., vol.9, no.3, 1966, pp.277-301

[27] C. Fridas, J. I. Sewell, "Symbolic analysis of networks by partitioned polynomial interpolation," IEEE Trans. on Circuits and Syst., vol. CAS-21, pp. 345-347, 1974

[28] V. N. Faddeeva, *Computational Methods of Linear Algebra*, New York: Dover, 1959

[29] K. Singhal, J. Vlach, "Interpolation using the Fast Fourier Transform," Proc. of IEEE, Dec. 1972, pp. 1158

[30] K. Singhal, J. Vlach, "Generation of immittance functions in symbolic form for lumped distributed active networks," IEEE Trans. on Circuits and Syst., vol. CAS-21, no.1, Jan. 1974, pp. 57-66

[31] J. A. Nossek, G. C. Temes, "Switched-capacitor filter design using bilinear element modeling," IEEE Trans. on Circuits and Syst., vol. CAS-29, Apr. 1982, pp. 215-220

[32] Y. Lu, R. K. Henderson, J. I. Sewell, "XFILT: an X-window based modern filter and equaliser design system," Proc. of EECTD, Davos, 1993, pp.305-310

[33] Z. Q. Shang, J. I. Sewell, "Efficient sensitivity analysis for large non-ideal switched capacitor networks", Proc. IEEE ISCAS, Chicago, May, 1993, pp. 1405-1407

[34] Z. Q. Shang, J. I. Sewell, "Efficient noise analysis methods for large non-ideal SC and SI circuits" Proc. IEEE ISCAS, London, Jun., 1994, pp. 5.565-5.568

CHAPTER 5

SWITCHED NONLINEAR NETWORK ANALYSIS

5.1 INTRODUCTION

The previous chapters were essentially concerned with switched linear networks. In the present Chapter, efficient time domain analysis methods for nonlinear switched networks will be studied.

For high performance filter design, many second order effects are of great interest to designers. In the design software for switched capacitor (SC) networks, linear nonideal imperfections such as switch resistance (R_{on} and R_{off}), and the finite gain and bandwidth of the opamp, have already been addressed. However the simulation of nonlinearities in SC networks is less developed. In practice the nonlinear behaviour of SC and SI (switched current) networks is of concern and it is also very important to be able to estimate the nonlinear effects efficiently during the design phase. Often, the only feasible CAD tool is to employ a general purpose circuit simulator in which sophisticated transistor models are incorporated. Consequently a very time consuming analysis results because the inherent sampling process of switched networks necessitates the computation of a large number of sharp waveform transitions. Since many existing switched network simulators are often extremely efficient in the linear domain, it is attractive to exploit such properties to compute some of the dominant nonlinear effects. It is of interest to examine which sources of distortion can be accurately and efficiently modelled without recourse to full device models and time consuming simulation.

In this Chapter, the dominant nonlinear characteristics of the elements in switched networks are described. By modelling nonlinearities with piecewise linear equations, the original nonlinear network can be replaced by a piecewise linear network. A modified Katzenelson algorithm is then proposed which is fully compatible with the existing linear techniques used in the simulator SCNAP4[1-2]. Hence many effective approaches can be directly applied and the efficiency of the proposed method is then guaranteed.

5.2 NONLINEARITIES IN SC AND SI CIRCUITS

Although both switched capacitor and switched current circuits are designed to be used in linear application, some nonlinear behaviour is also inherent. The following sections will outline the nonlinear characteristics of the basic elements of SC and SI circuits.

5.2.1 Switches and capacitors

The nonlinear behaviour of a MOS transistor switch can be studied by the model[3] depicted in Fig. (5.1).

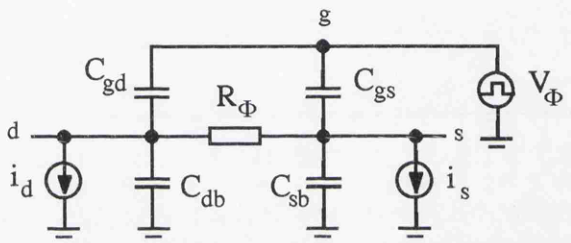


Fig. (5.1) MOS transistor switch model

where R_Φ denotes the switch resistance and could be either R_{on} ($\Phi = 1$) or R_{off} ($\Phi = 0$).

The clock feedthrough due to nonlinear coupling of the clock signals into the main signal path is modelled by a clock generator V_Φ coupled through the overlap capacitors C_{gd} and C_{gs} and onto the signal path.

The charge-dumping effect is caused by the charge in the channel dumping through the signal path during switch turn off. It can be represented by two equivalent current sources, i_d and i_s.

Though switch resistance also exhibits nonlinear behaviour, in most cases, it is judged to be of little consequence compared to other source of nonlinearities. The same conclusion can be applied to either circuit or parasitic capacitors. In fact, the accuracy of the capacitor ratio is of more concern in switched capacitor circuits since the major signal processing operation relies on the charge movement from one capacitor to another.

In switched current circuits, capacitors are usually used to store the transistor state and are not actively involved in charge transfer, hence their degree of linearity or

otherwise is of little significance. The ratio of transistor areas in a current mirror determines the accuracy of the signal processing. However, transistors within the memory cells exhibit signal dependent transconductance characteristics, and in some of the more advanced cells, slew-rate dependence is apparent [4].

5.2.2 Operational amplifier

It has been found that in SC networks, the major contributions to distortion come from operational amplifiers by means of the slew-rate limitation and the output voltage saturation mechanism. A typical macromodel for an opamp proposed in [5-6] is used to demonstrate the nonlinear behaviour of the opamp.

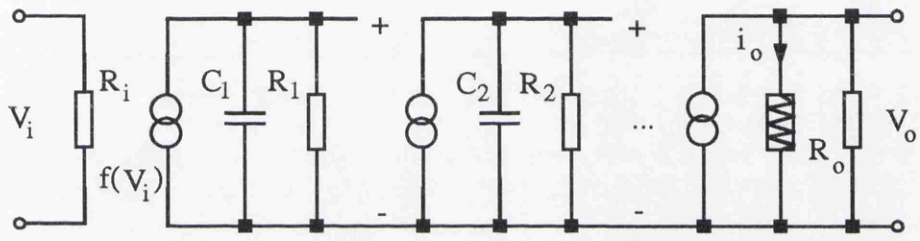


Fig. (5.2a) opamp macromodel with slew-rate, output limiting

The slew-rate limiting (and the nonlinear transconductance) can be modelled by a nonlinear voltage controlled current source as shown in Fig. (5.2b)

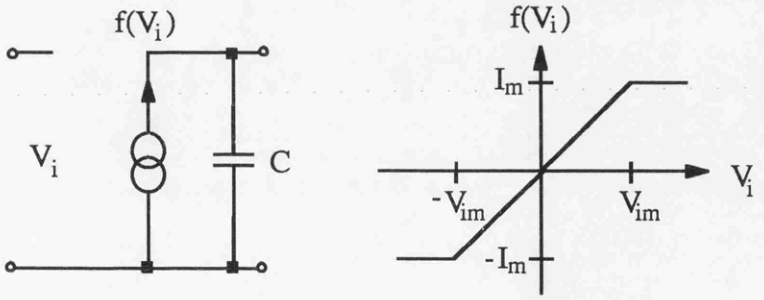


Fig. (5.2b) Model for slew-rate limiting

The maximum current I_m provided by nonlinear voltage controlled current source (VCCS) limits the voltage change across the capacitor. The slew-rate S_r can be selected as

$$\left|\frac{dv_c}{dt}\right| = \frac{|i_c|}{C} \leq \frac{I_m}{C} = S_r$$

The output voltage swing limitation is due to the power supply and can be simply modelled by a nonlinear resistor which has the I-V characteristic as shown in Fig. (5.2c)

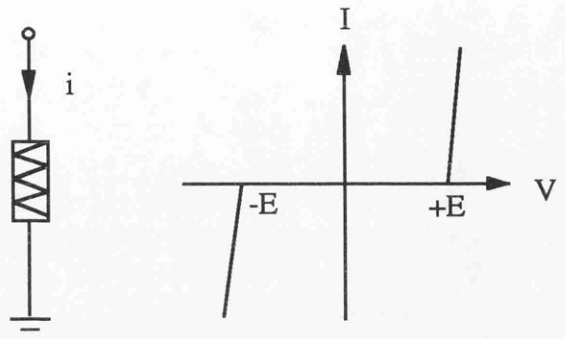


Fig. (5.2c) Model for output voltage limiting

If the slew-rate limiting effect can be ignored (in practice, high slew-rate is not difficult to achieve) and only output voltage saturation is considered, the opamp is then simplified to be a nonlinear voltage controlled voltage source as illustrated in Fig. (5.2d)

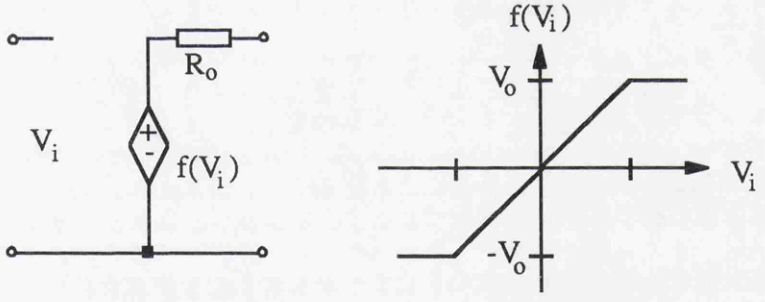


Fig. (5.2d) Model for output voltage saturation

One observation can be made is that only a few piecewise linear segments are required to describe the most important nonlinearities in opamp. However such simple nonlinear characteristics are not available in conventional circuit simulators. Instead, more sophisticated device models have to be involved to simulate these nonlinearities.

5.2.3 Overview

Generally, nonlinearities in electronic circuits can be analysed by two types of techniques, conventional and piecewise linear. Most of the circuit simulators are based on first type which represent the nonlinear characteristics by analytical

functions and establish a set of coupled, nonlinear, algebraic differential equations for the circuit. A number of well developed effective numerical approaches [7] can be used to solve the equations and very accurate results can be achieved. However, this is not adequate for time-variant networks due to the requirement for computation of a large number of sharp waveform transitions which are common in SC and SI networks. A piecewise linear technique [8], on the other hand, expresses the nonlinearities by their piecewise linear approximation. The accuracy and efficiency of the analysis can be controlled by the number of piecewise linear segments involved in the modelling. In the situation where nonlinearities can only be described by tables of measured values or analytical functions are too complicated, the piecewise linear segment description is a very good alternative. More importantly, since the original nonlinear network is replaced by piecewise linear network, it is possible to combine some efficient techniques used in the analysis of switched linear networks with the piecewise linear technique to deal with some dominant nonlinearities present in switched networks.

In general, for SC and SI networks, clock-feedthrough and charge dumping effects require full device models and imply full SPICE type simulation, though accurate computation of charge dumping effect by macro-modelling techniques in the linear simulator SCNAP4 have been reported [9]. Other nonlinearities can be represented by their piecewise linear approximation. In most cases, only monotonic nonlinear characteristics with a few piecewise linear segments are required when significant advantage can be gained by adopting piecewise linear techniques.

5.3 PIECEWISE LINEAR METHOD

5.3.1 Piecewise linearization

Consider a periodically switched nonlinear network, all nonlinear elements are represented by their piecewise linear models. In the *k*th time slot of the *n*th switching period (nT, (n+1)T], it can be represented by the system differential equation [10]

$$C_k \dot{v}_{n,k}(t) + G_k v_{n,k}(t) = w_{n,k}(t) \qquad k=1, ..., N \qquad (5.1)$$

where $w_{n,k}(t)$ is the vector of excitations, C_k and G_k are capacitance and conductance matrices where all nonlinear elements are represented by their piecewise linear approximation; $v_{n,k}(t)$ is the vector of unknown system variables. For the simplicity of notation, only one independent source is considered (for the general situation refer to section 3.2.4). Approximate the excitation by polynomials of order *m*

$$w_{n,k}(t) \approx \sum_{i=0}^{m} \alpha_{i,k}^{n} t^{i} \tag{5.2}$$

the new system differential equation is then

$$C_k \dot{v}_{n,k}(t) + G_k v_{n,k}(t) = \sum_{i=0}^{m} \alpha_{i,k}^{n} t^{i} \tag{5.3}$$

Since the network is linear in each piecewise region, take the Laplace transform of (5.3) to obtain

$$V_{n,k}(s) = \left[sC_k + G_k\right]^{-1} \left\{ C_k v_{k-1} + \sum_{i=0}^{m} \frac{\alpha_{i,k}^{n} i!}{s^{i+1}} \right\} \tag{5.4}$$

The inverse Laplace transform of (5.4) is then

$$v_{n,k}(t) = P_k v_{k-1}(nT + \sigma_{k-1}) + \sum_{i=0}^{m} \alpha_{i,k}^{n} B_{i,k}(t) \tag{5.5}$$

Definitions of P_k and $B_{i,k}$ can be found in Chapters 2, here $v_{k-1}(nT + \sigma_{k-1})$ are the initial conditions of the system for the *k*th slot.

Define an error vector f as

$$f = v_{n,k} - P_k v_{k-1} - \sum_{i=0}^{m} \alpha_{i,k}^{n} B_{i,k} \tag{5.6}$$

Apply the Newton-Raphson method to (5.6) and get

$$(v_{n,k}^{j+1} - v_{n,k}^{j}) = -f^{j} \tag{5.7}$$

where the superscript j denotes the iteration number. It can be notice that the Jacobian of (5.6) is an identity matrix since the network is linear in each region.

Substituting (5.6) into (5.7) to obtain

$$v_{n,k}^{j+1} = P_k^{j} v_{k-1} - \sum_{i=0}^{m} \alpha_{i,k}^{n} B_{i,k}^{j} \tag{5.8}$$

If there is no element crosses into a new region, v^{j+1} is the final solution, otherwise further iterations are required. The step size is reduced in such a way that only one element goes to the boundary of its present region. An effective approach was proposed in [11], the step size reduction coefficient μ can be determined from

$$\mu_i^{j} = \frac{v_b - v^{j}}{\Delta v^{j}} \tag{5.8}$$

where v_b is the boundary point, i denotes the *i*th element under consideration.

After μ^j has been determined, new solution is obtained from (5.8) and the iteration is repeated until the desired solution is reached.

5.3.2 Computer implementation

Before discussion of the computer implementation, it is necessary to give a new definition for network state. Let the *l*th network linear region Ω_l be represented by

$$\Omega_l = (\Gamma_1, \Gamma_2, \cdots, \Gamma_n)$$

where $\Gamma_i \in I$ (I represents the integer collective) denote the Γ_ith segment of the *i*th piecewise linear element, n is the number of nonlinear elements in the network.

The network state can then be defined as

$$\chi_j = (\Omega_l, \tau_k)$$

where τ_k is the time interval in the *k*th time slot.

According to the definition, it is evident that for linear switched networks, Ω_l is empty and χ_j only depends on τ_k. Hence the total number of network states is identical to the number of time slots in one period. Therefore the matrices related to the network states can be pre-calculated. It is worthwhile to recall the time domain analysis algorithm for switched linear networks so that further comparison can be made.

```
/* algorithm for switched linear networks */
for (k = 1; k <= NSLOT; k++) {
    t += Tstep;
    if (t > Tstop) stop;
    Processing excitations;
    v_{n,k}(t) = P_k v_{k-1}(nT + σ_{k-1}) + Σ_{i=0}^{m} α^n_{i,k} B_{i,k}(t);
    v_{k-1}(nT+σ_{k-1}) = v_{n,k}(t);
    output v_{n,k} ;
}
```

From the above algorithm, it is evident that the time domain solution is very efficient since the procedure merely employs some substitutions and P_k, $B_{i,k}$ have already been pre-calculated.

However, in the switched nonlinear situation, in each time slot, there could be many possible linear regions in terms of Ω_l and the subsequent network state is totally unpredictable. It is therefore not feasible to execute pre-processing of the matrices of each network state and this leads to increased complexity in the problem.

An effective solution was proposed in [12]. Instead of calculating P_k and $B_{i,k}$ at a pre-processing stage, they are generated on demand during the time domain analysis. The computer algorithm can be summarised as follows:

```
/* nonlinear iteration procedure */
j = 0;
IT: if (network_state == NEW)
    { calculate P_k^j , B_{i,k}^j and store them in cache; }
  else
    { retrieve P_k^j and B_{i,k}^j from cache; }
  solve equation (7);
  if (convergence != TRUE) {
    for (each element i that changes state)
      { μ_i^j = (v_b − v^j) / Δv^j ; /* v_b is the boundary point*/ }
    μ^j = min(μ_i^j);
    determine network_state;
    j++;
    goto IT;
  } else {
    output v^{j+1};
  }
```

The network state is checked at each iteration and when a new state is found, P_k and $B_{i,k}$ associated with that state are calculated and stored in matrix cache. If an old network state is detected, the appropriate matrices and vectors are retrieved from the matrix cache. Therefore, the new algorithm has the same property that only one calculation for P_k and $B_{i,k}$ per each network state is needed and the overall computation is minimised to retain efficiency. The program SCNAPDIS incorporates these routines along with sparse matrix techniques and interpretable code generation [8] which enhance the efficiency even further.

5.3.3 Computer examples

Several examples are presented to demonstrated the performance of SCNAPDIS.

The first example is a phase-compensated positive-gain amplifier [13]. The circuit diagram is illustrated in Fig. (5.3a). A sinusoidal input of 6 volt and 10kHz frequency is applied. The output voltage saturation level of amplifier is set to 15 volt. Although the circuit has no switches, SCNAPDIS is also able to handle this type of circuits. It requires 0.11 s CPU time for 100 time points. The simulation results are dipicted in Fig. (5.3b). It shows clearly the effects of output voltage saturation limitation.

Next example is a typical 7th-order Chebyshev lowpass SC filter. The circuit diagram and clock waveforms are shown in Fig. (5.4a) and Fig. (5.4b), respectively. Initially all components are linear and a 500Hz sine wave is applied, SCNAPDIS running on a SUN-Sparc ELC workstation required 17.5s to compute the time response at 6144 points, an FFT on 1024 samples produces the spectrum of the output voltage shown in Fig. (5.5a), where only the 500 Hz tone can be seen. The amplifiers were then given an asymmetric saturation characteristic (+7V, -3V) and the input drive increased to 7V amplitude. It now required 49.19 s to compute the time response and the FFT produced the spectrum of Fig. (5.5b), where even and odd harmonics are visible. If the saturation levels are set symmetrically (+7V, -7V), then it can be observed from Fig. (5.5c) that all even harmonics disappear. Fig. (5.5d) is the output spectrum when the amplifiers have an alternative slew-rate nonlinearity of 1.57×10^4 V/sec, odd harmonics are produced.

Fig. (5.6) shows the output spectrum for a 5th-order elliptic lowpass SI ladder based filter network [14] with symmetric slew-rate nonlinearities ($S_r = 3\mu V/\mu sec$), again odd harmonics are apparent.

Application to large switched networks is demonstrated in Fig.(5.7). The frequency response of a single-path frequency-translated filter [15] is shown in Fig. (5.7a), where the bandpass characteristic around 20kHz is of interest, this is a very narrow band filter of 80 Hz bandwidth. A 20kHz tone is applied and the resultant output spectra when linear amplifiers and ones with symmetric saturation characteristics are employed are shown in Fig. (5.7b), distortion components are translated into the appropriate bands. When asymmetric saturation limits are introduced, distortion products are distributed all over the spectrum as shown in Fig. (5.7c).

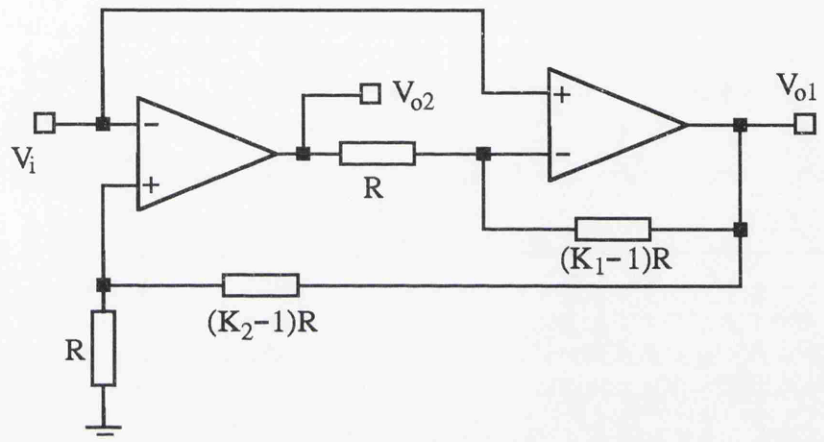


Fig. (5.3a) A phase-compensated positive-gain amplifier

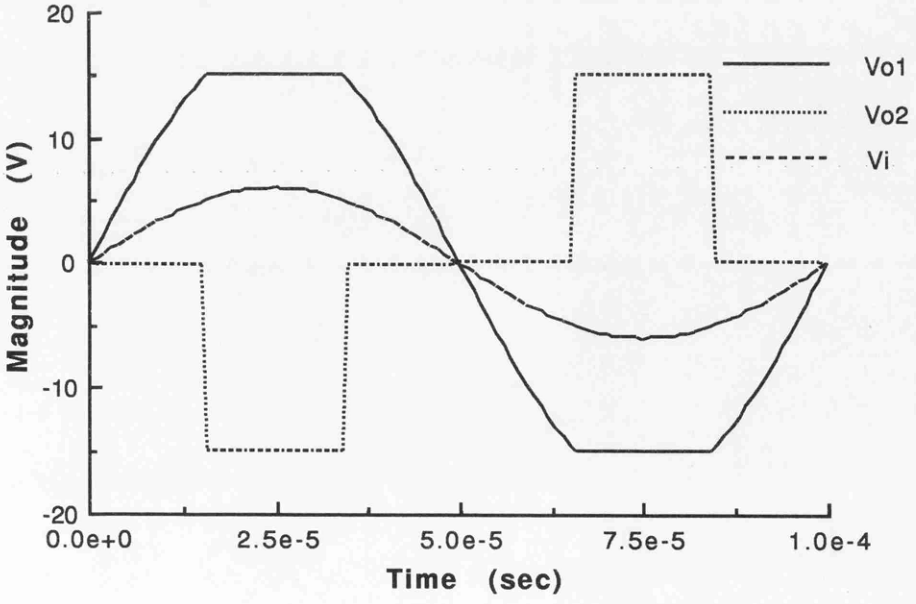


Fig. (5.3b) Simulated waveforms for the circuit in Fig. (5.3a)

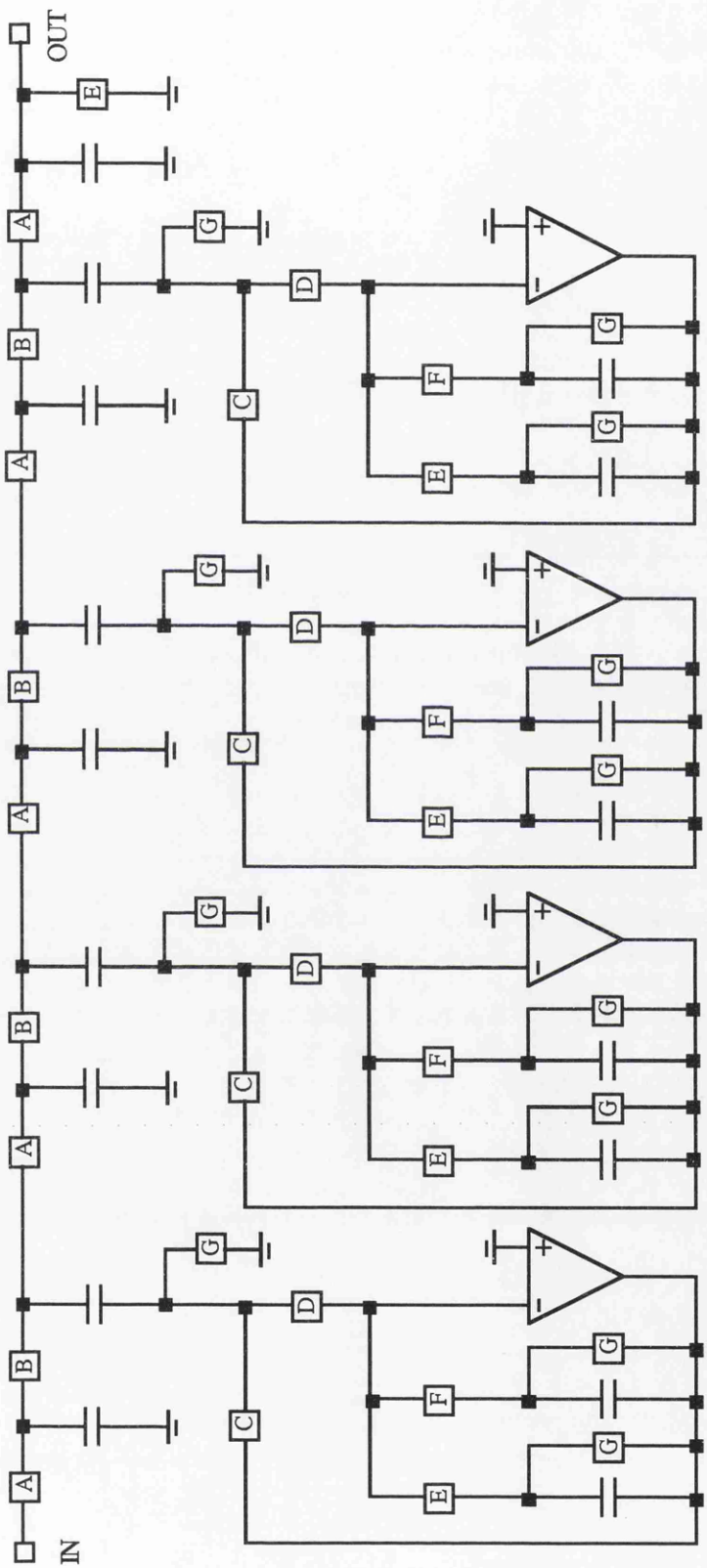


Fig. (5.4a) 7th-order Chebyshev SC filter

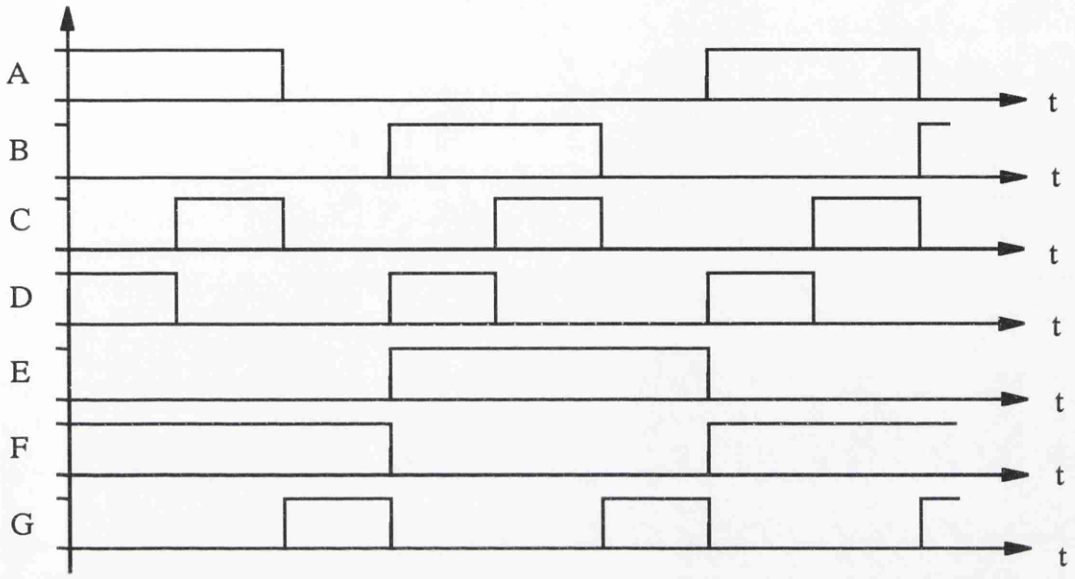


Fig. (5.4b) Clock waveforms for 7th-order Chebyshev SC filter

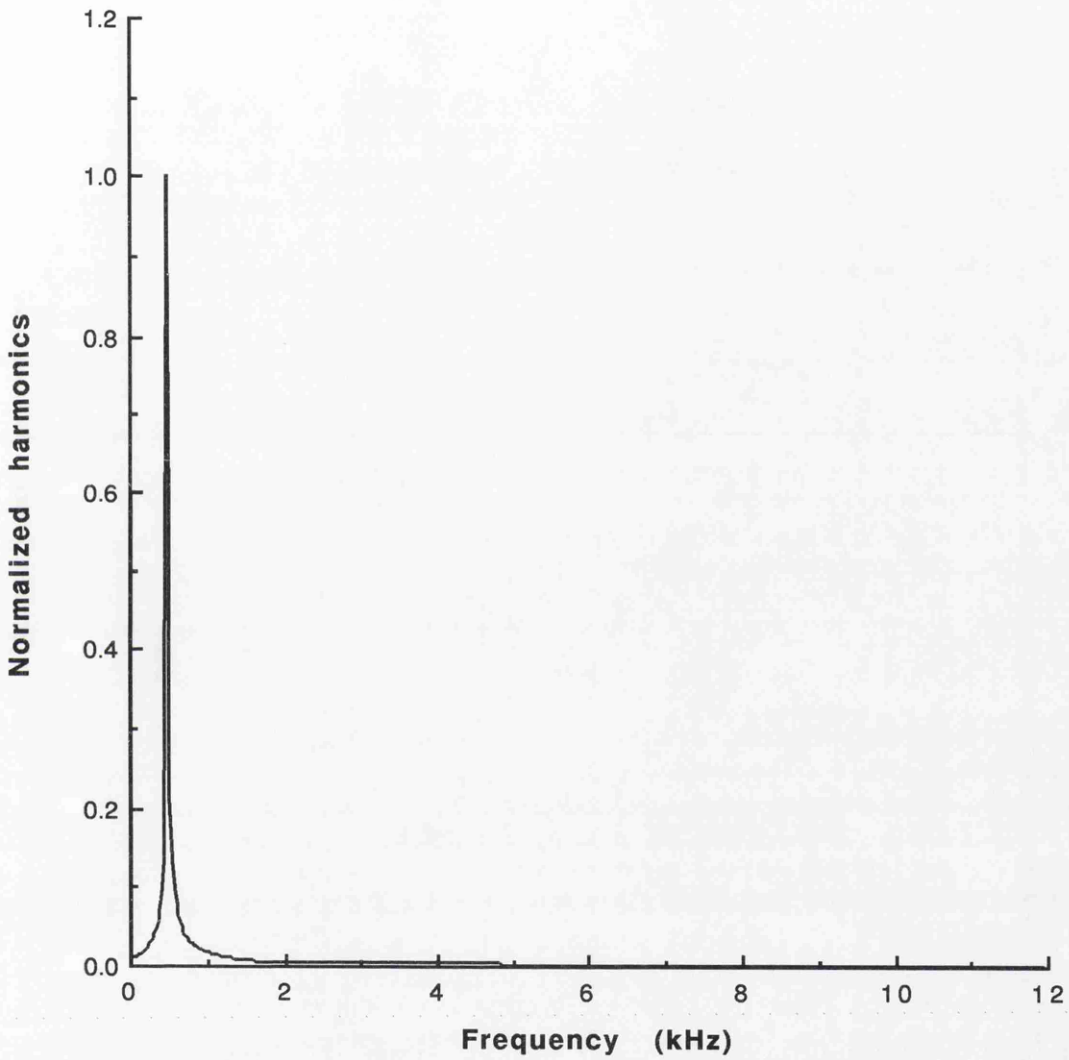


Fig. (5.5a) FFT spectrum for linear 7th-order lowpass SC filter with 500Hz sine wave excitation

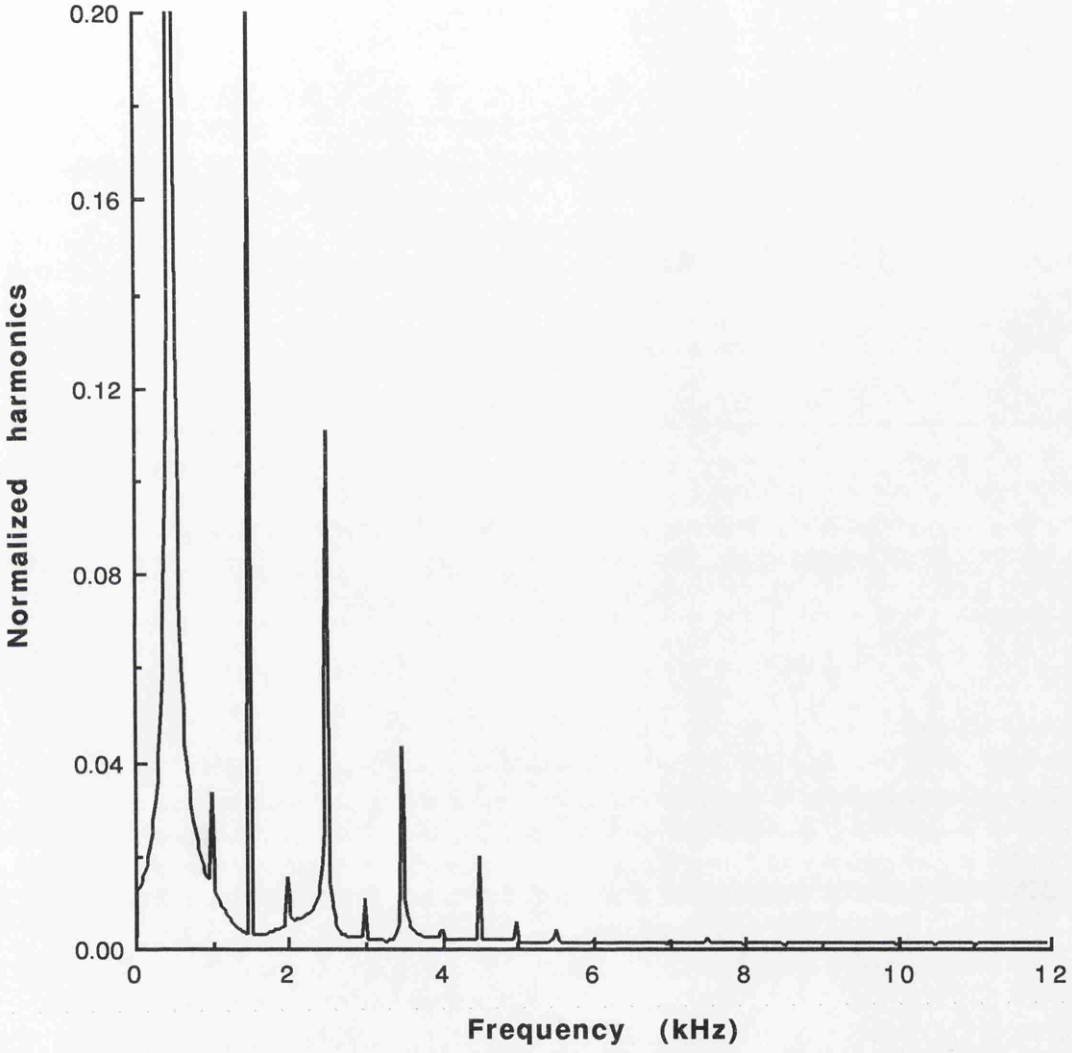


Fig. (5.5b) FFT spectrum for 7th-order lowpass SC filter, with asymmetric amplifier saturation: 500Hz excitation

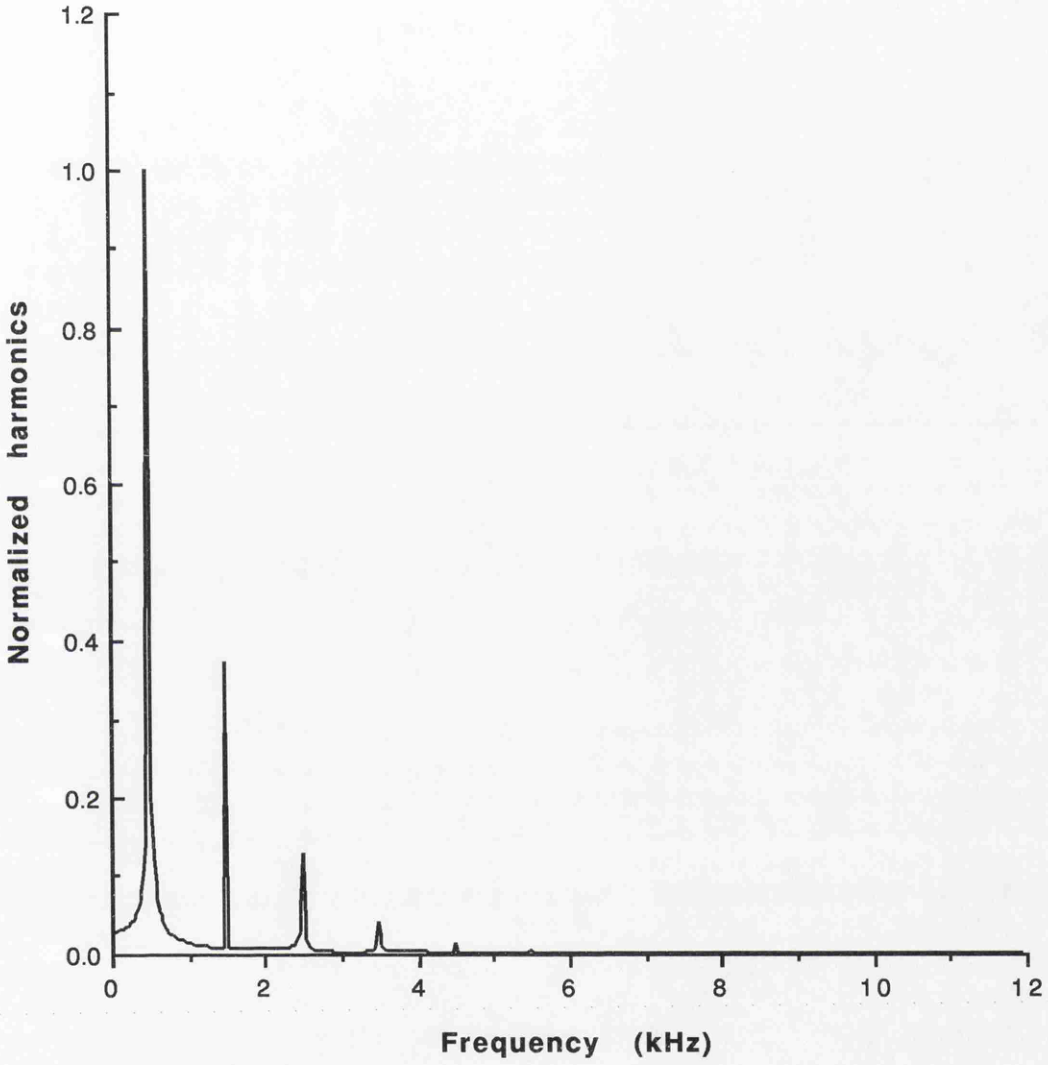


Fig. (5.5c) FFT spectrum for 7th-order lowpass SC filter, with symmetric amplifier saturation: 500Hz excitation

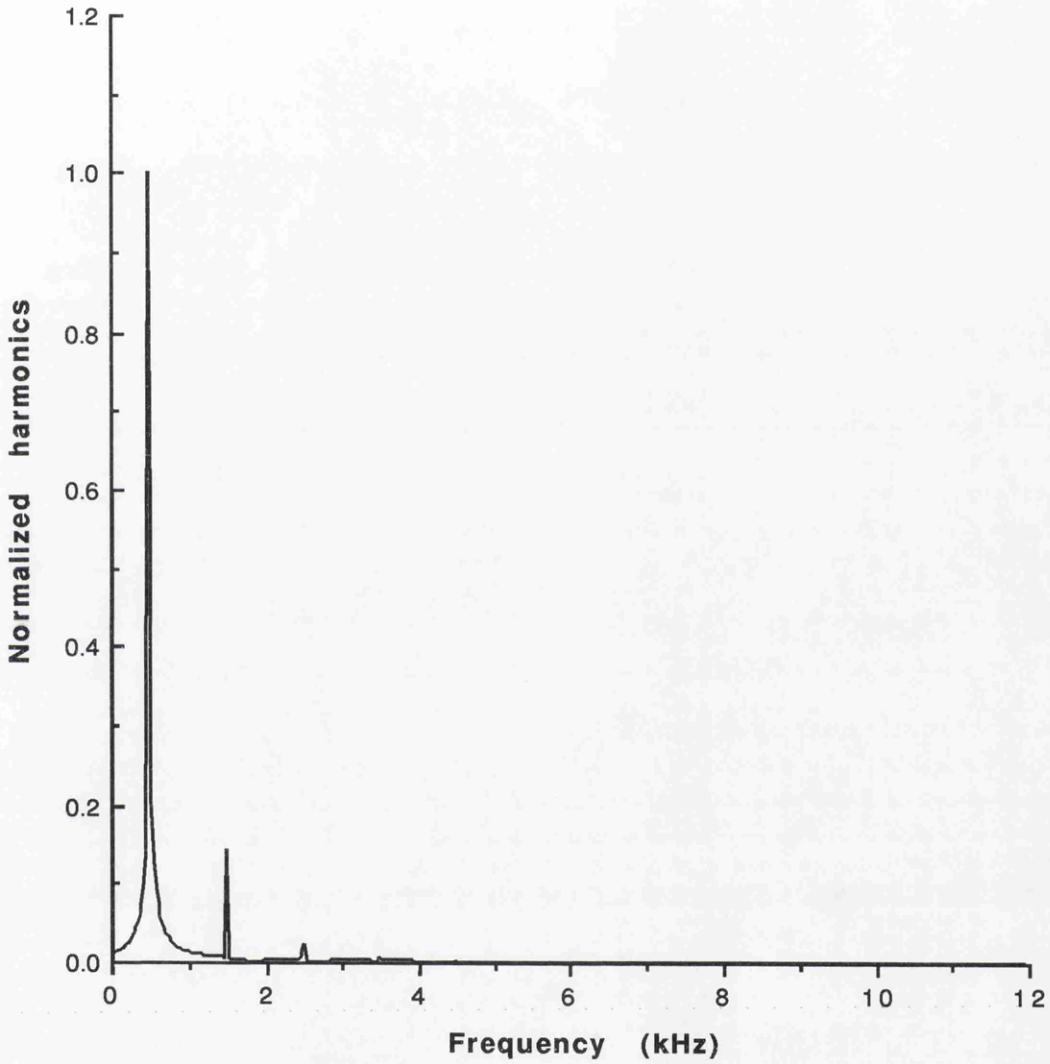


Fig. (5.5d) FFT spectrum for 7th-order lowpass SC filter with finite slew-rate

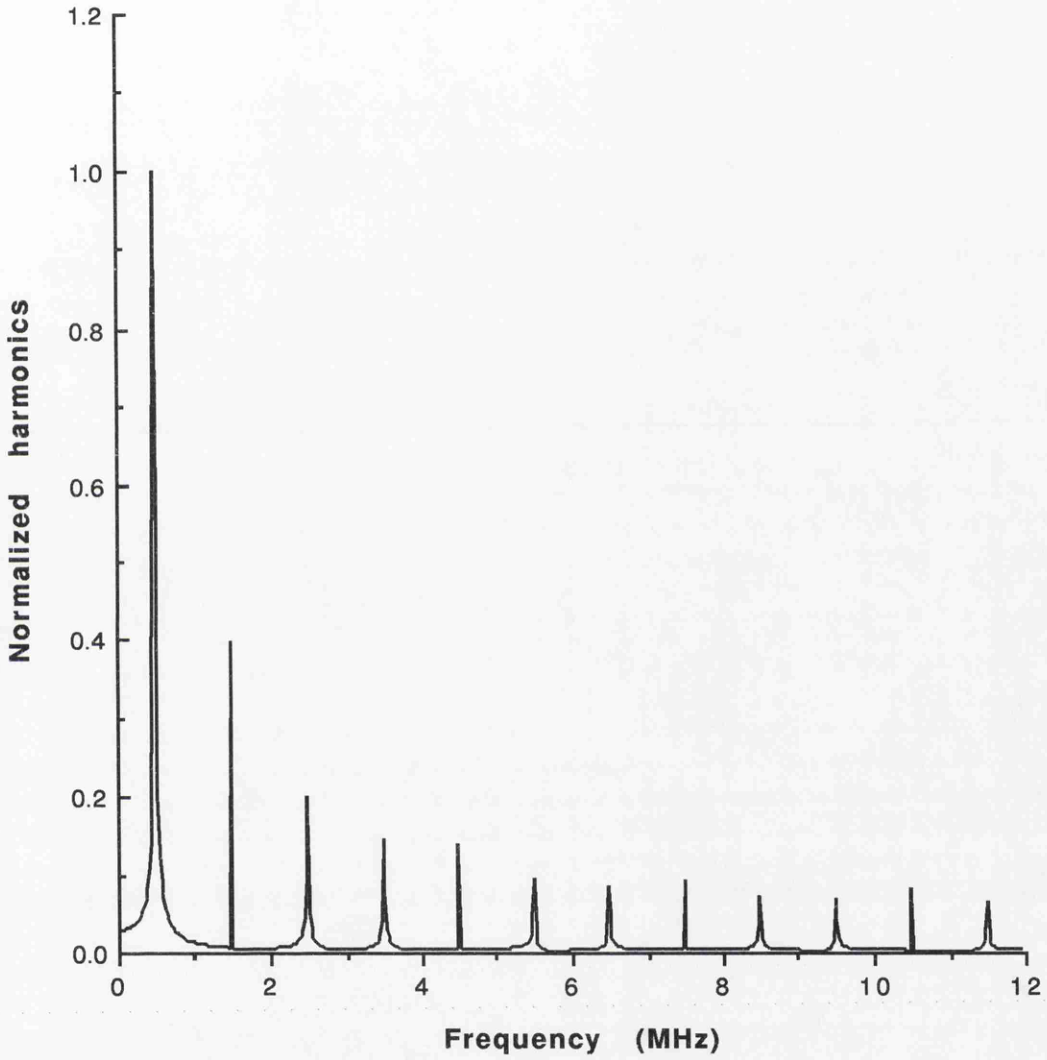


Fig. (5.6) FFT spectrum for 5th-order lowpass SI filter with slew-rate limits: 500kHz excitation

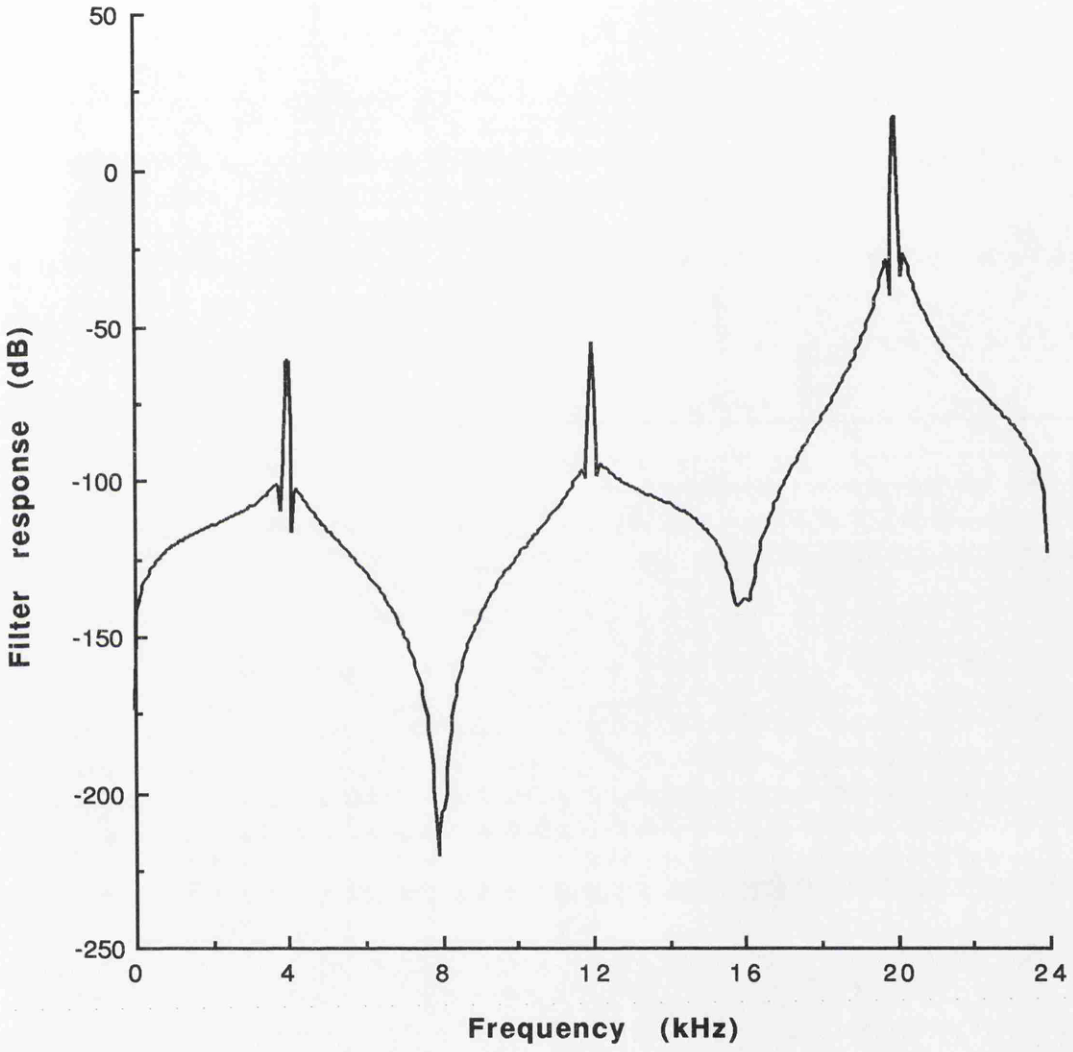
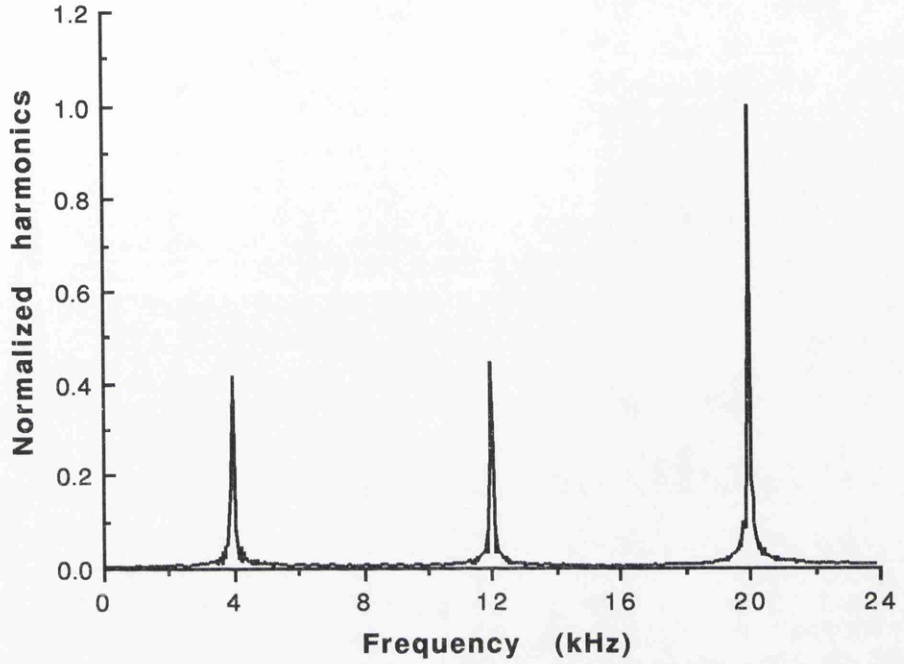
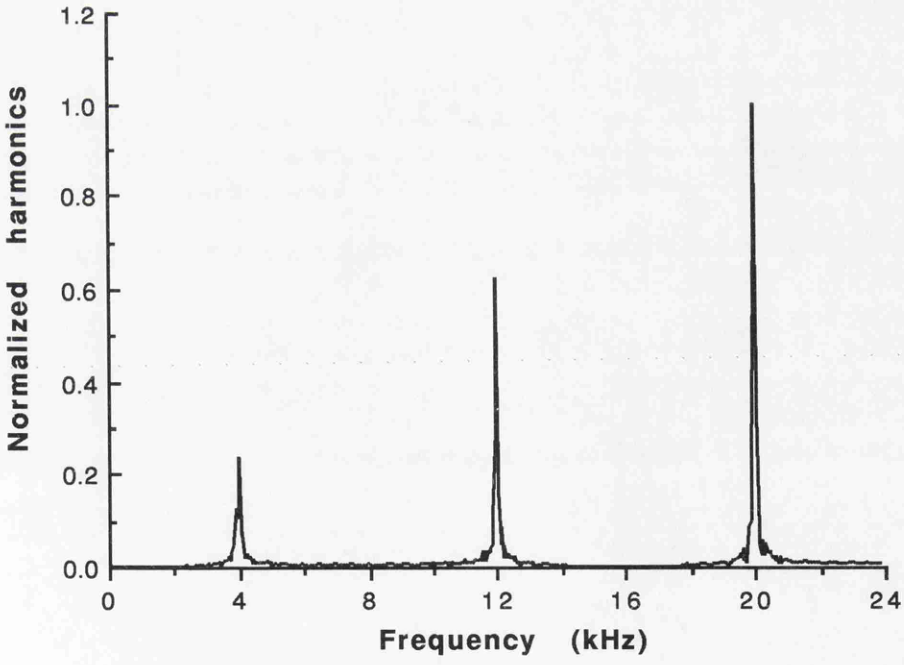


Fig. (5.7a) Linear frequency response of a frequency translated SC filter



(a)



(b)

Fig. (5.7b) FFT spectrum components
(a) linear operation; (b) with amplifier saturation

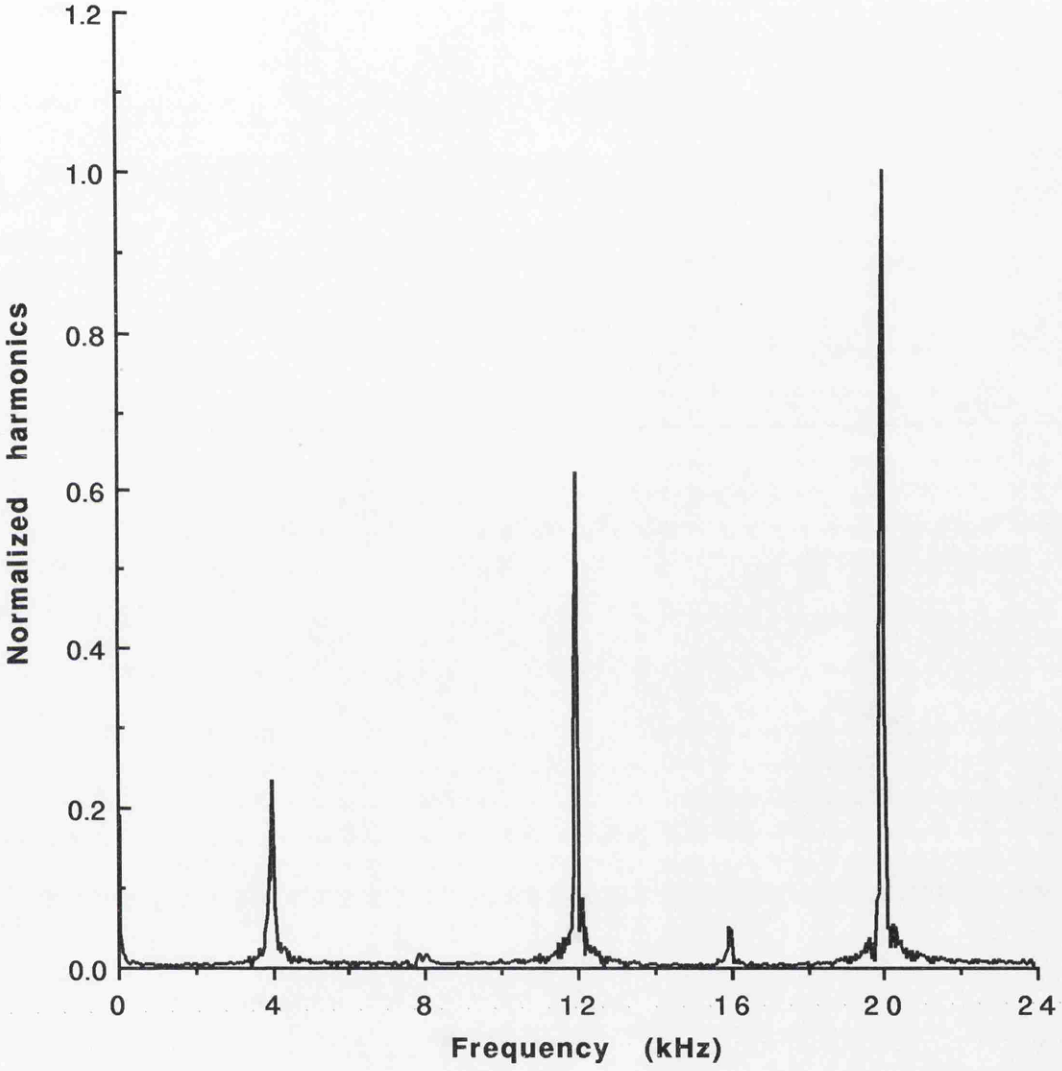


Fig. (5.7c) FFT spectrum with asymmetric amplifier saturation

5.4 SUMMARY

In this Chapter, efficient time domain simulation techniques are presented for SC and SI networks containing elements whose general monotonic nonlinear characteristics can be modelled by piecewise linear equations (this will apply to slew-rate, saturation and common signal dependent effects). General nonlinearities in both SC and SI circuits are described. The original nonlinear network is replaced by a piecewise linear network and by using a modified Katzenelson algorithm, all the sparse matrix and interpretable code generation features employed in existing simulators can be utilised to maximum advantage. Computer algorithms are given and implementation is organised to retain maximum efficiency. Numerical examples show the excellent performance of the proposed method.

It is readily acknowledged that this technique can only produce a limited analysis of a few dominant nonlinearities. If the number of piecewise linear elements is increased to give a better representation of device nonlinearities then severe penalties in computation time result and numerical uncertainties in the solutions increase. For more accurate simulation of moderate device nonlinearities, in highly linear amplifier design for instance, high order polynomial approximations are required [16]. Harmonic balance techniques[17] are also important. If the computation of severe nonlinear behaviour, such as jump-resonance is required, then describing-function and other techniques are needed [18] and the computing requirements escalate.

The conclusion of this work is that highly efficient algorithms for the linear analysis of switched networks can be easily adapted to perform the analysis of dominant nonlinearities in SC and SI networks giving acceptable results without recourse to major computing resources.

REFERENCES

[1] Z. Q. Shang and J. I. Sewell, "Efficient sensitivity analysis for large non-ideal switched capacitor networks, " Proc. IEEE ISCAS, Chicago, May 1993, pp.1405-1407

[2] Z. Q. Shang and J.I. Sewell, "Efficient noise analysis methods for large non-ideal SC and SI circuits," Proc. IEEE ISCAS, London, June 1994, pp.5.565-5.568

[3] R. J. Trihy and A. Rohrer, "A switched capacitor circuit simulator: AWEswit," IEEE J. of Solid-State Circuits, Vol., 29, no.3, Mar. 1994, pp.217-225

[4] C. Toumazou, J. B. Hughes and N. C. Battersby, *SWITCHED-CURRENTS an analogue technique for digital technology*, Peter Peregrinus Ltd. 1993

[5] L. O. Chua and P. M. Lin, *Computer Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques*, Prentice-Hall, 1975

[6] J. Vlach and K. Singhal, *Computer methods for circuit analysis and design*, Van Nostrand Reinhold Co., New York, 1983

[7] L. W. Nagel, "SPICE2, A computer program to simulate semiconductor circuits," Memo UCB/ERL M520, Univ. California, Berkeley, May 1975

[8] C. Visweswariah, and R. A. Rohrer,"Piecewise approximate circuit simulation," IEEE Trans. on Computer-Aided Design, vol. 10, no. 7, July 1991, pp.861-870

[9] J. B. Hughes, private communication, Philips Semiconductor Ltd., 1994

[10] L. B. Wolovitz and J. I. Sewell, "General analysis of large linear switched capacitor networks", Proc. IEE , Pt. G, vol. 135, no. 3, June 1988, pp.119-124

[11] J. Katzenelson, "An algorithm for solving nonlinear resistive networks", Bell Syst. Tech. J., vol. 44, Oct. 1965, pp.1605-1620

[12] Z. Q. Shang and J. I. Sewell, "Efficient analysis of some non-linearities in SC and SI filter networks," Digest of the 14th IEE Saraga Colloquim on "Digital and analogue filters and filtering systems", London, 1994, pp.10/1-10/5

[13] Belen Perez-Verdu, Jose L. Huerstas and Angel Rodriguez-Vazquez, "A new nonlinear time-domain op-amp macromodel using threshold functions and digitally controlled network elements", IEEE Journ. of Solid-State Circuits, vol. 23, no.4, August 1988, pp.959-971

[14] Lu Yue and J. I. Sewell, "A systematic approach for ladder based switched-current filter design", to be published in ISCAS' 95.

[15] J. E. Franca and D. G. Haigh, "Design and applications of single-path frequency-translated switched-capacitor systems," IEEE Trans. on Circuits and Syst., vol. CAS-35, no.4, pp.394-408. April 1988

[16] D. R. Webster, D. G. Haigh and A. E. Parker, "Distortion compensation of multi-MESFET circuits," Proc. IEEE ISCAS, London, May 1994, pp.5.189-5.192

[17] C. Nguyen, C. G. Christodoulou, "An efficient implementation of the harmonic-balance technique for solving nonlinear microwave problems," International Journal of Infrared and Millimeter Waves, Vol. 13, No. 5, 1992, pp.733-750

[18] P. Bowron and A. P. O'Carroll, "Evaluation of the large-signal behaviour of active filters," Proc. IEEE ISCAS, Finland, Jun. 1988, pp.2419-2422

CHAPTER 6

MIXED - MODE SIMULATION

6.1 INTRODUCTION

Signal processing is concerned with data acquisition and data processing. For data acquisition, the conventional approaches of implementing the analogue to digital (A/D) conversion function requires high-precision analogue components, consequently, A/D converters are usually implemented using special integrated circuit processes which complicates the system implementation. Oversampling techniques were proposed to overcome above difficulties by offering the property of insensitivity to component variations and compatibility with VLSI technology. The recently emerged switched current technique [1-3] makes the implementation of both the digital and analogue signal processing circuitry associated with and oversampling A/D converters in fully monolithic form ever more feasible. The trend of fabricating analogue and digital circuit on one chip is apparent and a large class of the mixed analogue/digital circuits can be classified as switched network (switched capacitor or switched current) / digital networks. The increased complexity demands even more efficient simulation tools. The major challenge is posed by the oversampled sigma-delta data converters which are difficult to simulate, since a large number of clock cycles are required in order to obtain meaningful measures of signal to noise ratio (SNR) or signal to distortion ratio (SDR). Traditionally most existing mixed-mode simulators employ SPICE like methods to handle analogue circuits. A transient analysis of a single clock cycle of the switched capacitor (SC) or switched current (SI) integrator may typically take several minutes of CPU time on a modest workstation. Detailed SNR curves are almost impossible to obtain due to the horrendous CPU time cost. Alternatively, special purposed simulators based on difference equation models of the integrators can reduce the simulation dramatically, however they can never provide any simulation of the combined effects of many nonidealities in the circuit.

An efficient technique which has been successfully applied in time domain analysis of switched linear networks, is shown also to be an excellent vehicle for the processing of SC or SI circuits involved in mixed-mode simulation. The analogue circuit topology can be modified by either periodical clock signal or nonperiodical logic feedback. By introducing the concept of the network state, only one calculation of the extended state transition matrix per state is required. The overall mixed-mode simulation system consists of two simulators which communicate with each other through a suitable interface. A number of techniques aiming to exploit the maximum efficiency are proposed and the corresponding computer implementation is demonstrated.

6.2 GENERAL MIXED-MODE SIMULATION STRATEGIES

The mixed-mode simulation techniques can be broadly classified into three different groups.

6.2.1 One algorithm based approach

The main feature of this group is that it uses a unique algorithm to deal with both analogue and digital circuit. A typical simulator which adopted this approach is SPICE[4]. It handles the digital logic by treating it as an analogue function at the transistor level. By ignoring the specifically digital characteristics of the digital circuit, it produces far more detailed waveforms for the digital portion than digital simulation generally requires and inevitably consumes a large amount of CPU time.

Derivatives of this approach are to use one simulator (analogue / digital) to perform both analogue and digital simulation [5-6]. If the core simulator is digital, then macromodels for analogue components can be appended, for an analogue core simulator, the logic function are interpreted as pseudo-analogue devices which introduce extra states into the simulation. However, the circuit is still simulated at the analogue / digital level. Limitations include long efforts of macromodeling, manual changes of the original netlist, and insufficient simulation accuracy. If the core simulator is analogue one, this leads to excessive accuracy but loss of efficiency. When a digital simulator is used as the core, the simulation is quite efficient but at the cost of losing accuracy.

6.2.2 Glued simulator approach

Designers are used to manually splitting the mixed-mode circuit design into analogue and digital portions, which are then verified separately using analogue and digital simulators, respectively. It is therefore quite natural to combine two simulators together to fulfil the task of mixed-mode simulation. The so called glued simulator approach usually requires the separate analogue and digital simulators to be run concurrently and the communication between simulators can be achieved by using an interprocess communication mechanism [7-10]. The effectiveness of this approach is restricted to the situation where only unidirectional elements are connected across the mixed-mode interface. It requires special procedures to handle the mismatch in voltage and impedance levels between two simulators and may also require time backtracking. In general the efficiency is dependent upon the analogue simulator and the overhead of communication between simulators.

6.2.3 Unified integrated approach

The unified integrated approach usually requires the incorporation of various simulation algorithms into one simulator to allow analysis of different parts of a partitioned circuit using different models and algorithms. It is believed to be the most flexible and most efficient approach for mixed-mode simulation. A number of simulators [11-23] based on this approach have been developed. Generally, the architecture of the simulation level in these simulators can be illustrated in Fig. (6.1)

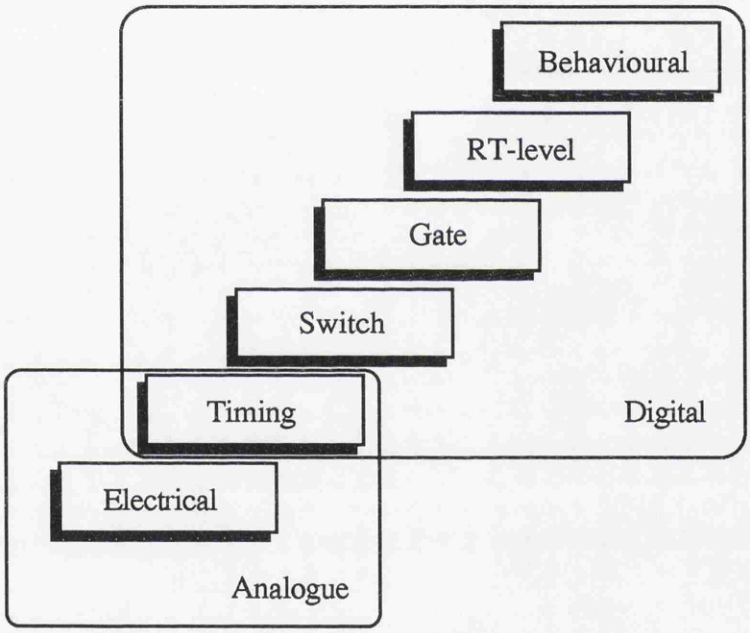


Fig. (6.1) Architecture of simulation level

Algorithms for different simulation levels can be integrated into one simulator with a unified simulation environment. It is feasible to have multi-level simulation in mixed-mode simulator. However, the increasing level of model abstraction increases the efficiency of simulation, but can reduce the accuracy. It is also evident that a number of alternative algorithms are available for digital simulation, but choice for analogue circuit simulation can only be SPICE type algorithms and timing simulation methods. Although the analogue portion of an integrated circuit is usually very small compared to its digital counterpart, the simulation is generally more time consuming and error prone. The overall performance of a mixed-mode simulator is very dependent on the algorithms for the analogue analysis.

6.2.4 Overview

The general mixed-mode simulation strategies have been reviewed and the task of simulating large and complex digital circuits precludes the possibility of using one of the single algorithm based simulators. It can be concluded that a unified integrated approach is the most suitable strategy for mixed-mode simulation purposes in general sense. However, since this approach aims to deal with general mixed-mode circuits, the analogue simulation methods should be capable of meeting the general requirements. For a special class of mixed-mode networks which consist of switched linear circuits and digital networks, the general analogue algorithms fail to take the advantage of the algebraic nature of network equations of switched linear networks. The requirement for computing a large number of sharp waveform transitions makes the general algorithms very inefficient. In the case of oversampled sigma-delta coders, it is almost impossible to use the unified approach mixed-mode simulators.

A number of special purpose simulators for oversampled A/D data converters have been developed. Most of them are based on difference equations and describe the network at behavioural level and high efficiencies have been reported. Unfortunately, computation of the effects of general nonidealities of the circuit are not possible by this method. It should theoretically be possible to model each individual effect of nonideality (for finite opamp gain perhaps) using difference equations, in practise however, the behavioural model for the combination of these effects is extremely difficult to obtained. In addition, mapping to circuit is still not transparent.

The glued simulator approach, although limited in scope of application is a very effective alternative when only unidirectional elements exist at the mixed-mode interface. It is also feasible to incorporate efficient algorithms to solve a particular class of problems for which the unified approach cannot provide the best solutions. This approach will be examined further.

6.3 MIXED SWITCHED AND DIGITAL NETWORKS ANALYSIS

6.3.1 Network architecture

In practice, mixed-mode circuit architectures can be very different depending on their applications. In the following, attention will be focused on the class of mixed

switched capacitor (SC) or switched current (SI) and digital networks. The general architecture is illustrated in Fig. (6.2).

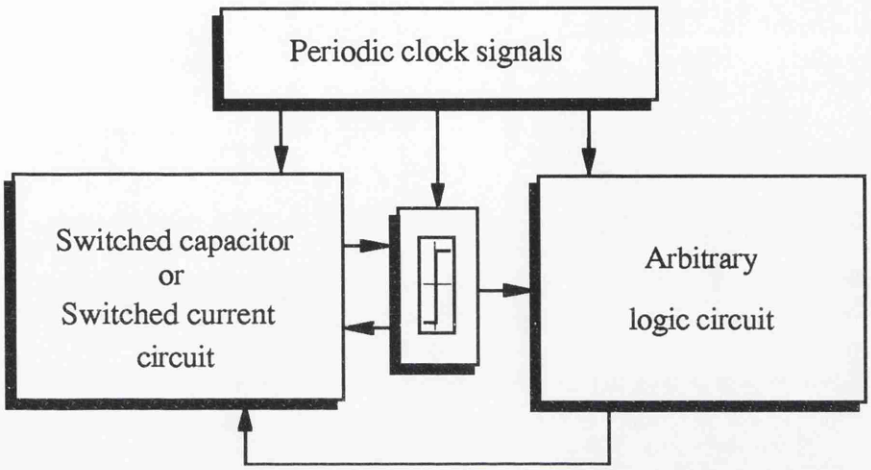


Fig. (6.2) Mixed-mode network architecture

The switched network could be either SC or SI circuit. Most of the analogue components are similar to those in pure SC or SI networks except that the switches can now be controlled by either periodic or nonperiodic signals.

The interface consists of latched comparators which translate the continuous analogue signal into a series of discrete binary states which feed into the digital environment. In SCNAP5, the comparator is implemented with built-in latched facility as shown in Fig. (6.3)

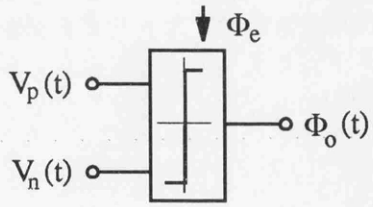


Fig. (6.3) Latched comparator

The comparator only changes its output Φ_o at the rising edge of the enable signal Φ_e when its input signal cross the threshold, otherwise the output will be latched.

The digital network can contain a wide variety of logic elements. There are many commercial logic simulators available to handle extensively large and complex digital networks. The choice of logic simulator for mixed-mode application is mainly determined by the complexity of the interface required with any analogue simulator. In our case the choice was dictated by an industrial sponsor! The digital network is analysed by Verilog[24] which is a very powerful commercial package, and any

arbitrary combination and sequential logic circuits can be described. The digital network is driven by the comparator output and generate some binary signals which may feed back to control the switches in the analogue circuit.

6.3.2 Non-periodic switched networks

A network state is defined as a unique combination of all clock states and time interval. Mathematically, it can be expressed by

$$\chi_j = (\psi_l, \tau_k) \tag{6.1}$$

where τ_k is the kth time interval, ψ_l represents the lth combination of clock states and defined by

$$\psi_l = (\Phi_1, \Phi_2, \cdots, \Phi_n) \tag{6.2}$$

and Φ_i indicates the ith clock state (1 or 0).

Since the binary feedback from the digital network depend on the comparator output, the subset of switches controlled by these signals are operated non-periodically. Most of the existing SC software packages are specially designed for periodic clocks, and hence are not directly applicable to this case. Substantial modifications are required.

Consider an arbitrary switched linear network at interval $(t_{k-1}, t_k]$ over the time period of interest, it can be represented by the algebraic-differential system,

$$C_k \dot{v}_k(t) + G_k v_k(t) = w_k(t) \tag{6.3}$$

where C_k and G_k are capacitance and conductance matrices in the kth time interval, respectively.

By using polynomial approximation for the excitation, the solution for (6.3) can be written as

$$v_k(t) = P_k(t) v_{k-1}(t_{k-1}) + \sum_{j=1}^{q} \sum_{i=0}^{m} \alpha_{i,k}^{j} B_{i,k}^{j}(t) \tag{6.4}$$

where q is the number of independent sources, and for the other notation refer to section 3.2.5.

In the case of periodically switched linear networks, the number of unique network states can be pre-determined. Therefore P_k and $B_{i,k}$ can be pre-calculated. Inspection

of equation (6.4), reveals several observations which can be made directly. The solution of (6.4) only requires m+2 matrix-vector multiplications. Normally the excitation vector is very sparse, the computation in reality is then reduced significantly and is very favourable when compared to the cost of matrix solution. Another observation is that equation (6.4) is quite general and does not imply periodic operation of the network. Hence it is feasible to extend the application of equation (6.4) to incorporate the non-periodic situation.

Instead of calculating P_k and $B_{i,k}$ at the pre-processing stage, they are calculated on demand. The strategy described in section 5.3.2 is equally applicable to these circumstances. Before computing equation (6.4), the network state is checked and when a new state is found, corresponding P_k and $B_{i,k}$ are calculated and stored in the matrix cache, otherwise the P_k and $B_{i,k}$ will be retrieved from the matrix cache. The above strategy may be summarised by the following computer algorithm,

```
        /* non-periodical network */
ST:     get time step;
        if(time > Tstop) escape;
        get each clock state & form network state;
        check network state;
        if(state exists)
            get corresponding P_k and B_i,k from cache;
        else
            calculate P_k and B_i,k and store in cache;
        solve equation (6.4);
        output v_k;
        goto ST;
    }
```

It is evident that only one calculation for P_k and $B_{i,k}$ per network state is required, hence the new algorithm should retain the efficiency observed in the SCNAP4 application.

The efficiency was further tested over a number of switched capacitor networks and listed in Table (6.1)

Table (6.1) Comparison between two algorithms

circuit name	circuit size	time slots	time points	periodic		non-periodic	
				calculate P_k, $B_{i,k}$ (sec)	time domain (sec)	calculate P_k, $B_{i,k}$ (sec)	time domain (sec)
but3.in	13	3	100	0.09	0.06	0.09	0.14
nos5.in	24	4	101	0.38	0.12	0.35	0.49
bp6.in	30	2	2501	0.32	3.64	0.33	4.12
fet7.in	36	6	151	0.78	0.31	0.83	1.17
lp10.in	60	† 4/16	1601	6.56	10.57	1.82	12.49
spft.in	102	†20/36	5761	50.98	105.05	27.6	133.9

Note: † The number of unique states is less than the number of time slots;
Above data obtained on SUN-Sparc ELC station

It can be seen that the time cost for calculating P_k and $B_{i,k}$ is almost the same for both algorithms. In some cases, the nonperiodic algorithm produces even better results, since no redundant states needs to be calculated. For time domain simulation, the non-periodic algorithm takes a bit longer time due to the overhead of detecting the network state and storing/retrieving matrices.

6.3.3 Communication between two simulators

The mixed-mode simulation system structure can be illustrated as follows,

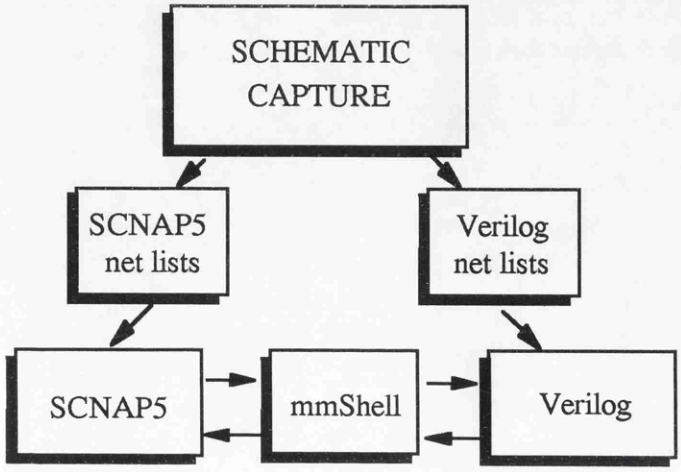


Fig. (6.4) Structure of mixed-mode simulation system

The mixed switched linear circuits and digital networks are partitioned into two portions, analogue and digital. Schematic capture produces corresponding netlists for SCNAP5 and Verilog, respectively. Comparators will be involved in the analogue portion. The two simulators communicate to each other through mmShell (Mixed-mode shell). The master simulator will send/receive messages to/from the slave simulator by spawning a series of subprocess [25]. The procedure may be illustrated by means of a typical situation shown in Fig. (6.5)

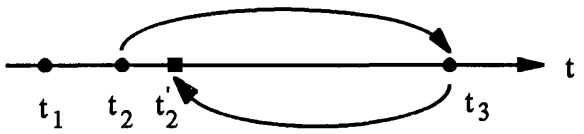


Fig. (6.5) Time advancement mechanism

Assuming the output of the comparator changed at t_2, SCNAP5 would send a message to Verilog informing that one of the digital inputs was changed at t_2. Verilog would evaluate the digital circuitry and find that a new event occurs at t_2', say. However Verilog does not send back the new event until the time is advanced to the next time point t_3. When SCNAP5 receives the message from verilog at t_3, the time wheel is set back to t_2' and the event table is then revised in order to put in the new event. Consequently, SCNAP5 would take the new event into account and recalculate the response of the analogue circuitry at t_2'.

6.3.4 Computer implementation

Efficient computer implementation is essential to the performance of the software. During the development of SCNAP5, the following issues have been considered.

(a) Network state expression

The concept of network state is very simple. it should be able to uniquely define the network characteristics. During a time interval there should be no change in state. Since the detection of network state during simulation is an extra overhead compared to SCNAP4 operation, it is essential to express the network state in a computationally efficient manner.

The following discussions are specifically refer to C programming language facilities. The network state can be defined as

```
typedef struct {
    type1 clock_state; /* combination of all clock states */
    type2 delta; /* time interval */
} network_state;
```

Two data types for the *clock_state* are considered. The long integer can be used, as each bit denotes either 1 or 0 which represents high or low of a particular clock. Bit operations in C, such as &, |, ^, <<, ~ are very efficient. However, for a long integer, only 32 bits are available. It is also possible to use an integer array to overcome the limitation of using a single integer variable. The *clock_state* can also be expressed by using character string. In this case each bit is represented by character '1' or '0' and the string comparison is also very simple.

An experiment trying different data types to express network state has been carried out. The procedure can be summarised by the following algorithm,

```
/* testing different data types */
loop (all time points) {
    loop(all clocks) {
        current state = getstate();
    loop(list of network state) {
        check with current state;
        if (state exists)
            report(state exists);
        else
            store current state;
    }
}
```

The experiment was carried out over 1,000 time points and 100 unique network states with a number of clocks from 10 to 100. The results are illustrated in Fig. (6.6)

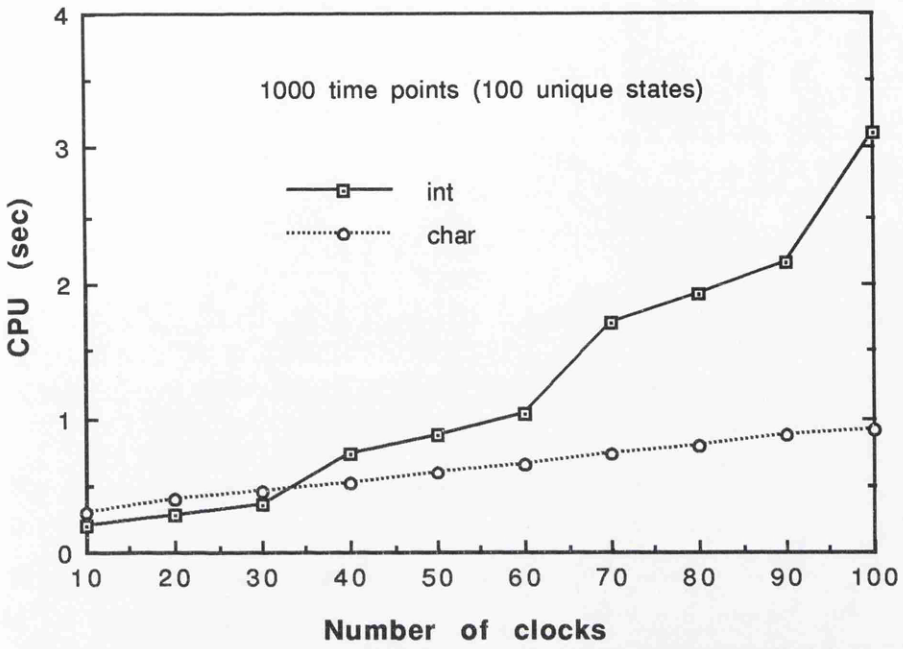


Fig. (6.6) Comparison of two different data types

It can be concluded that for the situation when the number of clocks is less then 32, it is more efficient to use an integer to identify the clock state. In general, a character string identifier is a better choice, hence this was adopted in SCNAP5.

(b) Breakpoint table design

The breakpoint table is very important for mixed-mode simulation. It might be revised frequently due to the digital to analogue (D/A) events generated from digital side. However, the traditional strategy of the breakpoint table design is not adequate for our purpose. This is mainly because most of the breakpoints in a switched linear network are periodically repeated, only the D/A events are random on the time axis. It is not efficient to follow the traditional scheme which checks all the time varying instances for breakpoints and then revises the breakpoint table when required. A new strategy was proposed and implemented in SCNAP5.

The breakpoint table is built for each time slot. The fundamental breakpoints in a particular slot are known and can be pre-set. If, during that particular time interval, some D/A events occur, then the breakpoint table is revised. The advantage of this strategy is that the breakpoints due to the periodic clocks can be pre-processed which eliminates the need to check and produce them at each time point. Although this technique is specially designed for mixed-mode simulation, it is also compatible with pure analogue SC or SI filter simulation.

(c) Evaluation of periodic and non-periodic clock states

The processing of a clock signal normally employs a series of operations which locate the timing information of the clock at a particular time point and determine the clock state (either open or closed). In the case of periodic clock signals, since the clock waveform is systematically repeated, the clock state at any time point can be referred to a certain state in one period. The next clock state is predictable. However, for a non-periodic clock signal (generated from digital output), such prediction is no longer possible and the only way to work out a clock state is to process the corresponding timing information around the time point of interest. However it is not particularly efficient for all the clock signals to share a unified processing procedure to determine their states. In SCNAP5, periodic clock signals are still pre-processed and used to form the fundamental events in the breakpoint table. Since the time interval of the breakpoint table is identical to one slot which was generated by periodic clocks, the states of the periodic clocks are still predictable, only the non-periodic clock signals have to recourse the general processing routines. By this strategy, a considerable reduction in computational cost results.

6.3.5 Computer examples

A number of circuits are now examined by SCNAP5 to demonstrate the performance of the proposed methods.

A fully balanced first order sigma-delta A/D converter is considered first. The circuit diagram is shown in Fig. (6.7).

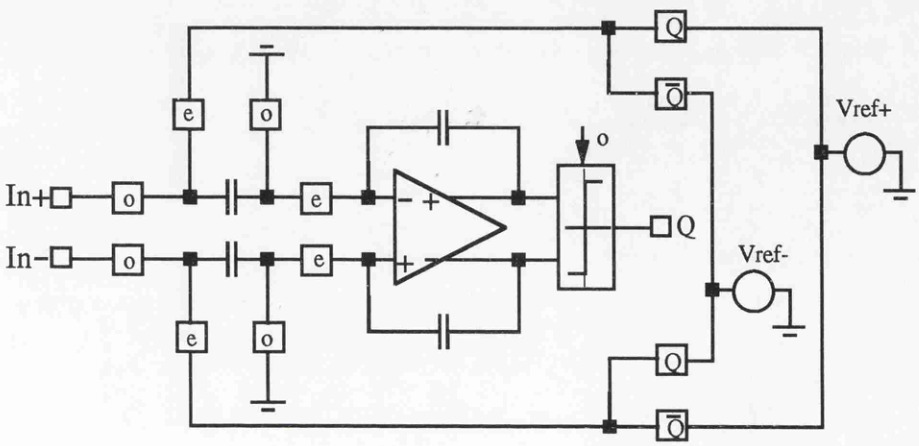


Fig. (6.7) Fully balanced first-order sigma-delta A/D converter

It can be noticed that the positive and negative reference voltage are feedback through two pairs of switches which are controlled by digital signals. A dc source with 0.5 volt amplitude is applied to the input of the circuit. The output waveforms of the opamp and comparator are illustrated in Fig. (6.8) and Fig. (6.9), respectively.

A second-order sigma-delta modulator is illustrated in Fig. (6.10)

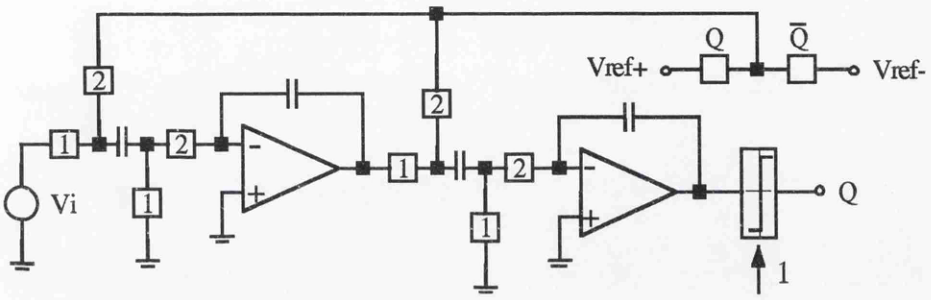


Fig. (6.10) SC implementation of a second-order sigma-delta modulator

A 1kHz sine wave is applied and the clock period is set to 6μ sec. SCNAP5/Verilog running on a SUN-Sparc ELC workstation required about 21 minutes to compute the time response for 66,666 clock cycles. The actual number of time points is up to 172,763. An FFT on 65,536 points produces the spectrum of the comparator output shown in Fig. (6.11). The SNR curve was calculated for different input signal amplitudes and illustrated in Fig. (6.12). It took about 3 days to get 100 points for the SNR curve on SUN-Sparc ELC station. Nonideal effects such as switch resistance and finite opamp gain are examined. It can be seen that when these nonideal parameters deteriorate (large switch-on resistance or low amplifier gain), the noise level increases at low frequencies, which can be noticed in Fig. (6.13) and Fig. (6.14), respectively. The gain bandwidth effect of opamp can also be investigated. The GB product is set to 5 MHz and the resulting SNR curve is shown in Fig. (6.15).

In Fig. (6.16), a fifth order sigma-delta modulator is depicted. A sinusoidal input with 0.2 volt amplitude and 1kHz frequency is applied. The clock is operated at 2MHz. Again for 65,600 clock cycles, it takes about 27 minutes to compute the time response on the SUN-Sparc ELC station. The typical FFT spectrum of the high-order sigma-delta modulator is shown in Fig. (6.17), notice the effects of high order noise shaping - a wider noise suppression band, zero in the noise stop band, sharper transition at the edge of the stop band and lower noise level over the stop band. Considering SCNAP5 performs circuit level, full nonideal time domain analysis for such high-order mixed-mode circuit, the performance is quite remarkable.

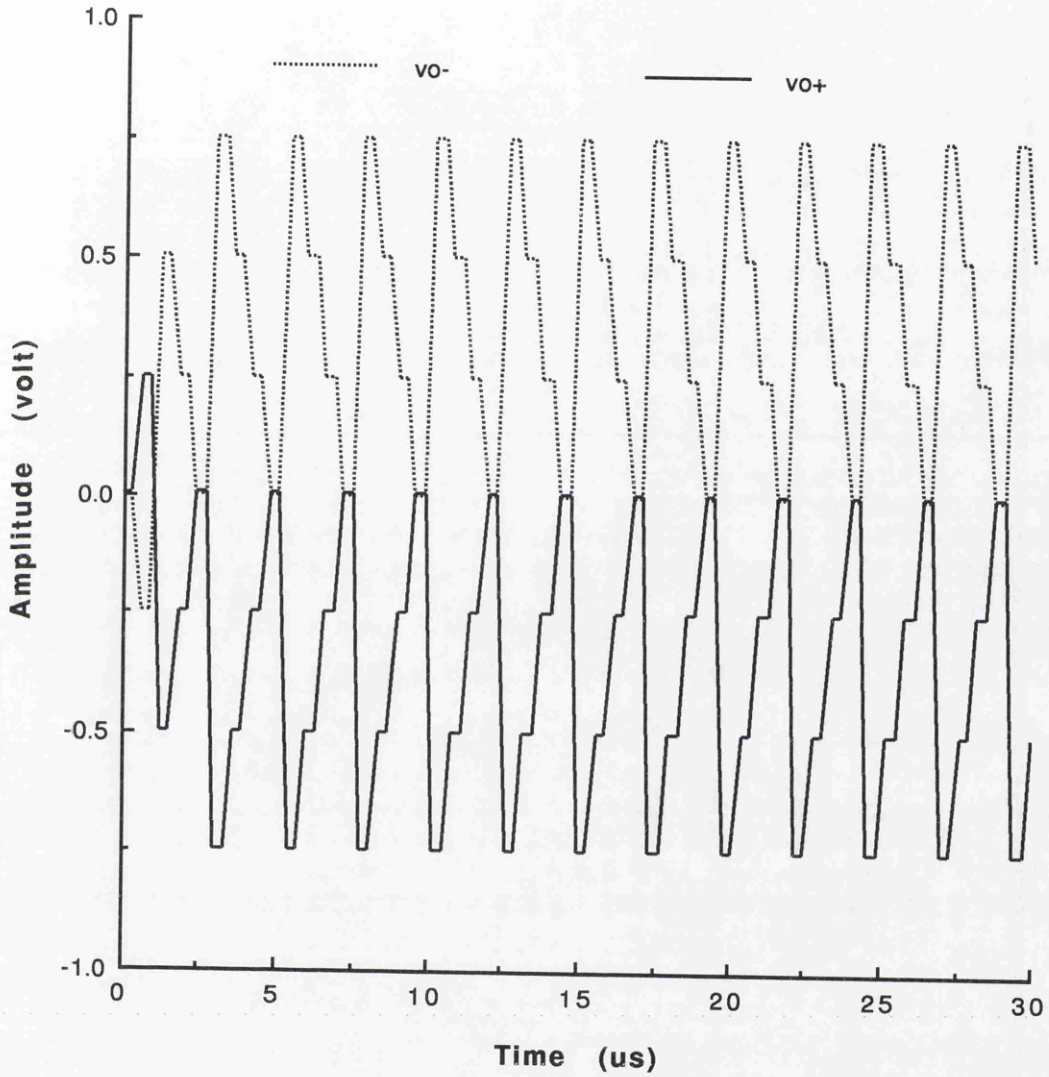


Fig. (6.8) Output waveform of balanced opamp of a 1st-order sigma-delta A/D converter

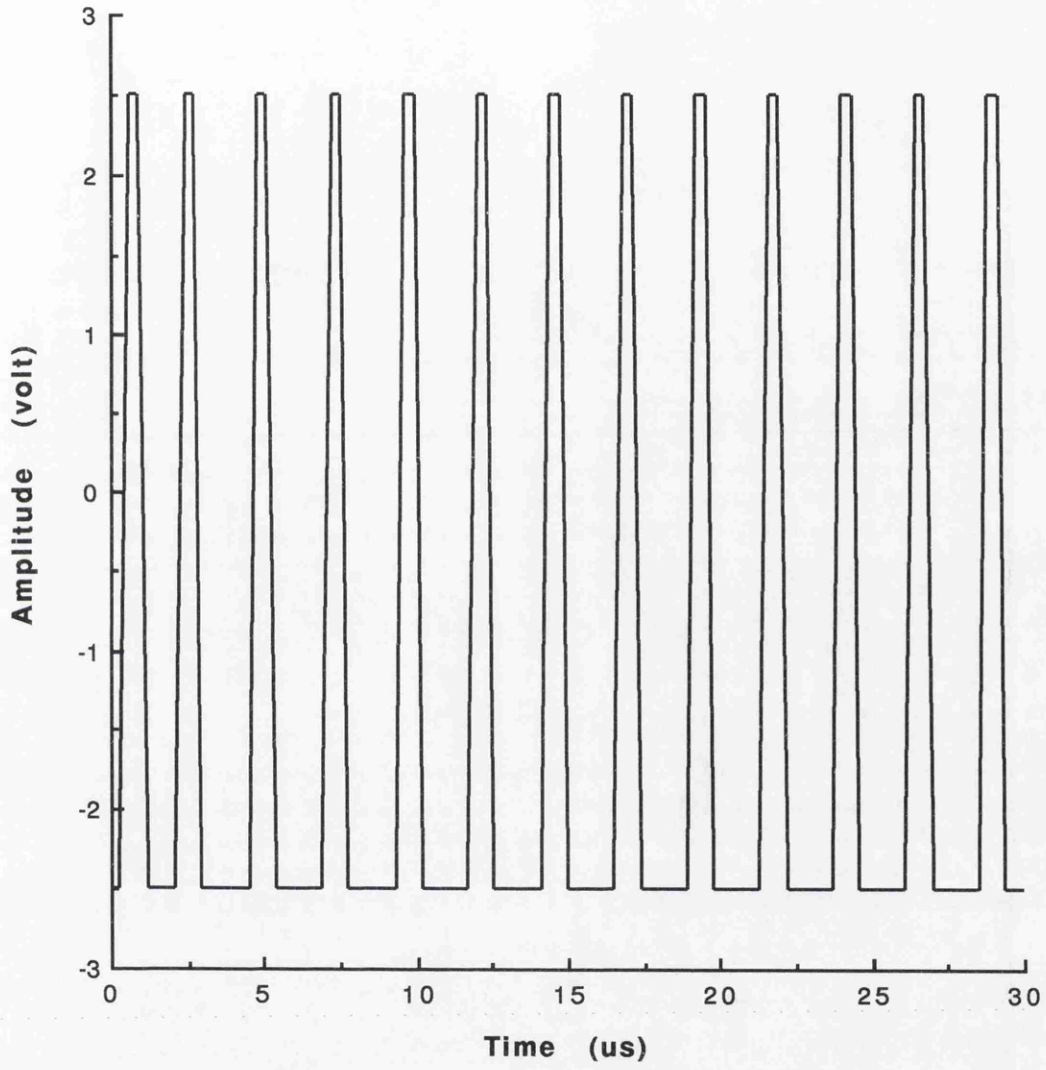


Fig. (6.9) Output waveform of the comparator of a 1st-order sigma-delta A/D converter

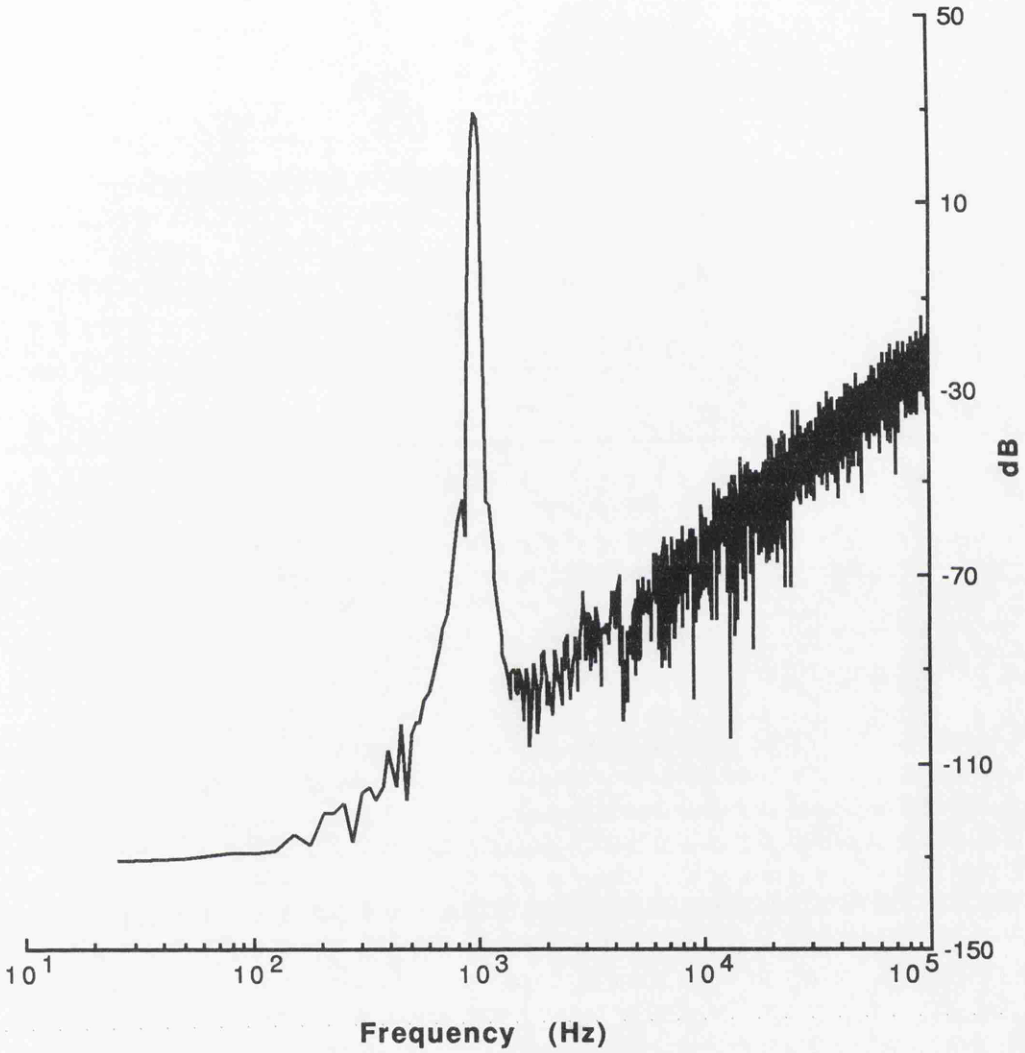


Fig. (6.11) FFT spectrum of the comparator output of 2nd order sigma-delta A/D converter

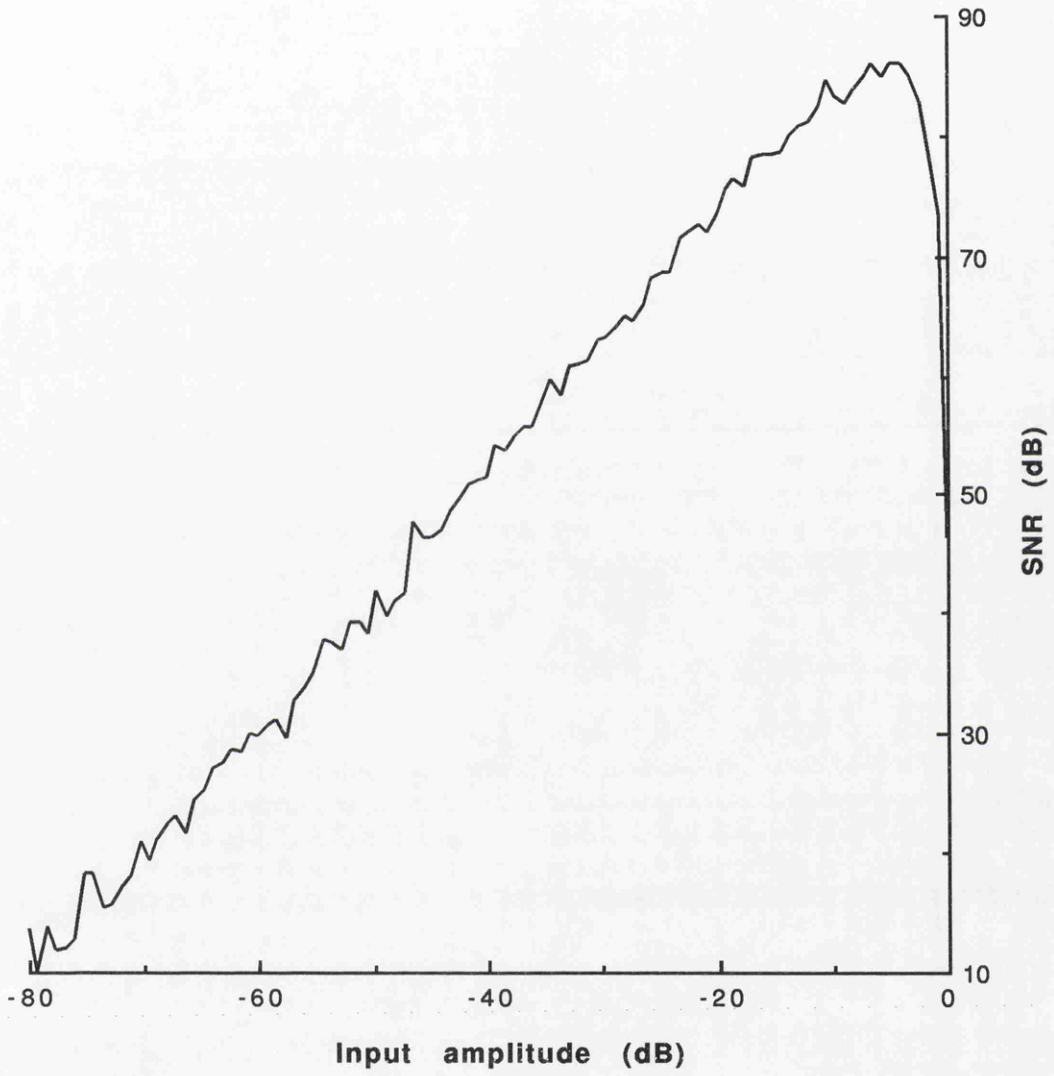


Fig. (6.12) SNR versus relative input amplitude for 2nd-order sigma-delta modulator

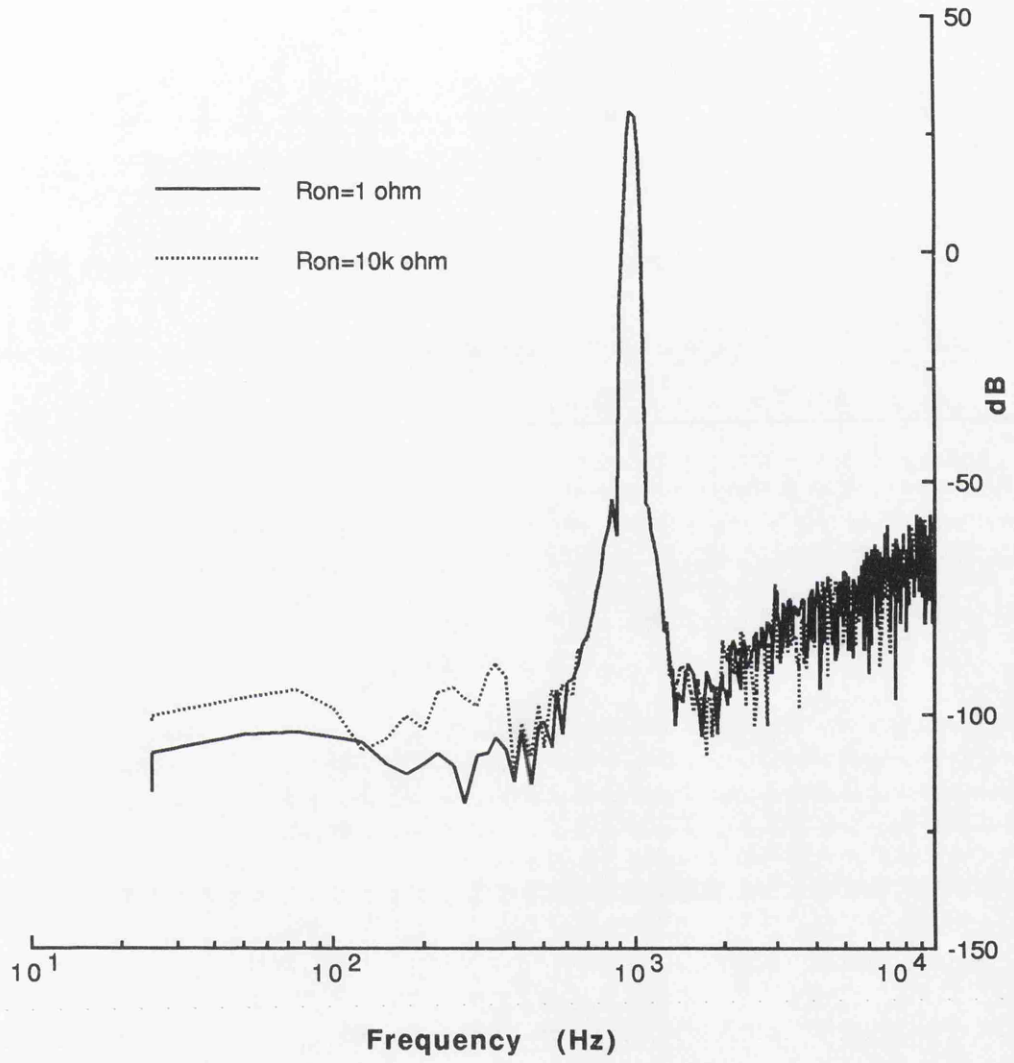


Fig. (6.13) FFT spectrum of 2nd order sigma-delta converter with various switch resistance Ron

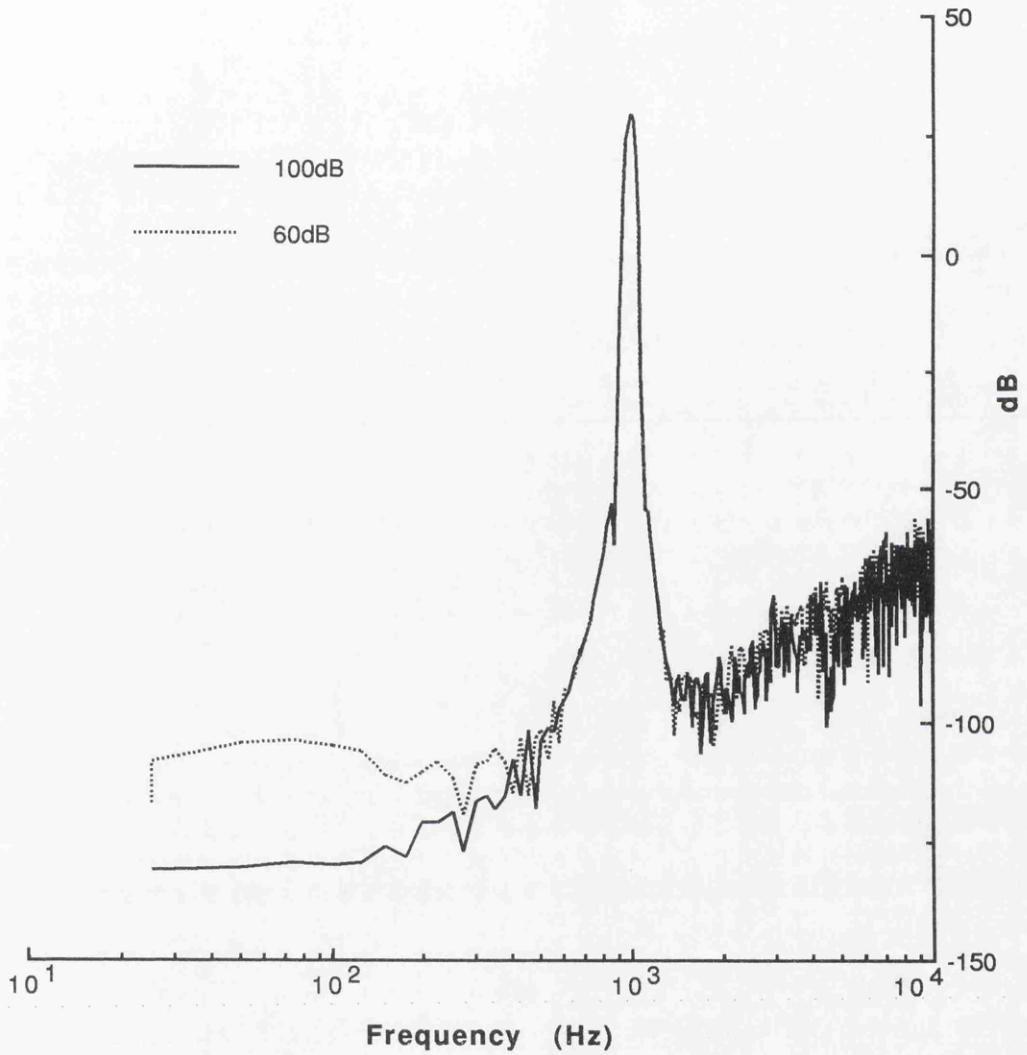


Fig. (6.14) FFT spectrum of 2nd order sigma-delta converter with various opamp gains

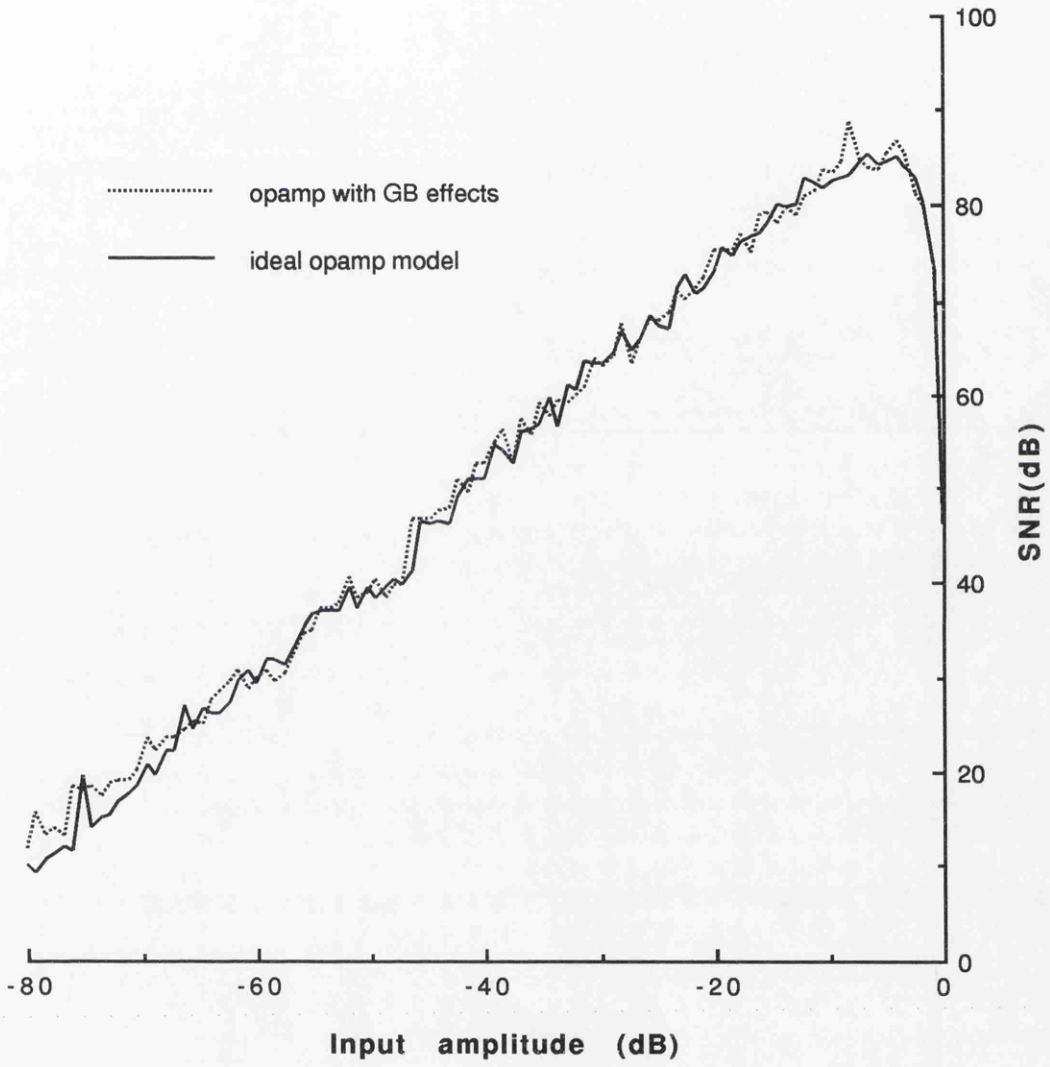


Fig. (6.15) SNR curves with GB effects of opamp for 2nd-order sigma-delta modulator

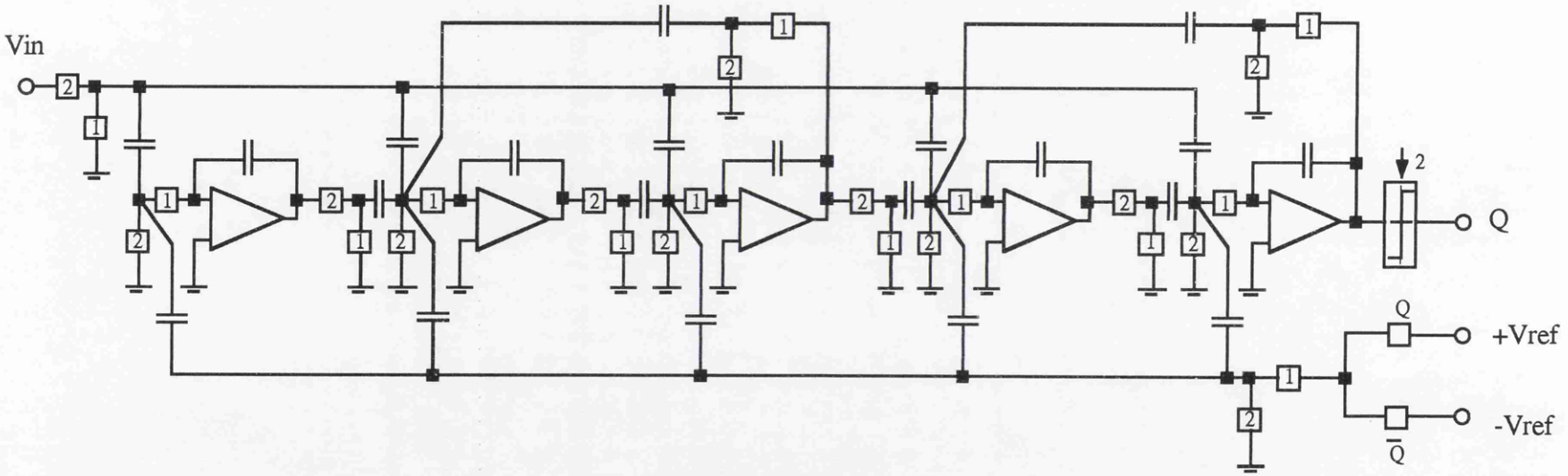


Fig. (6.16) 5th-order sigma-delta SC modulator

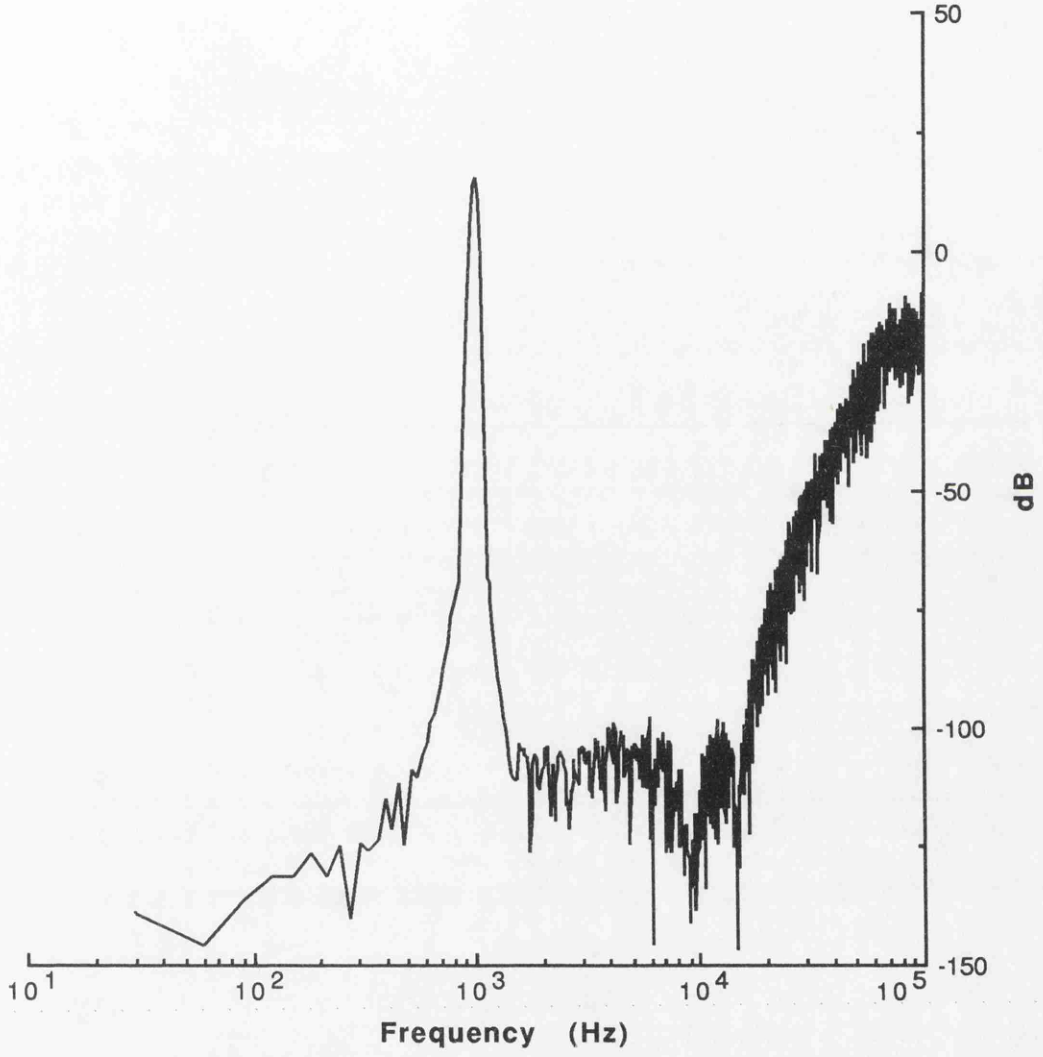


Fig. (6.17) FFT spectrum of the comparator output of 5th order sigma-delta modulator

All the above examples are operated non-periodically. SCNAP5 is also able to handle periodically switched networks. A fifth order elliptic lowpass SC filter is illustrated in Fig. (6.18). The input and output waveforms are shown in Fig. (6.19)

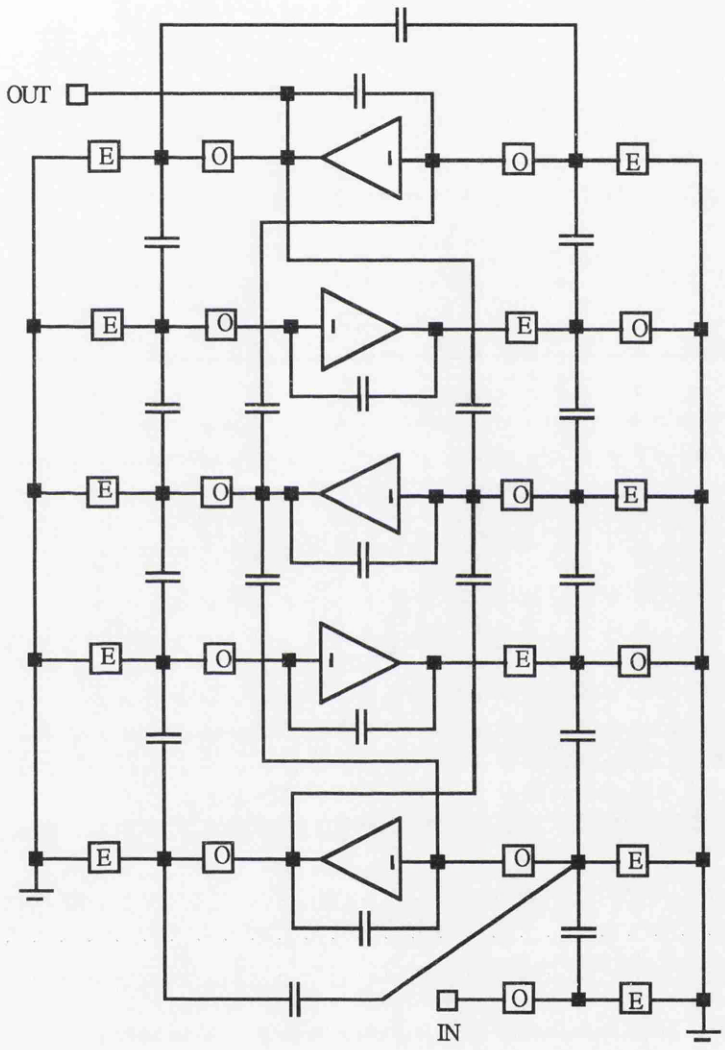


Fig. (6.18) 5th-order elliptic lowpass SC filter

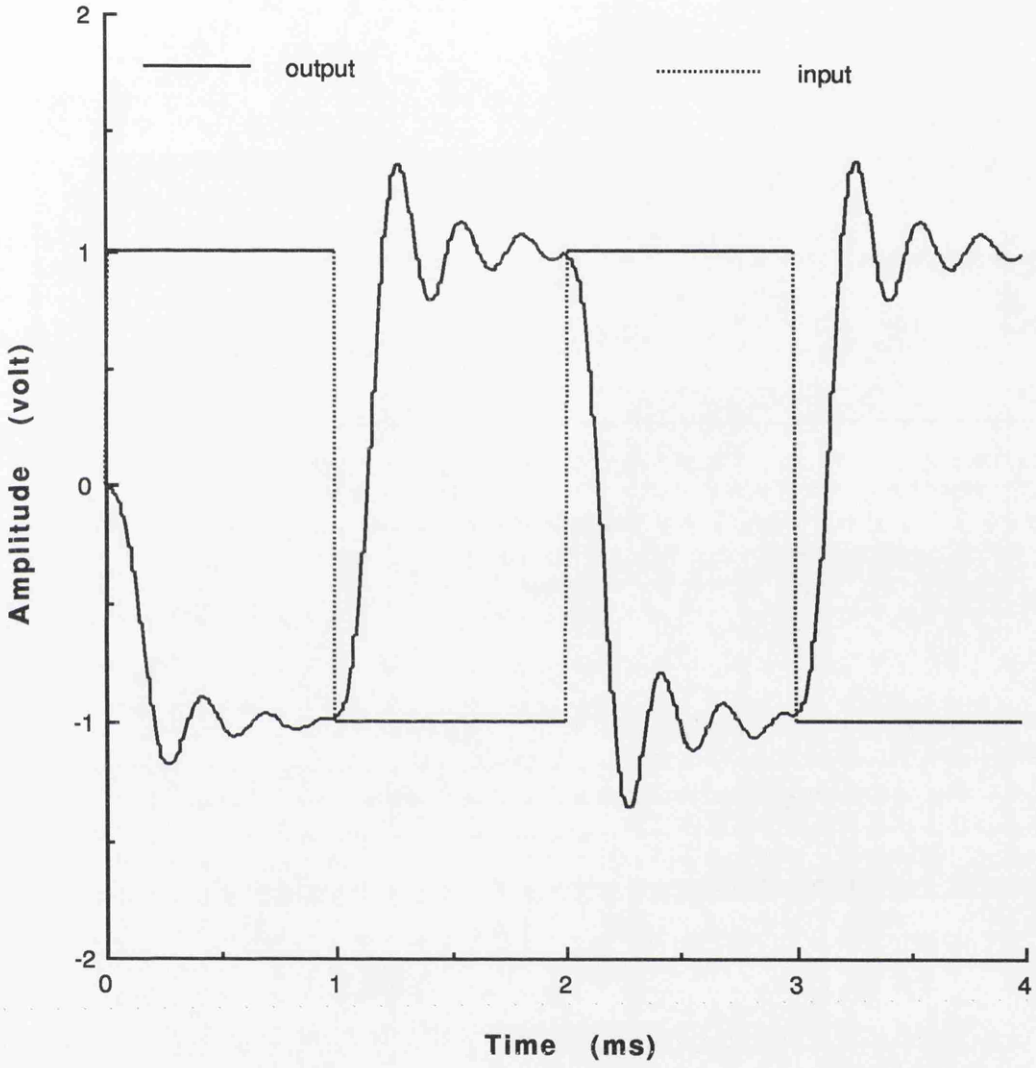


Fig. (6.19) Input and output waveforms of the 5th-order elliptic SC filter

6.4 SUMMARY

An efficient strategy suitable for mixed-mode simulation has been introduced. The analogue circuitry can consist of general nonideal SC or SI networks, or continuous time networks (or even mixture of them!). Hierarchical description of the digital network from primitive logic components to highly abstracted behavioural models is possible. The application to the most time consuming simulation task - analysis of oversampling sigma-delta data converters shows the excellent performance of this approach. A number of effective numerical techniques such as sparse matrix technique, interpretable code generation and numerical inverse Laplace transformation which have been successfully used in the analysis of periodically switched linear network have also been applied to the non-periodic situation. The requirement that switches are allowed to be controlled by either external periodic clock signals or internally generated digital logic is fulfilled by introducing a new definition in terms of network state from which the periodic algorithm can be generalised to non-periodic switched linear networks. The effects of nonidealities which practically exist in practical analogue circuits can be exactly investigated. A number of programming techniques and computer algorithms have been proposed for the objective of gaining maximum simulation efficiency.

REFERENCES

[1] J. B. Hughes, N. C. Bird and I. C. Macbeth, "A new technique for analog sample data signal processing," Proc. IEEE ISCAS, Portland, Oregon, May 1989, pp. 1584-1587

[2] G. Wegmann and E. A. Vittoz, "Very accurate dynamic current mirrors," Electronics Letter, vol. 25, pp.644-646, May 1989

[3] S. J. Daubert, D. Vallancourt and Y. P. Tsividis, "Current copier cells," Electronics Letters, vol. 24, pp. 1560-1562, Dec. 1988

[4] L. W. Nagel, "SPICE: a computer program to simulate semiconductor circuits," ERL Memo UCB/ERL M74/520 May 1975, Univ. of California, Berkeley

[5] P. E. Allen, W. M. Znberek, "Mixed-mode, analog-digital simulation using SPICE-like circuit analysis programs," IEE J. on Semicustom ICs, vol. 8, no. 3, pp.33-45, 1990

[6] R. Chadha and C. F. Chen, "Extension of a digital timing simulator to include first order analog behavior," Proc. IEEE Int. Conf. Computer Design, Rye Brook, 1988, pp.116-119

[7] E. Szeto-Lee and T. F. Fang, "A mixed-mode analog-digital simulation methodology for full custom design," Proc. IEEE Custom Integrated Circuits Conf., Rochester, 1988, pp.3.5.1-3.5.4

[8] J. Benkoski, J. Besnard, S. Gai, M. Magui, E. Profunmo, "Mozart-MM: a mixed-mode and multi-level simulation system," Proc. IEEE ISCAS, Singapore, Jun. 1991, pp. 2387-2390

[9] M. Schindler, "Demands on simulators escalate as circuit complexity explodes," *Electronic Design*, October, 1987

[10] T. Corman, M. U. Wimbrow, "Coupling a digital logic simulator and an analog circuit simulator," VLSI Systems Design, vol.9, no.2, pp.38-47, 1988

[11] Peter Odryna, "A unified mixed-mode digital/analog simulation environment," Proc. IEEE ISCAS, Finland, Jun. 1988, pp. 893-896

[12] D. Overhauser, I. N. Hajj and Y. Fan, "Automatic mixed-mode timing simulation," Proc IEEE ICCAD, Santa Clara, pp.84-87, Nov. 1989

[13] E. Acuna, J. Derrenis, A. Pagones and R. Saleh, "iSPLICE3: a new simulator for mixed analog/digital circuits," Proc. IEEE Custom Integrated Circuits Conf., San Diego, May 1989, pp.339-342

[14] H. E. Tahawy, A. Chianale and B. Wennion, "Functional verification of analog blocks in FIDELDO: a unified mixed-mode simulation environment," Proc. IEEE ISCAS, Portland, Oregon, May 1989, pp.2012-2015

[15] Y. H. Jun and S. B. Park, "KMIX: a mixed-mode simulator for analog/digital circuits using event driven waveform relaxiation method," Proc. IEEE ISCAS, Portland, Oregon, May 1989, pp. 877-879

[16] Y. H. Jun and I. N. Hajj, "A mixed-mode simulator for digital/analog VLSI circuits using an efficient timing simulation approach," Proc. IEEE ISCAS, Singapore, Jun. 1991, pp.2383-2386

[17] R. Beale, R. Chadha, C. F. Chen, A. Prosser, K. M. Tham, "Design methodology and simulation tools for mixed analog-digital integrated circuits," Proc. IEEE ISCAS, New Orleans, May 1990, pp.1351-1354

[18] P. Odryna, K. Nazareth and C. Christensen, "A workstation-based mixed mode circuit simulator," Proc. IEEE 23rd Design Automation Conf., Las Vegas, 1986, pp.186-192

[19] T. Tormey, "Mixed-mode simulator eases system integration," Computer Design, vol. 28, no. 9, 1989, pp.103-106

[20] S. K. Sullivan, "A mature mixed-mode simulator," Proc. IEEE ISCAS, Portland, May 1989, pp.689-692

[21] R. Chadha, C. Visweswariah. C. F. Chen, "M^3 - a multilevel mixed-mode mixed A/D simulator," IEEE Trans. on Computer-Aided Design, vol. 11, no.5, May 1992, pp.575-585

[22] J. Dabrowski, "Design of multilevel mixed-mode simulator for LSI/VLSI circuits," Proc. IEEE ISCAS, Finland, Jun. 1988, pp.1635-1638

[23] H. E. Tahawy, G. Mazare, B. Hennion, P. Senn, "A new implementation technique for the simulation of mixed (digital-analog) VLSI circuits," Proc. IEEE ICCAD, Santa Clara, 1987, pp.396-399

[24] Cadence Design System Inc., *Verilog-XL Reference Manual 2.0*, 1985

[25] AT&T Bell Laboratories, "UNIX system V release 2.0 Programmer Reference Manual," Dec. 1983

CHAPTER 7

CONCLUSIONS

CONCLUSIONS AND FUTURE WORK

REFERENCES

CONCLUSIONS AND FUTURE WORK

The objective of this work was to address the problem of developing advanced techniques for analysing general nonideal switched networks in the time and frequency domains efficiently. Today, the IC circuit designs are getting larger and more complex whilst designers are under ever-mounting pressure to shorten the design cycle. Hence efficient, versatile and reliable computer aided analysis tools are essential. For example, possible requirements for an analysis tool involved in a filter synthesis system would be: (a) the speed of analysis should be able to keep designers informed in a continuous manner and give out frequency response and time domain waveform of the circuit promptly; (b) designers are able to compare and choose the best circuit realisation in terms of capacitance spread, transconductance ratio or transistor aspect ratio, individual component and overall system sensitivities, hence efficient sensitivity analysis is indispensable; (c) nonideal effects are required to be taken into account; (d) the calculation of signal to noise ratio would require noise analysis capability; (e) if the circuit nonlinearities are of great concern to designers, it would be desirable to see the effects of the dominant nonlinearities without costing too much computer time; (f) for those mixed-mode circuit designers, the simulation tools should be able to provide them a sensible FFT spectrum in a few minutes rather than traditional method which can cost a few days of CPU time on a super computer. All this necessitates very efficient, advanced techniques and elegant computer implementations to meet various analysis requirements. In many cases, current techniques and related software packages for nonideal switched networks were not capable of providing satisfactory solutions. Contributions towards this objective have formed the main achievements of this thesis.

Methods for sensitivity analysis have been reviewed. The perturbation method is simple and straightforward and is a useful alternative to estimate the network sensitivities when an explicit network function, in terms of network parameters, is not available. However, this method is neither efficient nor accurate. A sensitivity network approach was then examined and this reveals that the accuracy can be assured but at the expense of efficiency. The well-known adjoint network approach provides a perfect solution to the above drawbacks. It was seen that only one extra system solution is required irrespective of the number of parameters. Unfortunately it is not directly applicable to general nonideal switched linear networks. A solution is given in Chapter 2, where a method for generating derivatives is proposed. Computer implementation shows that the corresponding algorithms can be conveniently

combined with the original frequency analysis procedure by sharing the same numerical approximation for the inverse Laplace transform. Most nonidealities of interest, such as GB product of the opamp, input and output impedances of the opamp, switch resistances and nodal parasitics can be exactly investigated by using sensitivity analysis. In addition, other applications for instance, group delay and sum sensitivity evaluation are found particularly useful in the problems of equaliser design and network comparison.

For signal processing, the noise behaviour of the filter is usually of great concern to designers. However, existing software packages are only suitable for small circuits due to the excessive computational costs. For instance, full noise analysis of a small SC filter taking a day and half on a modest computer was typical. The difficulties in the noise analysis are primarily due to the folding of wideband white noise from high frequency bands down to the baseband, inherently caused by the sampling nature of sampled-data networks. Therefore, without methods which are theoretically valid, generally applicable and high efficient, noise analysis for switched linear networks is either very expensive or the results tend to be unreliable. The method developed in Chapter 3 is based on the adjoint network technique which had been successfully applied to nonideal sensitivity analysis of switched linear networks. Both the thermal noise and flicker noise generated by MOS transistors are considered. It was demonstrated that only the adjoint system solution is required to obtain the noise behaviour and the original system solution is no longer necessary, hence very substantial savings can then be achieved. The spectrum analysis technique is utilised to take the folding effects into account. In addition, the extension of Hessenberg technique to the adjoint system, extensive sparse matrix routines and interpretable code generation are employed to improve the efficiency of noise analysis. A number of examples show that the speed of noise analysis on a relatively modest workstation is fairy fast, which indicates that noise analysis of large switched systems is feasible.

Further research into the noise analysis method would therefore be valuable. It seems likely that the effectiveness of the noise analysis is highly dependent upon the noise models employed. It would be worthwhile investigating the effects of each individual noise source in the wide range of building blocks being used and therefore build up accurate noise macro-models for them [1]. For instance, the use of noise macro-models for switched current integrators rather than for each individual transistors would simplify the noise analysis of the switched current filters significantly. When noise behaviour of the circuits becomes the major limiting factor to the circuit performance, a library of good noise models would then be necessary to assist in the

prediction of a noise performance which relates closer to the real behaviour. As noise is a random process, the noise analysis results are not usually expected as accurate as the frequency response. Therefore, some compromise between the efficiency and accuracy would always make other approximation techniques possible. The approximation employed at present is the polynomial expression of the excitation. Other forms of approximation could possibly lead to an alternative method with better performance, though perhaps at the expense of the simplicity and ease of implementation of the polynomial method. In particular, a trigonometric approximation would appear to be attractive since the frequency domain analysis always uses sinusoidal excitations. In this case, the approximation would be exact and some important implications might be discovered.

Symbolic analysis of linear analogue circuits is still very popular today. Generally, the problem inherent to the symbolic analysis is the escalation of computer time and memory requirements with increase in circuit size. Some improvements by means of expression approximation and hierarchical decomposition have been reported. However, despite all the effort, these methods still experience great difficulty with medium to large networks. The only practical scheme for large networks of any type is a semi-symbolic one, when polynomials in s or z or both, with purely numeric coefficients, are generated. In this case, the polynomial interpolation method is an ideal candidate for semi-symbolic analysis. However, it has only been applied to continuous-time and ideal SC networks. The application of semi-symbolic analysis to large nonideal switched linear networks is considered in Chapter 4. It was stated that other symbolic analysis methods are not appropriate for nonideal switched linear networks, a polynomial interpolation method was then adopted. Methods for improving the accuracy of the interpolation techniques are discussed. A number of examples show that even for large networks, accuracy has been satisfactory retained. The concept of theoretical computation cost was introduced and based on which the comparison between the symbolic method and numerical approach can be made. Consequently, insight into the efficiency performance of symbolic analysis method results. The upper bound efficiency of the symbolic method against numerical one can be estimated and has been proved in the application to noise analysis.

It would seem possible to extend the semi-symbolic analysis approach to find the poles and zeros of the system. In this respect, the use of QZ algorithm[2] to determine the eigenvalue of the system might be the best approach due to its numerically stable property. The algorithm needs to transform the system to upper Hessenberg form and this does not require extra processing since it happens to be a part of the pre-

processing for the Hessenberg solution technique. Therefore the OZ algorithm is extremely suitable to this application. It is also worthwhile to continue the investigation on the application of symbolic analysis to noise calculation. Although the direct noise application is not very successful, further improvement on this aspect is possible. In Chapter 4, it has been shown that the coefficients of those transfer functions in certain time slots could be very small. In addition, the noise contribution from switch-off resistance is so small and usually can be neglected. By this strategy, the total number of transfer functions required to be evaluated could be reduced dramatically. Therefore the efficiency of symbolic noise analysis might be greatly improved and could prove to be even faster than direct numerical methods.

For high quality filter design, second order effects sometimes could be limiting factors to the circuit performance and should be considered seriously. Techniques have been developed to permit linear nonideal imperfections to be studied quite effectively. However the nonlinearities in analogue sampled-data networks can only be investigated by using general purpose circuit simulators which usually have sophisticated transistor models. Unfortunately, the inherent sampling nature of switched networks will cause these simulators to compute a large number of sharp waveform transitions, and together with detailed device model evaluations, this becomes extremely time consuming. On the contrary, special purposed switched network simulators are often very efficient in the linear domain, is it possible to extend the linear techniques to compute some of the dominant nonlinear effects while maintaining the similar efficiency properties? Chapter 5 tried to answer this question. By modelling nonlinearities of circuit elements with piecewise linear equations, the original nonlinear network can be replaced by a piecewise linear network. A modified Katzenelson algorithm is then proposed which is fully compatible with the existing linear techniques. Due to the nonlinear characteristics of the element, the network status cannot be solely determined by the network topology and adds to the complexity of the problem. A new definition of network state is given and based on which the identification of the network status can be done easily. The simulation results successfully demonstrate that the dominant effects of nonlinearities in switched networks can be studied without recourse to major computing resources.

Further generalisation of the piecewise linear description of non-linear characteristics to allow more complicated nonlinear situations is possible, though perhaps at the cost of increasing simulation time and possible uncertainties regarding the accuracy of solution. However, elements with simple nonlinearities such as diode, or nonlinear capacitor can be incorporated. The efficiency of the present algorithm might be

further improved by taking the continuous property at the boundary of two regions into account [3], which could therefore simplify the procedure. Some further theoretical development is required.

A very important class of the mixed analogue and digital circuits are composed of analogue sampled-data circuits (SC or SI) and digital networks. Recently proposed oversampling techniques offer the possibility of integrating such mixed-mode circuits on one chip. A consequent challenge is the simulation of oversampled sigma-delta data converters which require thousands of clock cycles to obtain meaningful measures of signal to noise ratio (SNR) or signal to distortion ratio (SDR). Most existing mixed-mode simulators employ SPICE like methods to handle analogue circuits. However they are not particularly suitable in this case. Detailed SNR curves are almost impossible to obtain by using these mixed-mode simulators. Therefore special purposed simulators based on difference equations have been developed which did reduce the simulation dramatically, but have great difficulty in taking account of nonideal effects. In Chapter 6, the use of a glued simulator approach for mixed-mode simulation is presented. The highly efficient techniques which have been so effectively used in time domain analysis of switched linear networks were incorporated. Switches are allowed to be controlled by either external periodic clock signals or internally generated non-periodic digital logic. By introducing a new definition of network state, the periodic algorithm can be generalised and applied to non-periodic situation. The nonideal effects can be investigated exactly. A number of programming techniques and computer algorithms have been proposed to exploit maximum simulation efficiency.

Recent experience with MOS delta-sigma A/D converters has revealed that the performance is constrained by analogue circuit imperfections, and not by the oversampling process. Typically, the opamp settling time and limited opamp output swing, finite opamp gain and switch resistance are the most important second order effects. Most of the important nonideal effects in sigma-delta A/D data converters can be investigated by present linear techniques. However, nonlinear effects such as the slew-rate and the opamp output swing limiting cannot be studied at present. It would be possible to combine the present linear techniques with the approaches described in Chapter 5 to extend the simulation facility. The major problem is then to re-express the network state in a suitable manner so that the non-linear characteristics can be taken into account. A certain degree of loss in efficiency might be expected.

To summarise, the result of the present research has contributed towards several powerful simulation tools for analogue filter design and mixed-mode A/D converter verification. Only two years ago, it was not contemplated that simulators could be so powerful and so efficient in being able to handle all kinds of simulation tasks for any types of circuits during the filter design. From basic time and frequency domain analysis to advanced sensitivity, noise and spectrum analysis; from prototype circuit to SC or SI circuit realisation; no matter whether the signal is continuous in time or sampled and held; no matter the network is periodic or non-periodic; linear or nonlinear; a pure filter circuit or an oversampled sigma-delta A/D converter, everything can be done.

REFERENCES

[1] C. Toumazou, J. B. Hughes & N. C. Battersby, *SWITCHED CURRENTS an analogue technique for digital technology*, Peter Peregrinus Ltd. on behalf of IEE, 1993

[2] C. B. Moler and G. W. Stewart, "An algorithm for generalized matrix eigenvalue problems, " SIAM J. Numer. Anal. vol. 10, Apr. 1973, pp.241-256

[3] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*, Van Nostrand Reinhold Co., New York, 1983

APPENDICES

APPENDIX A: SCNAP4 USER'S GUIDE

SCNAP4 User's Guide

Version 1.6

Z Q Shang and J I Sewell

(Aug 1994)

Department of Electronics and Electrical Engineering

University of Glasgow

Glasgow G12 8LT

1. Introduction

This is a tutorial introduction and description of the input language for the SCNAP4 program. SCNAP4 is designed to be of high performance for analysing general linear networks, especially ideal and non-ideal switched capacitor and switched current circuits in both time and frequency domain. It also provides full non-ideal sensitivity analysis and noise analysis with a unified data structure and modularised program structure. The pre-processing of large amount of frequency independent material reduces computation costs and interpretable code generation, polynomial approximation of excitation, descretization of the whole system and Hessenburg techniques are fully exploited. It is expected to be the fastest switched network analysis software in the world. The program is written in C.

2. Element description

The input format of SCNAP4 is very similar to that of SPICE and is of the free format type. The first line of the input file must be a title line, and the last line must be a **.end** card. The order of the remaining lines is arbitrary. SCNAP4 is not case-sensitive.

2.1 Resistor

General form:

rxxxxxx node1 node2 val

Examples:

r1 1 2 100

rin 7 3 4.7k

node1 and **node2** are the two nodes to which the element is connected. **val** is the resistance value and may be positive or negative but not zero. If the resistance is greater than 1E30, the conductance will be set to zero.

2.2 Capacitor

General form:

cxxxxxx node1 node2 val

Examples:

c1 1 2 100pf

cb 7 3 4.7nf

node1 and **node2** are the two nodes to which the element is connected. **val** is the capacitance value and could be negative or zero.

2.3 Inductor

General form:

lxxxxxx node1 node2 val

Examples:

```
l1 1 2 100mH
la 2 3 4.55e-3
```

node1 and **node2** are the two nodes to which the element is connected. **val** is the inductance value and could be negative or zero.

2.4 Switch

General form:

sxxxxxx node1 node2 clknam [ron = val [roff = val]]

Examples:

```
s1 4 7 even
sa 2 3 clk1 ron = 4.7k
s7 8 5 odd ron = 5.7k roff = 4Meg
```

node1 and **node2** are the two nodes to which the element is connected. **clknam** is the name of the clock waveform which controls the switch and must be defined elsewhere. **ron/roff** are optional switch on and off resistances respectively. If the values are not specified then default values are assumed. The default values can be specified by using the option card. However, they cannot override the values which have been set in element card.

2.5 Linear Dependent Sources

SCNAP4 allows circuits to contain linear dependent sources. They are characterised by any of the four following equations

$I = G*V \quad V = E*V \quad I = F*I \quad V = H*I$

where G, E, F and H are constants representing transconductance, voltage gain, current gain and transresistance, respectively.

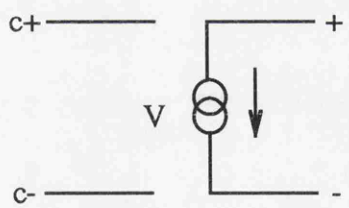
2.5.1 Linear Voltage-Controlled Current Source

General form:

gxxxxxx cnode+ cnode- node+ node- val

Examples:

g1 2 0 5 0 0.1mmho



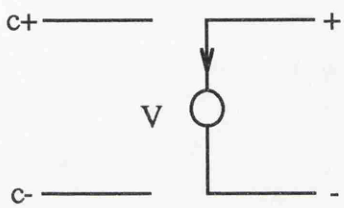
node+ and **node-** are positive and negative source nodes, respectively. Current flow is through the source from the positive node to the negative node. **cnode+** and **cnode-** are the positive and negative controlling nodes, respectively. **val** is the transconductance value.

2.5.2 Linear Voltage-Controlled Voltage Source

General form:

exxxxxx cnode+ cnode- node+ node- val

Examples:



e1 2 3 5 1 100k

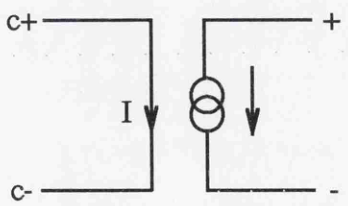
node+ and **node-** are positive and negative source nodes, respectively. **cnode+** and **cnode-** are the positive and negative controlling nodes, respectively. **val** is the voltage gain.

2.5.3 Linear Current-Controlled Current Source

General form:

fxxxxxx node+ node- vname val

Examples:



f1 4 7 vz1 5.0

node+ and **node-** are positive and negative source nodes, respectively. Current flow is through the source from the positive node to the negative node. **vname** is the name of controlling source through which the controlling current flows. The controlling source can be a voltage source, or a voltage-controlled voltage source, or a current-controlled

voltage source or an inductor. The direction of positive controlling current is from the positive controlling node, through the source, to the negative controlling node. **val** is the current gain.

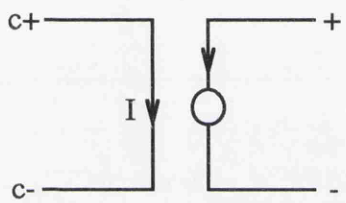
2.5.4 Linear Current-Controlled Voltage Source

General form:

hxxxxxx node+ node- vname val

Examples:

h1 5 17 vz1 0.5k



node+ and **node-** are positive and negative source nodes, respectively. **vname** is the name of controlling source through which the controlling current flows. The controlling source can be a voltage source, or a voltage-controlled voltage source, or a current-controlled voltage source or an inductor. The direction of positive controlling current is from the positive controlling node, through the source, to the negative controlling node. **val** is the transresistance value.

2.6 Independent Source

General form:

vxxxxxx node+ node- [source function]/[ac [acmag [acphase]]]

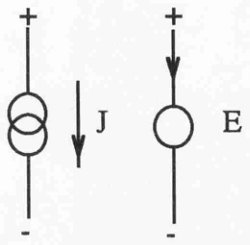
ixxxxxx node+ node- [source function]/[ac [acmag [acphase]]]

Examples:

vin 1 0 dc 5.0v

vs 7 3 ac 1.0 0.0

isrc 4 9 sin 0.0 100mA 5kHz



node+ and **node-** are positive and negative source nodes, respectively. For a voltage source, positive current is assumed to flow from the positive node, through the source and out from the negative node. A current source of positive value will force current to flow from the node+, through the source and from the node-.

For frequency domain analysis, only one AC independent source is allowed. The AC magnitude and phase of the source must be specified. For time-domain analysis, no restrictions are put on the number of independent sources. The source function must be specified for each source. There are five source functions available in SCNAP4:

(1) Step function

General form: **step val**

Examples:

vin 1 0 step 0.1v

isrc 3 7 step 10mA

(2) Impulse function

General form: **pulse val**

Examples:

vs 1 0 pulse 3v

is 3 7 pulse 120mA

(3) DC source

General form: **dc val**

Examples:

vin 1 0 dc 5.0v

isrc 3 7 dc 3mA

(4) Sinusoidal

General form: **sin vo va freq**

Examples:

vs 1 0 sin 0.1v 1.0v 10kHz

isrc 3 7 sin 0 20mA 98kHz

The function is defined as **vo + va*sin(two*pi*freq*time)**

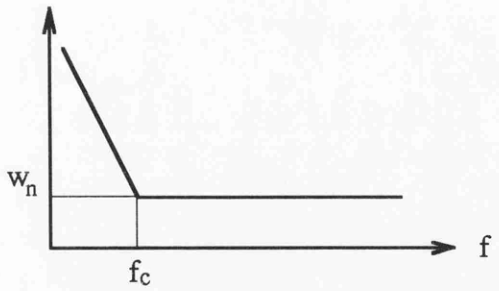
(5) Noise source

General form: **noise fc wn**

Examples:

vn 1 0 noise 1khz 20nv

in 3 7 noise 0 1.004nA



The first statement defines a voltage noise source with both flicker and white noise; The second statement describes a current noise source with white noise only.

fc is the corner frequency where the magnitude of 1/f noise is equal to white noise. **wn** is the magnitude of the white noise. The unit depends on the type of noise sources. They can be either $V/\sqrt{Hz}$ or $A/\sqrt{Hz}$. In addition, the noise sources associated with switches and resistors are added by the program automatically, it is not necessary to define them explicitly.

3. The clock description

To describe the clock signals that control the switches of the network, the **.phase** card is used.

General form:

.phase clknam pwl time0 v0 [[[time1 v1] time2 v2] ...]

Examples:

.phase clk1 pwl 0us 0v 5us 1v 10us 0v

.phase clk2 pwl 0us 1v 10us 0v 20us 1v

The above clock definitions are illustrated in Figure 1.

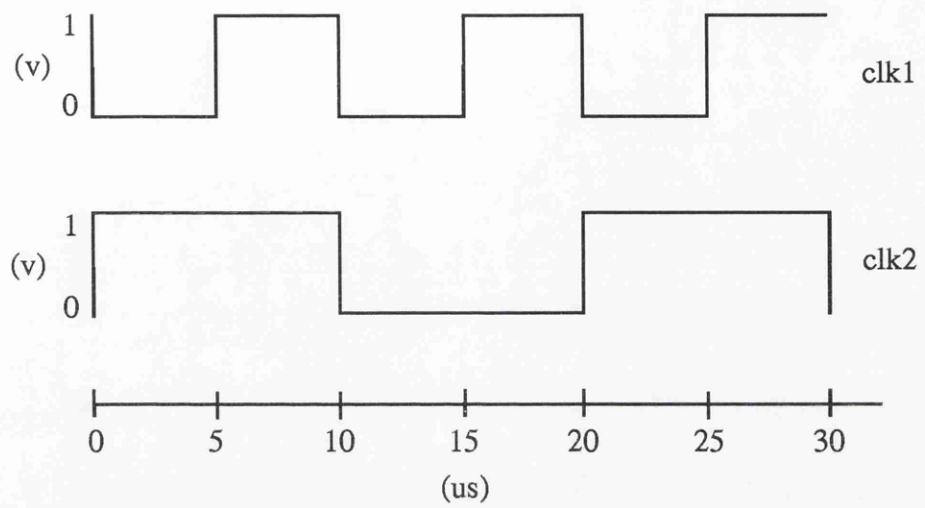


Figure 1 Clock waveform definition

The **clknam** is the clock name that controls switches. **pwl** is a key word which means that the clock waveform is described in a piecewise linear manner. It is defined by pairs of number (time and value) for each of the breakpoints in a waveform. Each waveform need only be described over one complete period, irrespective of the periods of other clocks of the whole system. The value associated with each time is the value of the waveform at the instant immediately after that time.

4. Subcircuit

SCNAP4 provides a powerful subcircuit facility. A subcircuit can be formed with any elements supported by SCNAP4.

4.1 Subcircuit definition

General form:

.subckt subnam node1 [node2 node3 ...] [(parm1 [parm2 parm3 ...])]

Examples:

.subckt opamp 1 2 3 4 (rin, rout, gain)

Subcircuit definition starts from **.subckt** card. **subnam** is the name of the subcircuit; **node1, node2**... are subcircuit's external nodes; **parm1, parm2**... are parameters of the subcircuit and separated by space or comma. Subcircuits having the same circuit structure but different parameters only need one subcircuit definition. After **.subckt** card, there are a series of element cards which describe the subcircuit structure. Subcircuit should use **.ends** card to finish its definition. (See following) No control cards and options cards are allowed within a subcircuit definition. However nested subcircuit definition is possible. Any nodes in subcircuit definition except those in **.subckt** card are considered as local. Ground node is always global.

4.2 Subcircuit end card

General form:

.ends subnam

Example:

.ends opamp

This card should be the last line of any subcircuit definition. **subnam** denotes the subcircuit name.

4.3 Subcircuit call

General form:

xyyyyyy node1 [node2 node3 ...] subnam [(parm1 [parm2 parm3 ...])]

Examples:

x1 3 0 2 4 opamp (800k 100, 1e5)

xa 1 2 3 block1

SCNAP4 treats subcircuit as a pseudo element. **node1, node2**... are nodes which connect to the subcircuit. **subnam** represents the subcircuit name, followed by a series of parameter values. The node and parameter sequence should be the same as defined in corresponding **.subckt** card. If the user wants to reference an element in a subcircuit (eg. sensitivity to that element), the element name should be written as follows; from left to right, first the element name, followed an underline, then the name of subcircuit call. For example, **c1** is in a subcircuit named **opamp**, the subcircuit call is **x1**. To reference **c1**, one should write it as **c1_x1**. If there are nested subcircuits, then write all subcircuit calls from inside to outside with underline in between them. eg. **e1_xa_xb_xc** .

5. Analyze control description

SCNAP4 supports time-domain analysis, frequency-domain analysis (amplitude, phase and group delay), full non-ideal sensitivity analysis (amplitude, phase and group delay)

and noise analysis for both switched capacitor circuits, switched current circuits and continuous time circuits. When there are no switch cards or clock definition cards in a circuit description file, the circuit is automatically treated as general linear network and different algorithms will then be adopted. For current version of SCNAP4, analysis in time and frequency domain must be performed separately. For later versions of SCNAP4, this restriction will be removed.

5.1 Frequency-domain analysis

General form:

.freq fstart fstop lin/log npoints

Examples:

.freq 10 10khz lin 200

.freq 100 32kHz log 10

fstart is the start frequency and **fstop** is the stop frequency. **lin** specifies a linear sweep and **npoints** then specifies the total number of frequency points. **log** defines a logarithmic sweep and **npoints** the specifies the number of frequency points per decade.

5.2 Time-domain analysis

General form:

.tran tstart tstop tstep

Examples:

.tran 0 5ms 0.03ms

.tran 0 3200us 16us

tstart is the time at which the analysis is to start and must be greater than or equal to zero. **tstop** is the time at which the analysis is to stop and must be greater than **tstart**. **tstep** is the printing or plotting increment. **tstep** should be greater than zero.

5.3 Sensitivity analysis [2] (amplitude , phase and group delay)

General form:

.sens [name = [name_list]]

Examples:

.sens name = ca

.sens name = c1 , r1, s4

name is a key word and followed **name_list** is the list of element names. If an element name is specified, the sensitivity of the frequency response of the output node to this element is evaluated. The units are **dB/%** for amplitude sensitivity, **degrees/%** for phase sensitivity and **sec/%** for group delay sensitivity. It is also necessary to specify the option **gdsens** when sensitivity of the group delay with respect to any element is required. If the sensitivity to node parasitics is required, a pseudo capacitor with zero value is

inserted between that node and ground. Total capacitance sensitivity and total transconductance sensitivity as well as the group delay sensitivity to total capacitance or total transconductance are described in a later section.

5.4. Spectrum Analysis

For spectrum analysis, the sweep frequency range defined in **.freq** card always means the output frequency. The input frequency range can only be defined by using the option **band**. An input frequency band n is defined as the frequency interval from $(n-1)f_s/2$ to $(n)f_s/2$ where f_s is the sample frequency.

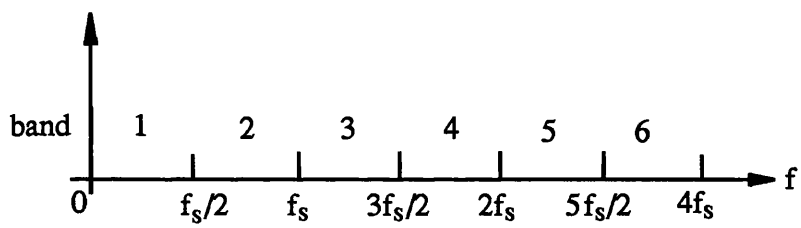


Fig.2 Band definition

Examples

(1) .option band=30

.freq 1Hz 20kHz lin 100

The above specifications will calculate the folding effects from high band 30 to the specific low band in the range of 1Hz-20kHz.

(2) .option band=1

.freq 121kHz 139Khz lin 100

This allows the user to study the imaging effects from baseband to high band in the range of 121-139kHz.

It should be pointed out that the frequency range defined in **.freq** card must belong to one band and should not include the band boundaries.

5.4. Noise Analysis [3]

General form:

.noise fstart fstop lin/log npoints stop_band=val [temp=val]

Examples:

.noise 10 20khz lin 200 stop_band=30

.noise 1 20MHz log 20 stop_band=1 temp=300

fstart is the start frequency and **fstop** is the stop frequency. **lin** specifies a linear sweep and **npoints** then specifies the total number of frequency points. **log** defines a logarithmic sweep and **npoints** the specifies the number of frequency points per decade. It should be stated that **fstart** and **fstop** have to be in the same band for the sake of folding effects evaluation. **stop_band** is the number of bands to be considered. This allows users to take folding effects into account. Because the noise is normally bandlimited by op-amp frequency response and the time constant effects from switch resistances and circuit capacitance, the **stop_band** is set equal to that noise bandwidth. However, for continuous time circuits, **stop_band** should not be specified. **temp** is the temperature in Kelvin

which is used for white noise evaluation. Note, unlike sensitivity analysis where **.freq** card and **.sens** card should be used together, only the **.noise** card is needed for noise analysis.

6. Option facilities

General form:

.options opt1 opt2... (or opt=optval...)

Examples:

.options cont ron = 10 ohm

The above card means that continuous I/O coupling effects are taking into account in frequency domain analysis and all switch-on resistances are set to 10 ohm. SCNAP4 allows user to select following options in an arbitrary order.

options	function
cont	treat input as a continuous time signal. (default S/H signal) This option will ignore any sampling cards.
band = x	set the frequency range over which the generator is swept. This option is used for spectrum analysis.
order = x	specify approximation order of input source for transient analysis or AC analysis if **cont** has been specified.[1] For frequency analysis, the order cannot be zero (default is 5)
roff = x	set switch-off resistances which have not been specified in switch cards. (default 1E10 ohm)
ron = x	set switch-on resistances which have not been specified in switch cards. (default 1 ohm)
acct	print out the run time statistics.
nosinc	the effect of sin(x)/x is not considered. (default sin(x)/x effect is computed)
tgsens	calculate total transconductance sensitivity. Sensitivity analysis must be specified. (useful for SI circuits and transconductor-C circuits)
tcsens	calculate total capacitance sensitivity. Sensitivity analysis must be specified. (useful for SC circuits)
groupd	calculate group delay. Sensitivity analysis must be specified.
gdsens	calculate group delay sensitivity to those elements specified in **.sens** card, this option implies the specification of the **groupd** option.
tgdsens	The group delay sensitivity to total transconductance
tcdsens	The group delay sensitivity to total capacitance

7. Sample and hold

General form:

.sample input clknam

.sample output clknam

Examples:

.sample output clk2

.sample input clk1

SCNAP4 allows users to study sample and holds effect for both input signal and output response. **input** and **output** are key words and referring to signal type. **clknam** is the clock name which describes sample and hold time intervals. These time intervals correspond to certain basic time slots. For input signals, it means that input is only supplied in defined clock phases. Similarly, for output sample and hold effect, the outputs are only observed in defined clock phases.

8. Plot card

General form:

.plot pltype ov1 < ov2...>

Examples,

.plot tran v(2) v(3) i(vin#branch)

.plot ac vdb(3,4) vr(4) vi(4) vm(4)

.plot noise vdb(up3)

After each successful run, an output file named rawfile will be created automatically. It is in the same format as produced by SPICE3. So it directly fits into SPICE post-processor NUTMEG. **pltype** denotes the required analysis type. (transient, AC, noise) For transient analysis, the output variable types are node voltage and branch current. To reference a

branch current, the branch name should be formed correctly. For most of the elements, it is the combination of an element name, which produces the branch current, and "#branch" . For current-controlled voltage source, "#contbranch" is needed to reference its controlling branch. For Frequency domain analysis, the output can be any of the following types, **vr** for real part, **vi** for imaginary part, **vm** for the magnitude, **vp** for the phase in degree and **vdb** for magnitude in dB. Actually the rawfile contains both real part and imaginary part data. For example, if vi(6) has been specified, then the real part data is referring to vr(6) and imaginary part is for vi(6). Same rule applies on **vm** and **vp, vdb** and **vp**. In the case of group delay calculation, the real part is the amplitude sensitivity and the imaginary part is the group delay. For noise analysis, there are two types of output noise power spectral density, vdb(5) is in $dB/\sqrt{Hz}$ and v(5) is in $V/\sqrt{Hz}$ (or i(vn#branch) in $A/\sqrt{Hz}$)

9. Examples

Several circuits are used to show various facilities of SCNAP4.

A) Second order band pass SC filter

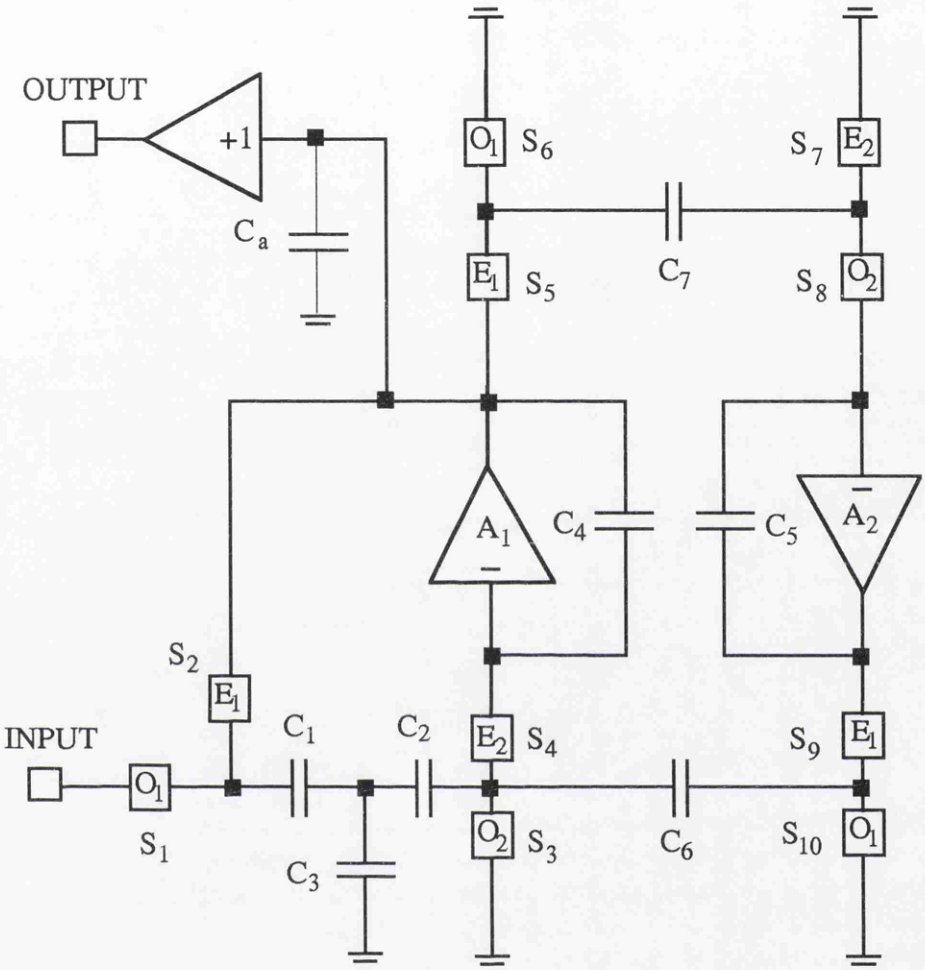


Figure 3. Second order bandpass SC filter

The following listed circuit description file is for a frequency domain analysis and a sensitivity analysis. All switch-on resistances are set to 7.15k.

```
sc band-pass fc=400kHz
* this is a comment line
s1 1 2 clk3
s2 2 6 clk1
s3 4 0 clk4
s4 4 5 clk2
s5 6 7 clk1
s6 7 0 clk3
s7 8 0 clk2
s8 8 9 clk4
s9 10 11 clk1
s10 11 0 clk3

c1 2 3 0.1pf
c2 3 4 0.1pf
c3 3 0 0.70206pf
c4 5 6 0.70206pf
c5 9 10 0.70206pf
c6 4 11 0.17668pf
c7 7 8 0.17668pf

.subckt opamp 1 2 6 (gain,unit)
* gain = 10k, gb = 30M, bw = 3k
 ra 1 0 500Meg
 rb 2 0 500Meg
 rin 1 2 1Meg
 e1 1 2 3 0 gain
 e2 4 0 5 0 unit
* GB=30Meg
 r1 3 4 .00005305164769729844
 c1 4 0 1.0
 rout 5 6 100
.ends opamp

x1 0 9 10 opamp (10k,1)
x2 0 5 6  opamp (10k,1)

vin 1 0 ac 1 0

.phase clk1 pwl 0ns 0v 10ns 1v 47.5ns 0v 100ns 0v
.phase clk2 pwl 0ns 1v 37.5ns 0v 100ns 0v
.phase clk3 pwl 0ns 0v 60ns 1v 97.5ns 0v 100ns 0v
.phase clk4 pwl 0ns 0v 50ns 1v 87.5ns 0v 100ns 0v
.freq 360kHz 440kHz lin 55
.sens name= e1_x1, s4, c1
.plot ac vdb(6)
.option ron = 7.15k
.end
```

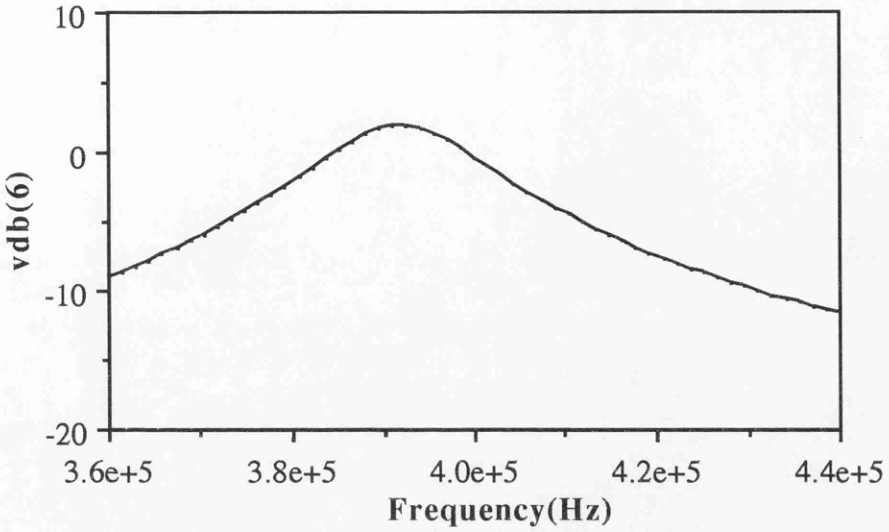


Figure 4a. Circuit response of 2nd order bandpass SCF

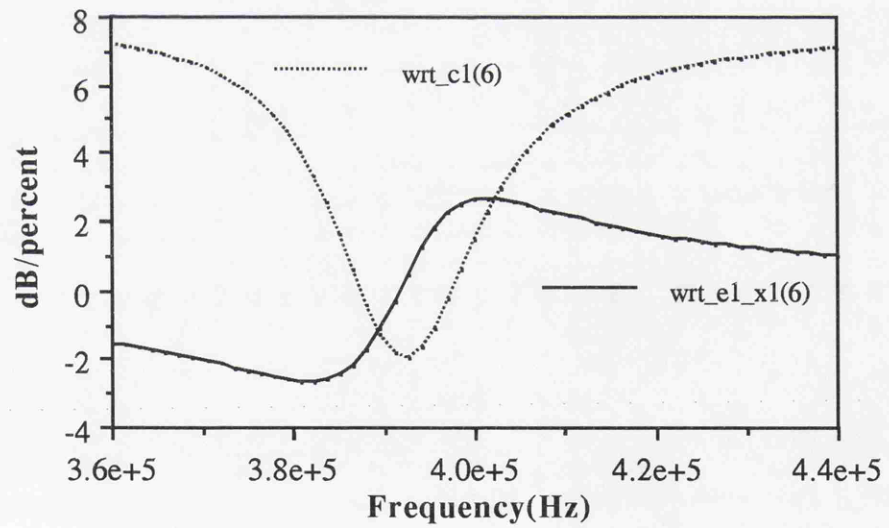


Figure 4b. Sensitivities w.r.t. C1 and E1 in subcircuit X1

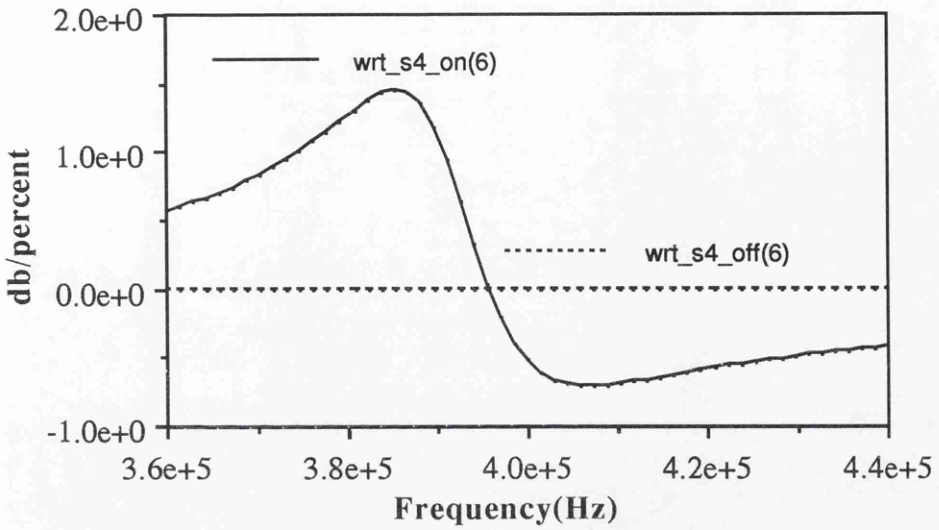


Figure 4c. Sensitivity w.r.t. switch on/off resistances of S4

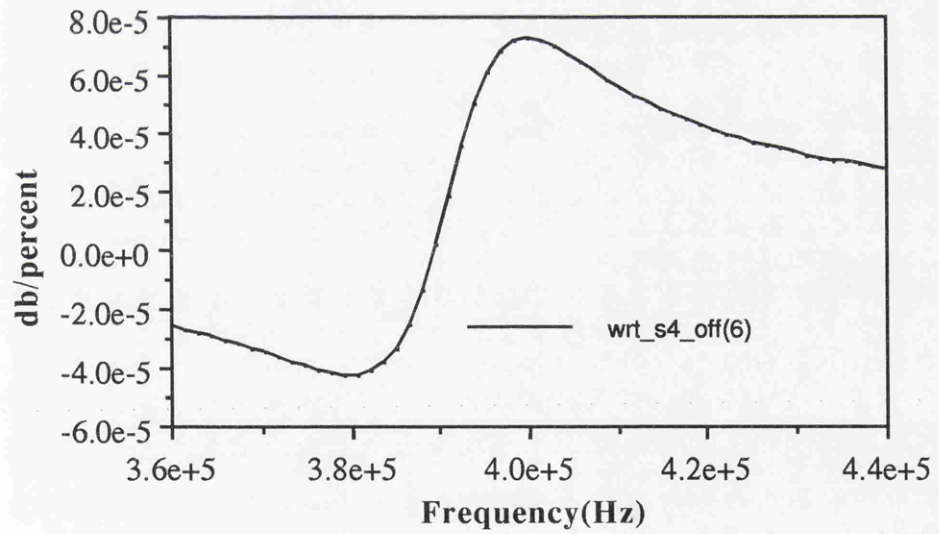


Figure 4d. Detail of sensitivity for switch off resistance of S4

For this circuit, sensitivity to switch off resistance has less effect than switch on resistance and can be ignored.

To perform transient analysis, the following statements should be added,

```
vin 1 0 step 1
.tran 0u 3u 0.1u
.plot tran v(6) i(vin#branch)
```

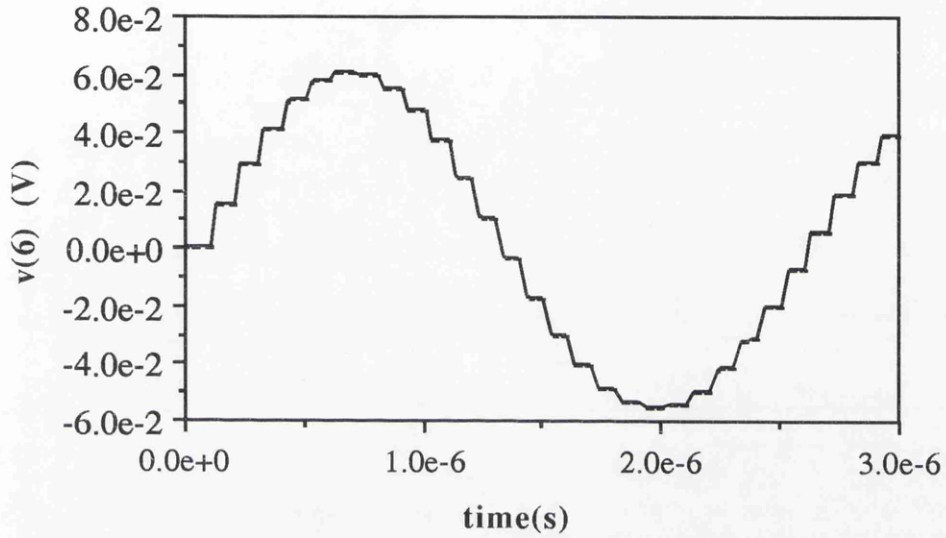


Figure 4e. Circuit response in time domain

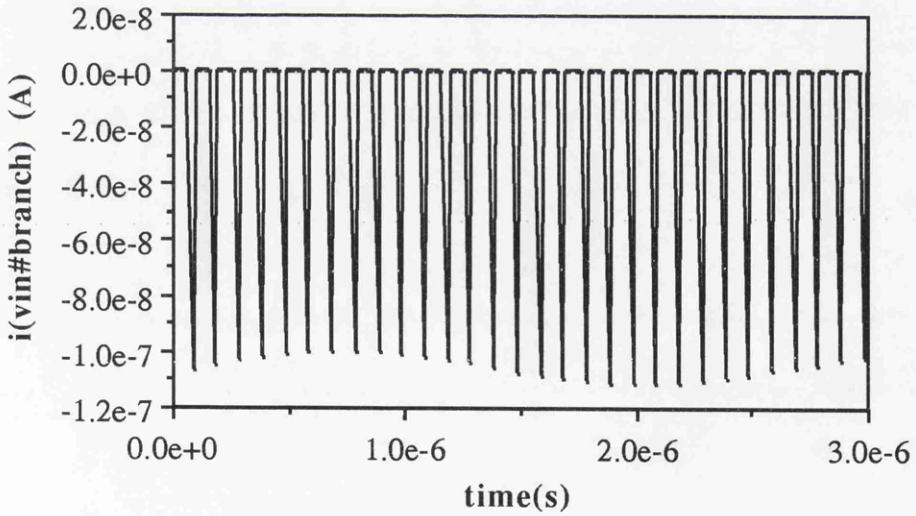


Figure 4f. Current through the voltage source

B) 9th order active RC low pass filter

SCNAP4 can also analyse general linear networks. Here is an example circuit of 9th order elliptic lowpass filter. Analysis will be performed both in frequency and time domain. Circuit description file is listed as follows,

```
active rc

r1 26 1 5.4779k
r6 1 2 4.440k
r11 2 3 3.2201k
r16 3 4 3.63678k
r21 4 5 1.2201k

.subckt blk 1 (r1 r2 c1)
 ra 1 2 r1
 ca 2 3 c1
 rb 3 4 3.3k
 rc 4 5 3.3k
 rd 5 6 r2
 cb 6 0 10nf
 e1 6 4 3 0 1e5
 e2 2 4 5 0 1e5
.ends blk

x1 1 blk (2.0076k 4.5898k 12nf )
x2 2 blk (5.9999k 4.25725k 6.8nf )
x3 3 blk (5.88327k 5.62599k 4.7nf )
x4 4 blk (1.0301k 5.808498k 6.8nf )
c9 5 0 10nf

vin 26 0 sin 0 1 2k

.tran 0 1ms 10u
.plot tran v(5) i(e1_x4#branch)
*vin 26 0 ac 1 0
*.freq 0.5k 50k log 50
*.plot ac vdb(5)
.end
```

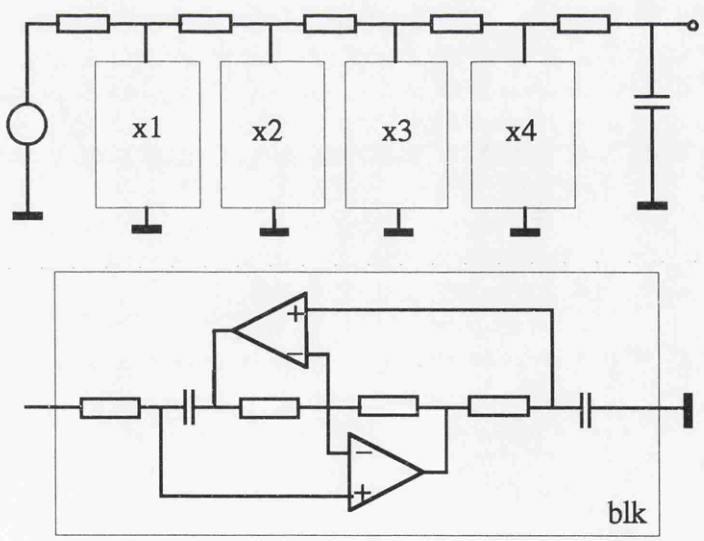


Figure 5. Circuit diagram of 9th-order elliptic low-pass filter

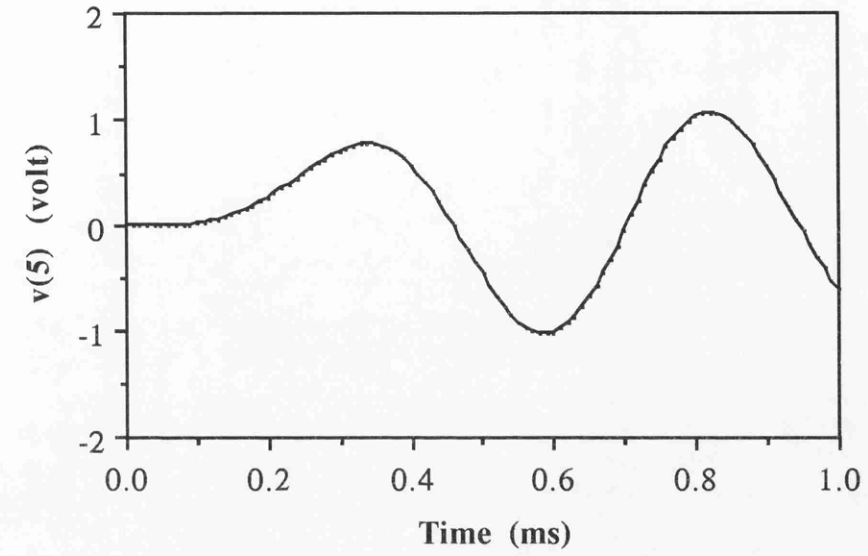


Figure 5a. Circuit response in time domain

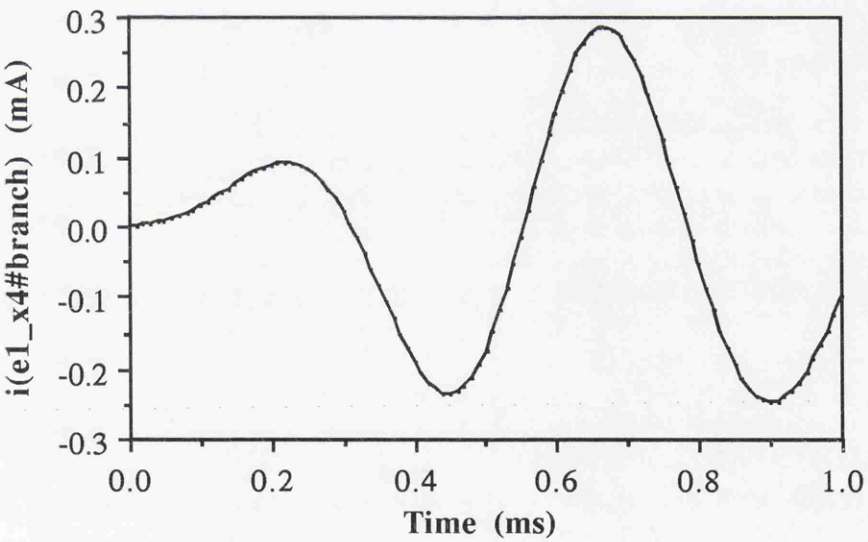


Figure 5b. Branch current of E1 in subcircuit X4

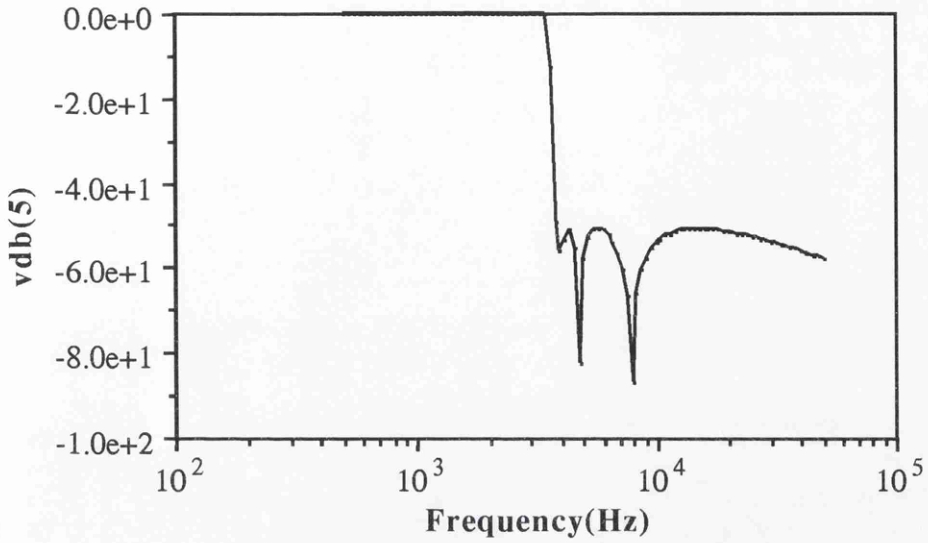


Figure 5c. Circuit response in frequency domain

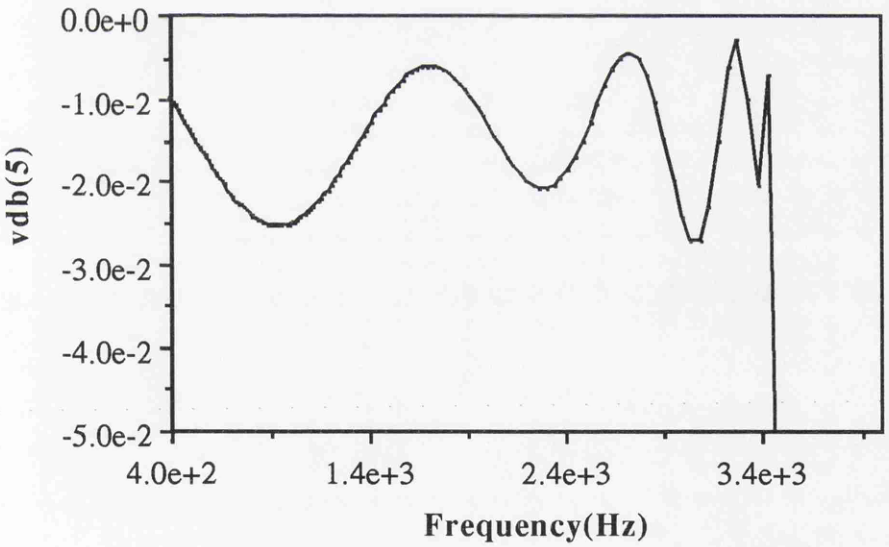


Figure 5d. Pass band response of the filter

C) An tenth order LC filter

Its input file is listed below,

```
10th order LC filter

IIN 1 0 AC 1.0 0.0
R1 1 0 1.000000E+00
L2 1 0 1.182735E-05
C3 1 2 3.665065E-04
L4 1 2 3.251082E-05
L5 2 0 1.081065E-06
C6 2 3 5.108854E-03
L7 2 3 1.448280E-06
C8 3 0 3.263391E-02
C9 3 4 2.383789E-02
L10 3 4 4.976204E-08
C11 4 0 4.684072E-01
C12 4 5 1.700702E-01
L13 4 5 4.331158E-09
L14 5 0 3.398811E-10
C15 5 0 8.591970E+00
R16 5 0 3.955826E-05

.FREQ 1.000000E-01 8.000000E+03 LIN 100
.PLOT AC VDB(5)
.END
```

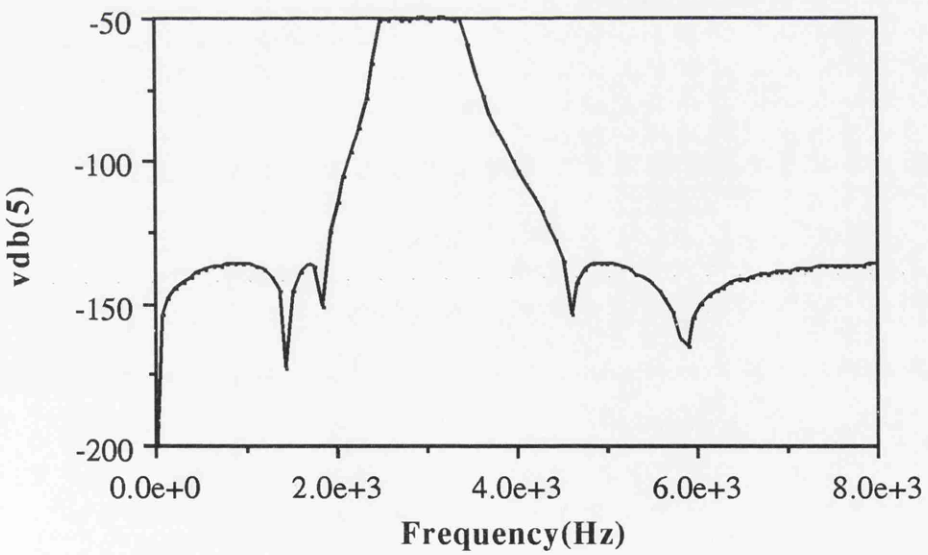


Figure 6. Frequency response of the 10th order LC filter

D) 5th order low-pass transconductor-C filter

This is a transconductor-C filter with differential output. Circuit response in frequency domain, group delay as well as the group delay sensitivity with respect to C1 are evaluated. The filter netlist produced by XFILT[4] is listed as follows,

```
5th order filter
* Number of Transconductors = 5
* Number of Low Impedance Transconductors = 3
* Number of Capacitors =18
* Total Capacitance = 1.753995E+02 units
* Capacitance Spread = 2.354031E+01
* Average Capacitor = 2.700047E-07 units
* Transconductance Spread = 1.000000E+00

.SUBCKT IDEGM 1 2 44 33 3 4
  G1 1 2 4 3  1.00000000E+00
  VZ1 33 44 DC 0
  F1 4 3 VZ1 1
*  H1 5 6 VZ1 1
*  G2 5 6 4 3 1
  R1 1 0 1E9
  R2 2 0 1E9
  R3 3 0 1E9
  R4 4 0 1E9
*  R5 5 0 1E9
*  R6 6 0 1E9
  R33 33 0 1E9
  R44 44 0 1E9
.ENDS IDEGM

.SUBCKT GM 1 2 3 4
  G1 3 4 2 1  1.0E+00
  R1 1 0 1.0E+9
  R2 2 0 1.0E+9
  R3 3 0 1.0E+9
  R4 4 0 1.0E+9
.ENDS GM

VIN 1 12 AC 1.0 0.0
C1 3 0 4.39829085E-07
C2 3 9 2.77086549E-08
C3 4 8 2.77086549E-08
C4 4 0 6.52270205E-07
C5 4 10 7.53280503E-08
C6 5 9 7.53280503E-08
C7 5 0 3.98705776E-07
C8 6 0 3.89693543E-07
C9 7 0 3.43470193E-07
X10 17 6 19 8  3 14 IDEGM
X11 6 17 20 9 4 15 IDEGM
X12 4 15 18 7 GM
X13 7 18 21 10 5 16 IDEGM
X14 6 17 3 14 GM
X15 6 17 15 4 GM
```

```
X16 7 18 4 15 GM
X17 7 18 16 5 GM
G18 3 14 3 14 1.00000000E+00
G19 5 16 5 16 9.99999977E-01
G20 1 12 3 14 1.00000000E+00
C21 14 0 4.39829085E-07
C22 14 20 2.77086549E-08
C23 15 19 2.77086549E-08
C24 15 0 6.52270205E-07
C25 15 21 7.53280503E-08
C26 16 20 7.53280503E-08
C27 16 0 3.98705776E-07
C28 17 0 3.89693543E-07
C29 18 0 3.43470193E-07
R30 1 0 1.00000000E+09
R31 12 0 1.00000000E+09

.OPTION ACCT GDSENS
.FREQ 1.000000E-01 3.00000001E+06 LIN 100
.SENS NAME = C1
.PLOT AC VDB(5,16)
*.PLOT AC VR(5) VR(16)
.END
```

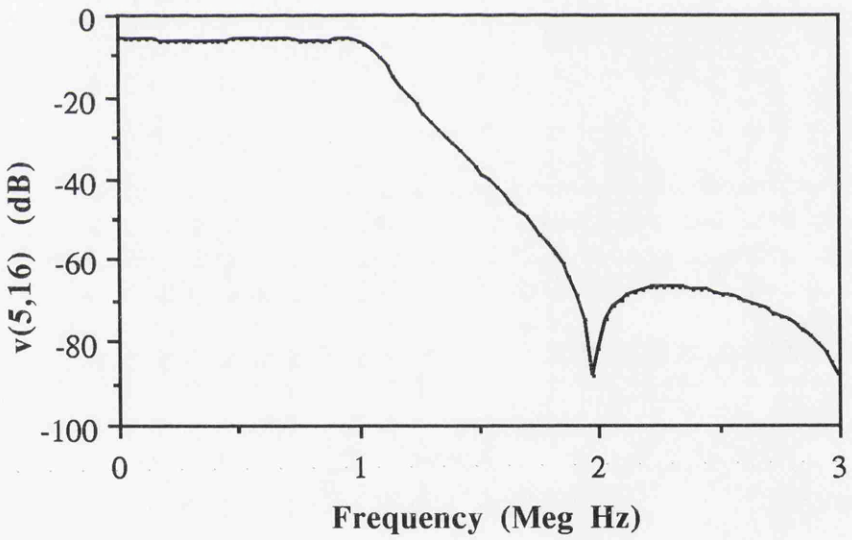


Figure 7a Circuit response in frequency domain

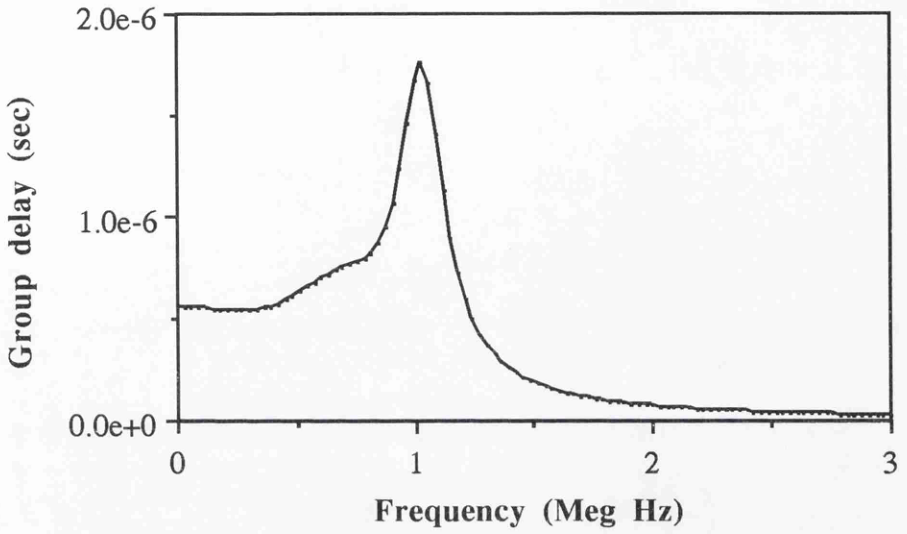


Figure 7b. Group delay response

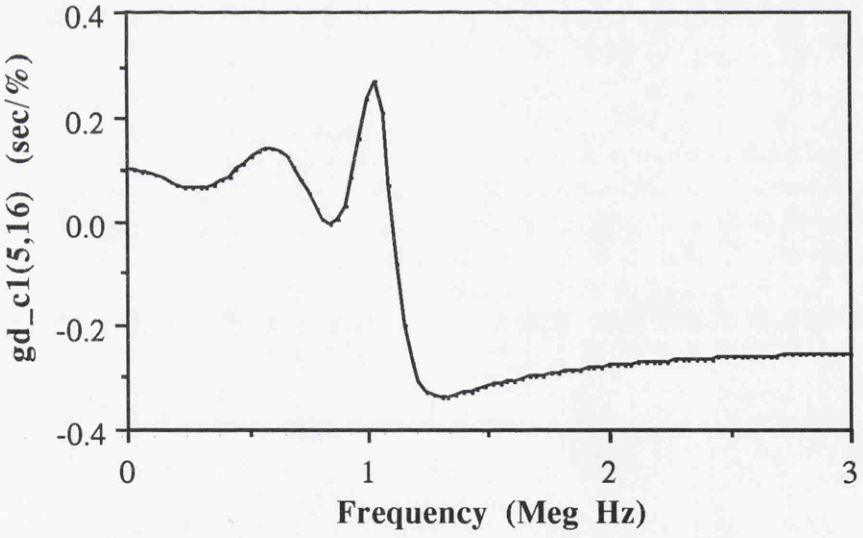


Figure 7c. Group delay sensitivity w.r.t. C1

E) 5th order switched current low-pass filter

SCNAP4 can simulate switched current circuits as well. For this particular example, frequency response as well as the total transconductance sensitivity are computed. The circuit is illustrated in Fig. 8. The filter netlist is also created by XFILT and given below,

```
TITLE: 5TH ORDER SI FILTER

*  Number of integrator = 7
*  Number of output of integrator = 19
*  Number of switches   = 21
*  Maximum transistor ratio = 21.227285 units
.OPTION RON=1.000000e-01 ROFF=1.000000e+20
.subckt si_int 1 2 3 (t1, t2, t3)
  s1 1 2 t1
  s2 2 4 t2
  c3 4 0 1.000000e-15
  c4 3 0 1.000000e-15
  g5 4 0 2 0 1
  r5 2 0 1.000000e+08
  c5 2 0 1.000000e-15
  g6 3 0 2 0 1
  r6 2 0 1.000000e+08
  c6 2 0 1.000000e-15
  s7 2 3 t3
.ends si_int

.subckt MOS 1 2 ( k )
  c1 1 0 1.000000e-15
  g5 1 0 2 0 k
  r5 2 0 1.000000e+08
  c5 2 0 1.000000e-15
.ends MOS

Xsi_f_int4  24 29 4 si_int (T1, T1, T2)
Xsi_f_int5  25 30 5 si_int (T1, T1, T2)
Xsi_f_int24  35 33 31 si_int (T1, T1, T2)
Xsi_b_int1  6 11 1 si_int (T2, T1, T2)
Xsi_b_int2  7 12 2 si_int (T2, T1, T2)
Xsi_b_int3  8 13 3 si_int (T2, T1, T2)
Xsi_b_int25  35 34 32 si_int (T2, T1, T2)
XMOS6  2 11  MOS (1.044944e-01)
XMOS7  1 12  MOS (4.710918e-02)
XMOS8  3 12  MOS (7.922061e-02)
XMOS9  2 13  MOS (3.326804e-01)
XMOS10  4 6  MOS (4.052371e-01)
XMOS11  4 7  MOS (2.221415e-01)
XMOS12  5 7  MOS (2.341582e-01)
XMOS13  5 8  MOS (5.813819e-01)
XMOS14  1 24  MOS (2.002396e-01)
XMOS15  2 24  MOS (2.434770e-01)
XMOS16  2 25  MOS (2.693633e-01)
XMOS17  3 25  MOS (1.592581e-01)
XMOS18  1 6  MOS (1.576095e-01)
XMOS19  3 8  MOS (1.542148e-01)
XMOS20  31 11  MOS (1.133061e-01)
```

```
XMOS21  32 37  MOS (1.133061e-01)
XMOS22  37 37  MOS (1.000000e+00)
XMOS23  37 11  MOS (1.000000e+00)
XMOS27  3 38  MOS (1.000000e+00)
R28 0 38 1.000000e+00
G26 36 0 35 0 1.000000e+00
VIN 36 0 AC 1 0
.PHASE T2 PWL 0.0 0V 1.666667e-08 1V 3.333333e-08 0V
.PHASE T1 PWL 0.0 1V 1.666667e-08 0V 3.333333e-08 1V
.FREQ 1.000000e+00 1.000000e+07 LIN 100

.PLOT AC VDB(38)
.END
```

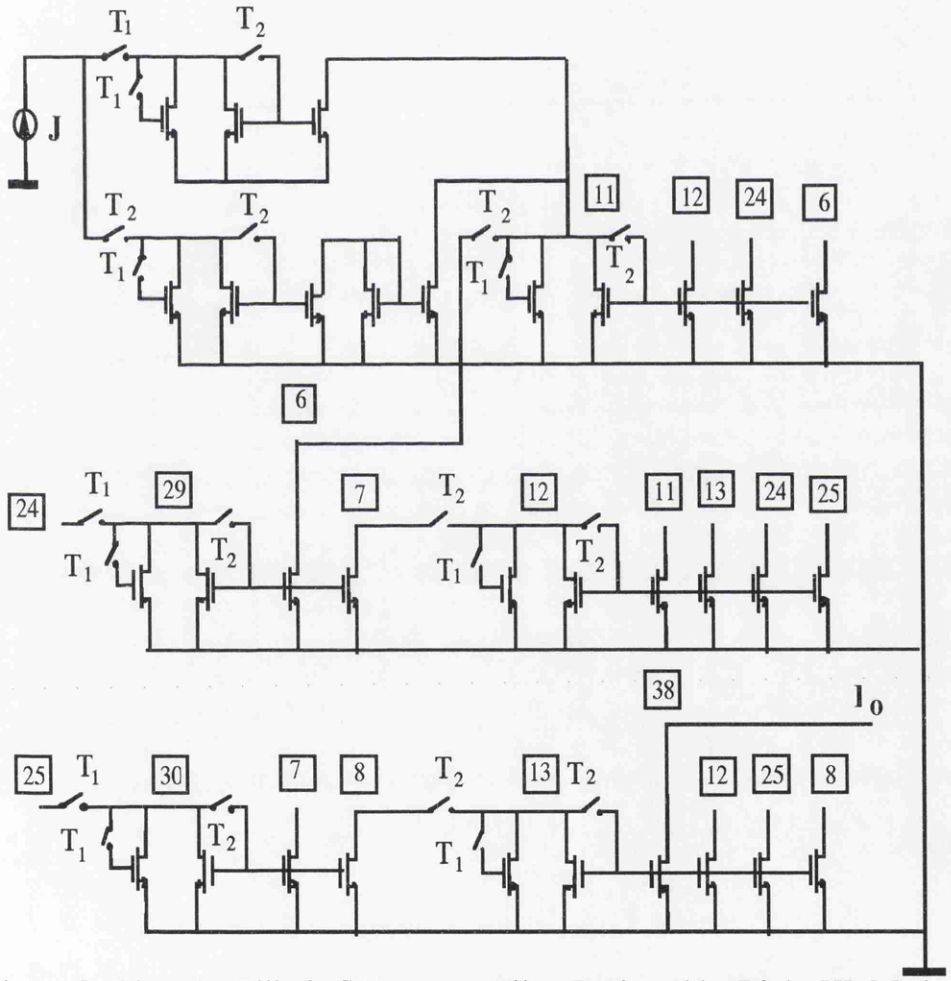


Figure 8. 5th-order Elliptic SI Lowpass Filter Designed by Right-UL Method

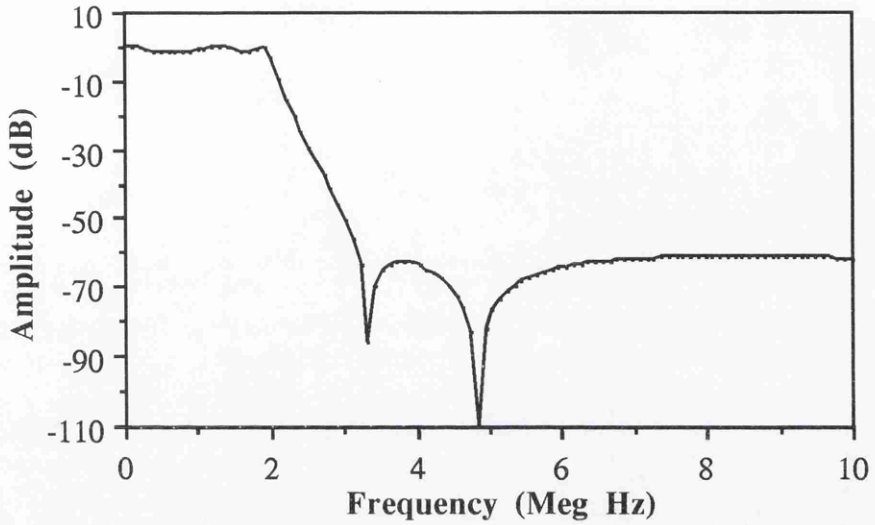


Figure 9a. Frequency response of 5th-order Elliptic Lowpass SI Video Filter

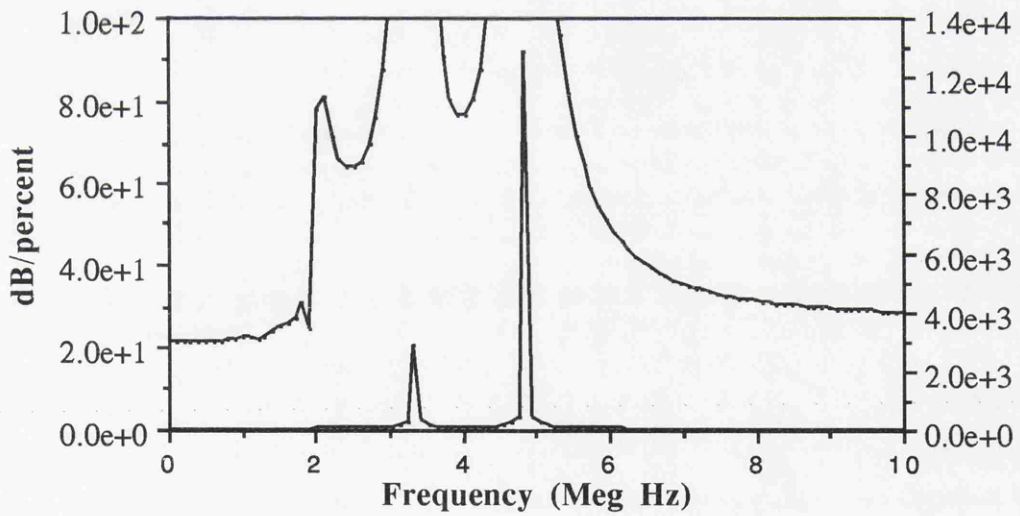


Figure. 9b. Total sensitivity of transconductance on two different scales

F) Noise analysis of a 6th order band-pass SC filter

For this filter, the folding back effects of 10 high bands are calculated. The filter netlist is listed as follows,

```
6th order bpf 2-phase switching

.option ron = 0.001 acct
vin 1 0 ac 1 0
c1  2  5 2.2pf
c2  3  4 2.2pf
c3  6  7 138.782pf
c4 10 11 116.089pf
c5 14 15 116.089pf
c6 18 19 96.253pf
c7 22 23 116.089pf
c8 27 28 138.782pf
c9  5 12 35.007pf
c10  8  9 29.283pf
c11 12 13 2.2pf
c12  9 16 2.2pf
c13 13 20 29.083pf
c14 16 17 24.076pf
c15 16 24 2.2pf
c16 13 21 2.2pf
c17 21 26 35.007pf
c18 24 25 29.283pf
c19 26 25 2.2pf

s1 1 3 e1
s2 3 0 o1
s3 1 2 o1
s4 2 7 e1
s5 4 0 e1
s6 5 0 o1
s7 5 6 e1
s8 4 6 o1
s9 12 0 o1
s10 11 12 e1
s11 13 0  e1
s12 13 14 o1
s13 21 0  e1
s14 21 22 o1
s15 26 0  o1
s16 26 27 e1
s17 7 8 e1
s18 8 0 o1
s19 9 0 e1
s20 9 10 o1
s21 16 0 e1
s22 16 15 o1
s23 17 0 e1
s24 17 18 o1
s25 20 0 o1
s26 19 20 e1
s27 24 0 o1
```

```
s28 24 23 e1
s29 25 0 o1
s30 25 28 e1

.subckt opamp 1 2 3 4 (gain,fb,wn)
 vn a 1 noise fb wn
 e1 a 2 3 4 gain
.ends opamp

xe1 0 27 28 0 opamp (1e3,1k,80n)
xe2 0 23 22 0 opamp (1e3,1k,80n)
xe3 0 18 19 0 opamp (1e3,1k,80n)
xe4 0 14 15 0 opamp (1e3,1k,80n)
xe5 0 10 11 0 opamp (1e3,1k,80n)
xe6 0  6  7 0 opamp (1e3,1k,80n)

.phase e1 pwl 0us 1v 2us 0v 4us 1v
.phase o1 pwl 0us 0v 2us 1v 4us 0v

.noise 1 20k lin 100 stop_band=10
.plot noise vdb(28)
.end
```

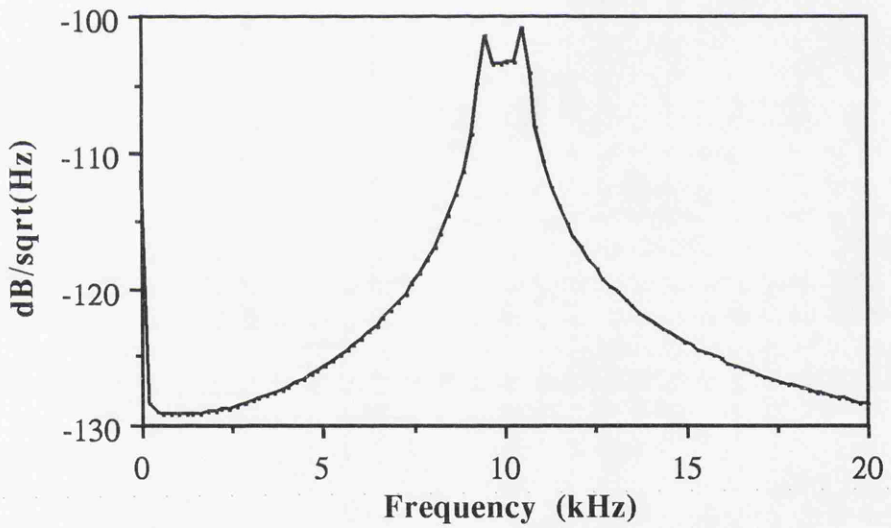


Figure 10. Noise behaviour of 6th order bandpass SC filter

References

[1] L. B.Wolovitz and J. I. Sewell, "General analysis of large linear switched capacitor networks", Proc.IEE , Pt. G, vol. 135, no. 3, June 1988, pp.119-124

[2] Z. Q.Shang and J. I.Sewell, "Efficient sensitivity analysis for large non-ideal switched capacitor networks", Proc. IEEE ISCAS, Chicago, May 1993, pp.1405-1407

[3] Z. Q. Shang and J. I. Sewell, "Efficient noise analysis methods for large non-ideal SC and SI circuits", Proc. IEEE ISCAS, London, June 1994, pp.5.565-5.568

[4] Lu Yue and J. I. Sewell, "XFILT Reference Manual and User Guide", Department of Electronics and Electrical Engineering, University of Glasgow, October, 1992

APPENDIX B: SCNAP5 USER'S GUIDE

SCNAP5 User's Guide

Version 1.0

Z Q Shang and J I Sewell

(March 1995)

Department of Electronics and Electrical Engineering

University of Glasgow

Glasgow G12 8LT

1. Introduction

SCNAP5 is an extension of SCNAP4 [1] with the enhancement to time domain analysis of both periodic and non-periodic sampled-data analogue networks. It is designed to run under the CADENCE socket environment for mixed-mode simulation, hence part of this manual relates to mixed-mode simulation and users are referred to the appropriate CADENCE document [2]. Many highly efficient techniques which were so effectively used in SCNAP4 have been incorporated. Switches are allowed to be controlled by either external periodic clock signals or internally generated non-periodic digital logic. The nonideal effects such as gain and bandwidth product of opamps and switch resistance can be investigated exactly. The program is written in C.

2. How to run SCNAP5

For mixed-mode simulation purposes, SCNAP5 allows a series of command line options which permit to be used as either a master simulator or as a slave simulator.

(a) SCNAP5 as master

General form:

scnap5 -mixmod -slave<slave command line> -slvhost<name> -shellhost<name>
-shellport<name> -mmdebug <file>

Example:

scnap5 -mixmod -slave"verilog.vmx +vmxcconfig.vmx test.v"
-shellhosthogmanay.elec -shellport1312 -mmdebug test.in

option	function
mixmod	Indicates a mixed-mode simulation
slave	Indicates the name of slave simulator and any required command line
slvhost	Indicates the hostname of slave and is only needed if the slave is running on a seperate machine
shellhost	Indicates the name of the machine where ISC † is running
shellport	Indicates the socket port that the ISC has opened
mmdebug	Flag that turns on the ISC side of the IPC ‡ degug

† ISC stands for **I**nteractive **S**imulation **C**ontrol

‡ IPC stands for **I**nter **P**rocess **C**ommunication

(b) SCNAP5 as slave

General form:

scnap5 -mixmod -host<name> -sport<name> -shellhost<name> -shellport<num>
-mmdebug <file>

Example:

scnap5 -mixmod -hosthogmanat.elec -shellhosthogmanay.elec -shellport1432
-mmdebug test.in

option	function
mixmod	As in above table
host	Indicates the name of the machine where the master simulator is running
sport	Indicates the socket port that the master has opened
shellhost	As in above table
shellport	As in above table
mmdebug	As in above table

3. Element description

The input format of SCNAP5 is very similar to that of SPICE and is of the free format type. The first line of the input file must be a title line, and the last line must be a **.end** card. The order of the remaining lines is arbitrary. SCNAP5 is not case-sensitive.

3.1 Resistor

General form:

rxxxxxx node1 node2 val

Examples:

r1 1 2 100
rin 7 3 4.7k

node1 and **node2** are the two nodes to which the element is connected. **val** is the resistance value and may be positive or negative but not zero. If the resistance is greater than 1E30, the conductance will be set to zero.

3.2 Capacitor

General form:

cxxxxxx node1 node2 val

Examples:

c1 1 2 100pf

cb 7 3 4.7nf

node1 and **node2** are the two nodes to which the element is connected. **val** is the capacitance value and can be positive, negative or zero.

3.3 Inductor

General form:

lxxxxxx node1 node2 val

Examples:

l1 1 2 100mH

la 2 3 4.55e-3

node1 and **node2** are the two nodes to which the element is connected. **val** is the inductance value and could be positive, negative or zero.

3.4 Switch

General form:

sxxxxxx node1 node2 clknam [ron = val [roff = val]]

Examples:

s1 4 7 even

sa 2 3 clk1 ron = 4.7k

s7 8 5 odd ron = 5.7k roff = 4Meg

node1 and **node2** are the two nodes to which the element is connected. **clknam** is the name of the clock waveform which controls the switch and must be defined elsewhere. **ron/roff** are optional switch on and off resistances respectively. If the values are not specified then default values are assumed. The default values can be specified by using the option card. However, they cannot override the values which have been set in element card.

3.5 Linear Dependent Sources

SCNAP5 allows circuits to contain linear dependent sources. They are characterised by any of the four following equations

I = G*V V = E*V I = F*I V = H*I

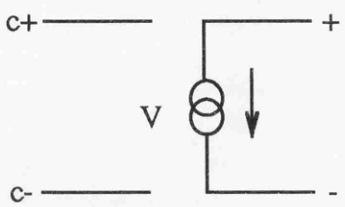
where G, E, F and H are constants representing transconductance, voltage gain, current gain and transresistance, respectively.

3.5.1 Linear Voltage-Controlled Current Source

General form:

gxxxxxx cnode+ cnode- node+ node- val

Examples:



g1 2 0 5 0 0.1mmho

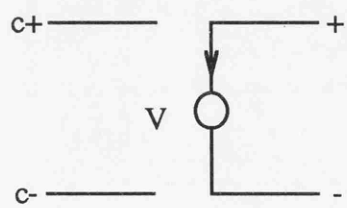
node+ and **node-** are positive and negative source nodes, respectively. Current flow is through the source from the positive node to the negative node. **cnode+** and **cnode-** are the positive and negative controlling nodes, respectively. **val** is the transconductance value.

3.5.2 Linear Voltage-Controlled Voltage Source

General form:

exxxxxx cnode+ cnode- node+ node- val

Examples:



e1 2 3 5 1 100k

node+ and **node-** are positive and negative source nodes, respectively. **cnode+** and **cnode-** are the positive and negative controlling nodes, respectively. **val** is the voltage gain.

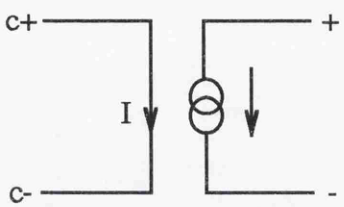
3.5.3 Linear Current-Controlled Current Source

General form:

fxxxxxx node+ node- vname val

Examples:

f1 4 7 vz1 5.0



node+ and **node-** are positive and negative source nodes, respectively. Current flow is through the source from the positive node to the negative node. **vname** is the name of controlling source through which the controlling current flows. The controlling source can be a voltage source, or a voltage-controlled voltage source, or a current-controlled voltage source or an inductor. The direction of positive controlling current is from the positive controlling node, through the source, to the negative controlling node. **val** is the current gain.

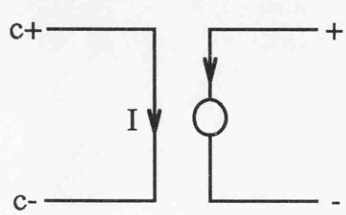
3.5.4 Linear Current-Controlled Voltage Source

General form:

hxxxxxx node+ node- vname val

Examples:

h1 5 17 vz1 0.5k



node+ and **node-** are positive and negative source nodes, respectively. **vname** is the name of controlling source through which the controlling current flows. The controlling source can be a voltage source, or a voltage-controlled voltage source, or a current-controlled voltage source or an inductor. The direction of positive controlling current is from the positive controlling node, through the source, to the negative controlling node. **val** is the transresistance value.

3.6 Independent Sources

General form:

vxxxxxx node+ node- [source function]

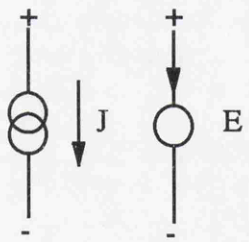
ixxxxxx node+ node- [source function]

Examples:

vin 1 0 dc 5.0v

vs 7 3 step 1.0v

isrc 4 9 sin 0.0 100mA 5kHz



node+ and **node-** are positive and negative source nodes, respectively. For a voltage source, positive current is assumed to flow from the positive node, through the source and out from the negative node. A current source of positive value will force current to flow from the **node+**, through the source and from the **node-**.

3.7 Source function

For time-domain analysis, no restrictions are put on the number of independent sources. A source function must be specified for each source and five of these are available in SCNAP5:

3.7.1 Step function

General form: **step val**

Examples:

vin 1 0 step 0.1v

isrc 3 7 step 10mA

3.7.2 Impulse function

General form: **impulse val**

Examples:

vs 1 0 pulse 3v

is 3 7 pulse 120mA

3.7.3 DC source

General form: **dc val**

Examples:

vin 1 0 dc 5.0v

isrc 3 7 dc 3mA

3.7.4 Sinusoidal

General form: **sin vo va freq**

Examples:

vs 1 0 sin 0.1v 1.0v 10kHz

isrc 3 7 sin 0 20mA 98kHz

The function is defined as **vo + va*sin(two*pi*freq*time)**

3.7.5 Square wave

General form: **pulse v1 v2 td tr tf tw per**

Examples

vin 1 0 pulse -1 1 0 8u 8u 984u 2ms

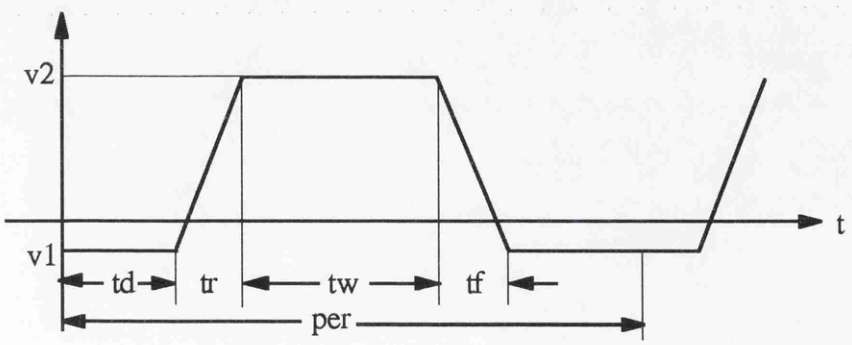


Fig. 1 Square waveform definition

4. Comparator

General form:

kxxxxxx in+ in- clknam [vref = val [voff = val]]

Examples:

k1 10 0 99 clk1

ka 0 net7 net99 phi1

in+ and **in-** are the two input node names of the comparator. **out** is the output node name and should be the same as defined in **.a2d** card. **clknam** is the name of the clock waveform which controls the comparator and must be defined elsewhere. The comparator only changes its output at the rising edge of the controlling clock signal. **vref** is the reference voltage and **voff** is the offset voltage associated with node **in+** (has not been implemented yet!)

5. The clock description

To describe the clock signals that control the switches of the network, the **.phase** card is used.

General form:

For periodic clock signal

.phase clknam pwl time0 v0 [[[time1 v1] time2 v2] ...]

For non-periodic clock signal

.phase clknam d2a t1 v1 t2 v2

Examples:

.phase clk1 pwl 0us 0v 5us 1v 10us 0v

.phase clk2 pwl 0us 1v 10us 0v 20us 1v

.phase phi1 d2a 0us 0v 0.3ms 5v

The **clknam** is the clock name that controls switches. **pwl** is a key word which means that the clock waveform is periodically repeated and described in a piecewise linear manner. It is defined by pairs of numbers (time and value) for each of the breakpoints in a waveform. Each waveform need only be described over one complete period, irrespective of the periods of other clocks of the whole system. The value associated with each time is the value of the waveform at the instant immediately after that time. **d2a** is the key word which means that the clock waveform is determined by digital to logic events. (**t1, v1**) and (**t2, v2**) are pairs of breakpoints in a waveform which only serves as an initial condition and will be overridden during the simulation. It should be stressed that the **clknam** should be defined elsewhere in **.d2a** card.

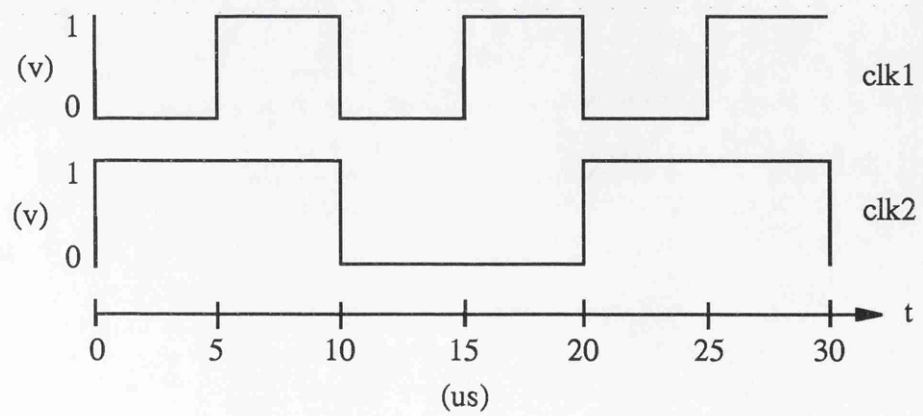


Fig. 2 Clock waveform definition

6. Interface nodes between analogue and digital network

6.1 Analogue to digital node

General form:

.a2d dout ain vl vh tx

Example:

.a2d 99999 net99 0v 5v 0us

This card starts with **.a2d** which defines an analogue to digital node. **dout** is the interface node name which is known by the digital simulator. **ain** is the node name which describes the same node known to SCNAP5. **vl** and **vh** are low and high voltage levels, respectively, the node will be automatically translated to digital "0" or "1" correspondingly. **tx** is the length of time that it takes a node voltage to transit between **vl** and **vh**, in our situation, **tx** is usually set to zero.

6.2 Digital to analogue node

General form:

.d2a din clknam vl vh tr tf

Examples:

.d2a 99998 clk1 0v 5v 0us 0us

.d2a 99997 clk2 0v 5v 0us 0us

This card begins with **.d2a** which indicates a digital to analogue node. **din** is the node name known by digital simulator. **clknam** is the clock name which controls some switches in the analogue circuit. **vl** and **vh** are the low and high voltage levels which describe the clock waveform and should be greater than or equal to zero. **tr** and **tf** are the rise and fall times of the transition. Since SCNAP5 only allows ideal clock waveforms, **tr** and **tf** are normally set to zero.

7. Subcircuit

SCNAP5 provides a powerful subcircuit facility. A subcircuit can be formed with any elements supported by SCNAP5.

7.1 Subcircuit definition

General form:

.subckt subnam node1 [node2 node3 ...] [(parm1 [parm2 parm3 ...])]

Examples:

.subckt opamp 1 2 3 4 (rin, rout, gain)

Subcircuit definition starts from **.subckt** card. **subnam** is the name of the subcircuit; **node1**, **node2**... are subcircuit's external nodes; **parm1**, **parm2**... are parameters of the subcircuit and separated by space or comma. Subcircuits having the same circuit structure

but different parameters only need one subcircuit definition. After **.subckt** card, there are a series of element cards which describe the subcircuit structure. Subcircuit should use **.ends** card to finish its definition. (See following) No control cards and options cards are allowed within a subcircuit definition. However nested subcircuit definition is possible. Any nodes in subcircuit definition except those in **.subckt** card are considered as local. Ground node is always global.

7.2 Subcircuit end card

General form:

.ends subnam

Example:

.ends opamp

This card should be the last line of any subcircuit definition. **subnam** denotes the subcircuit name.

7.3 Subcircuit call

General form:

xyyyyyy node1 [node2 node3 ...] subnam [(parm1 [parm2 parm3 ...])]

Examples:

x1 3 0 2 4 opamp (800k 100, 1e5)

xa 1 2 3 block1

SCNAP5 treats subcircuit as a pseudo element. **node1, node2**... are nodes which connect to the subcircuit. **subnam** represents the subcircuit name, followed by a series of parameter values. The node and parameter sequence should be the same as defined in corresponding **.subckt** card. If the user wants to reference an element in a subcircuit, the element name should be written as follows; from left to right, first the element name, followed by an underline, then the name of subcircuit call. For example, **c1** is in a subcircuit named **opamp** and the subcircuit call is **x1**. To reference **c1**, one should write it as **c1_x1**. If there are nested subcircuits, then write all subcircuit calls from inside to outside with an underline between each. eg. **e1_xa_xb_xc** .

8. Analyse control description

SCNAP5 supports time-domain analysis for both switched capacitor circuits, switched current circuits and continuous time circuits.

General form:

.tran tstart tstop tstep

Examples:

.tran 0 5ms 0.03ms

.tran 0 3200us 16us

tstart is the time at which the analysis is to start and must be greater than or equal to zero. **tstop** is the time at which the analysis is to stop and must be greater than **tstart**. **tstep** is the printing or plotting increment. **tstep** should be greater than zero.

9. Option facilities

General form:

.options opt1 opt2... (or opt=optval...)

Examples:

.options ron = 10 ohm

The above card means that all switch-on resistances are set to 10 ohm. SCNAP5 allows user to select following options in an arbitrary order.

options	function
order = x	specify approximation order of input source for transient analysis (default is 5)
roff = x	set switch-off resistances which have not been specified in switch cards. (default 1E10 ohm)
ron = x	set switch-on resistances which have not been specified in switch cards. (default 1 ohm)
acct	print out the run time statistics.

10. Sample and hold

General form:

.sample input clknam

.sample output clknam

Examples:

.sample output clk2

.sample input clk1

SCNAPDIS allows users to study sample and hold effects for both input signal and output response. **input** and **output** are key words and refer to signal type. **clknam** is the clock name which describes sample and hold time intervals. These time intervals correspond to certain basic time slots. For input signals, it means that input is only supplied in defined clock phases. Similarly, for output sample and hold effect, the outputs are only observed in defined clock phases.

11. Plot card

General form:

.plot pltype ov1 < ov2...>

Example,

.plot tran v(2) v(3) i(vin#branch)

After each successful run, an output file named rawfile will be created automatically. It is in the same format as produced by SPICE3. So it directly fits into SPICE post-processor NUTMEG. **pltype** denotes the required analysis type. (transient in this case) For transient analysis, the output variable types are node voltage and branch current. To reference a branch current, the branch name should be formed correctly. For most of the elements, it is the combination of an element name, which produces the branch current, and "#branch" . For current-controlled voltage source, "#contbranch" is needed to reference its controlling branch.

12. Examples

Several circuits are used to show various facilities of SCNAP5.

A) Fully balanced first-order sigma-delta SC modulator

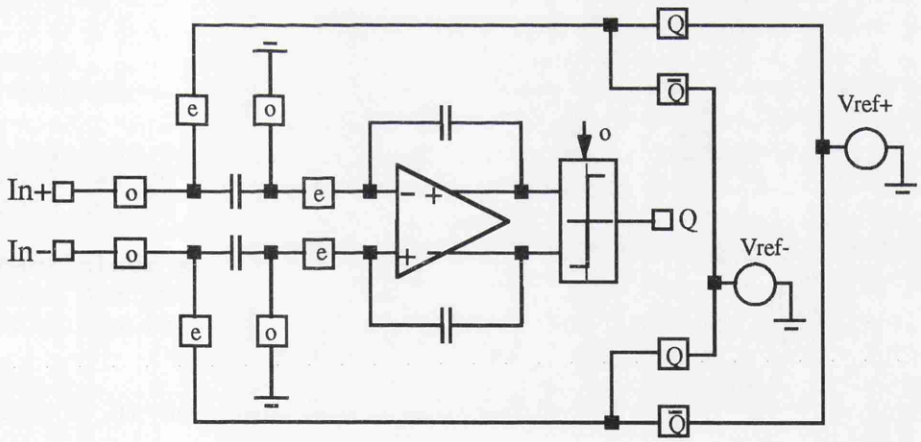


Fig. 2 First-order sigma-delta SC modulator

The following is the SCNAP5 circuit description file,

```
* First-order sigma-delta modulator
* scnap5 netlist
*
* periodically controlled switches
```

```
s1 1 6 phi1
s6 6 11 phi2
s7 7 0 phi1
s8 7 8 phi2

s2 2 3 phi1
s3 3 12 phi2
s4 4 0 phi1
s5 4 5 phi2

* feed back controlled switches
s9 11 100 clk1
s10 11 200 clk2
s11 12 100 clk2
s12 12 200 clk1

* reference voltage
vref- 100 0 dc -1v
vref+ 200 0 dc +1v

* sampled capacitors
c3 6 7 0.5pf
c1 3 4 0.5pf

* integrator capacitors
c4 8 9 1.0pf
c2 5 10 1.0pf

* opamps
e2 5 8 9 0 50k
e1 9 0 10 0 -1

* comparator
k1 9 10 99 phi1

* switching rate = 1.66666667MegHz
.phase phi1 pwl 0u 1v 0.3u 0v 0.6u 1v
.phase phi2 pwl 0u 0v 0.3u 1v 0.6u 0v

* feed back clock signal
.phase clk1 d2a 0u 0v 0.3m 5v
.phase clk2 d2a 0u 0v 0.3m 5v

* a2d node
.a2d 99999 99 0v 5v 0us

* d2a node (never used)
.d2a 99998 clk1 0v 5v 0.0us 0.0us
.d2a 99997 clk2 0v 5v 0.0us 0.0us

* input
v1 1 0 dc -0.5
v2 2 0 dc 0.5

* print out job statistics
.option acct

* analyse control
```

```
.tran 0u 30u 0.3u
.plot tran v(99999) v(9) v(10) v(11) v(12)
.end
```

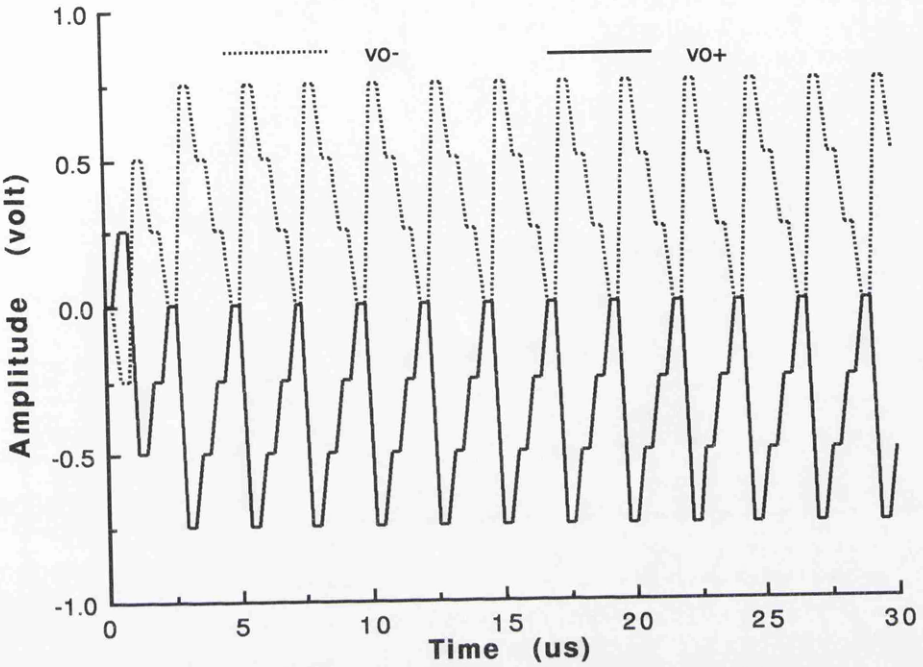


Fig. 3.1 Output waveform of balanced opamp of a 1st-order sigma-delta A/D converter

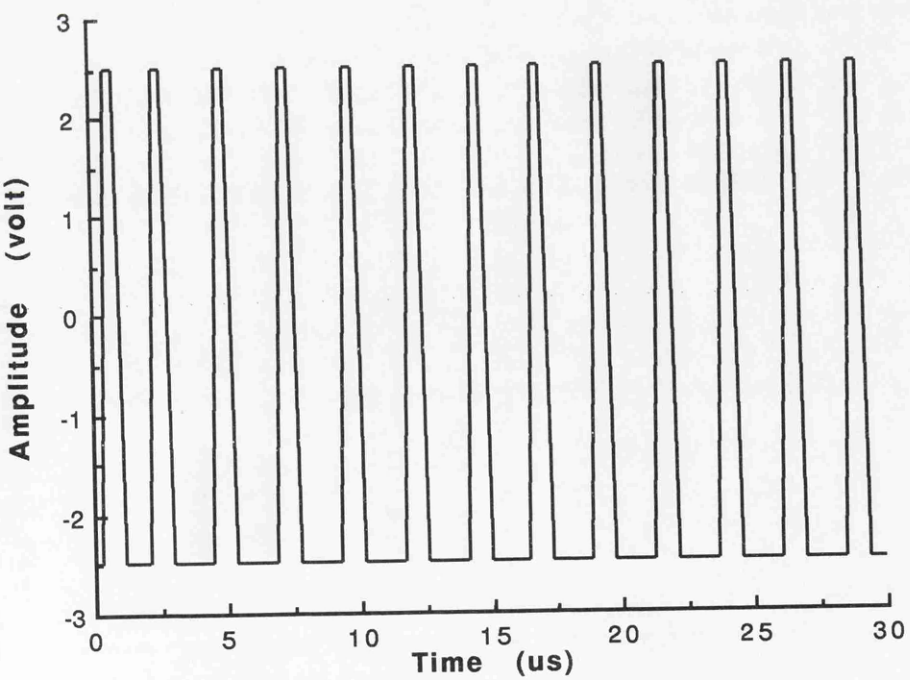


Fig. 3.2 Output waveform of the comparator of a 1st-order sigma-delta A/D converter

B) A 2nd-order sigma-delta SC modulator

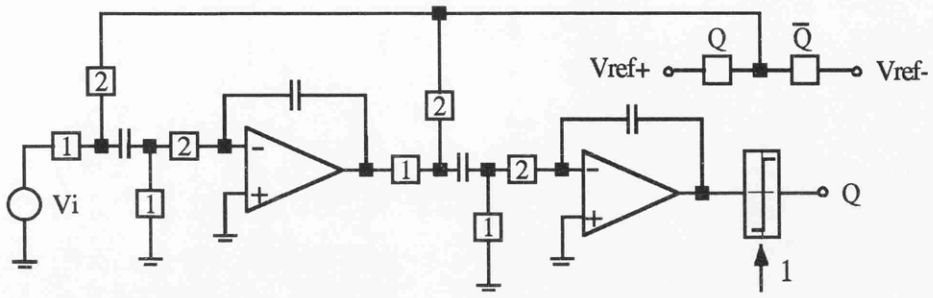


Fig. 4 SC implementation of a second-order sigma-delta modulator

The netlis is given as follows,

```
* second-order sigma-delta modulator
* scnap5 netlist
*
* input frequency = 1kHz
vin 1 0 sin 0 0.5v 1kHz

* periodicaly controlled switches
s1 1 2 phi1
s2 2 11 phi2
s3 3 0 phi1
s4 3 4 phi2
s5 5 6 phi1
s6 6 11 phi2
s7 7 0 phi1
s8 7 8 phi2

* feed back controlled switches
s9 11 100 clk1
s10 11 200 clk2

* sampled capacitors
c1 2 3 0.5pf
c3 6 7 0.5pf

* integrator capacitors
c2 4 5 1.0pf
c4 8 9 1.0pf

e1 0 4 5 0 1e3
e2 0 8 9 0 1e3

* comparator
k1 0 9 99 phi1
```

```
* switching rate = 1.66667MegHz
.phase phi1 pwl 0u 1v 0.3u 0v 0.6u 1v
.phase phi2 pwl 0u 0v 0.3u 1v 0.6u 0v

* feed back clock signal
.phase clk1 d2a 0u 0v 0.3m 5v
.phase clk2 d2a 0u 0v 0.3m 5v

* a2d node
.a2d 99999 99 0v 5v 0us

* d2a node (never used)
.d2a 99998 clk1 0v 5v 0.0us 0.0us
.d2a 99997 clk2 0v 5v 0.0us 0.0us

* reference voltage
vref+ 100 0 dc 1v
vref- 200 0 dc -1v

* print out job statistics
.option acct ron=1

* analyse control
*.tran 0u 4.0e-2 0.3u
.plot tran v(99)
.end
```

N. B. FFT routines are required to produce the following graphs. A considerable amount of computer time is required to produce Fig. 4.1!

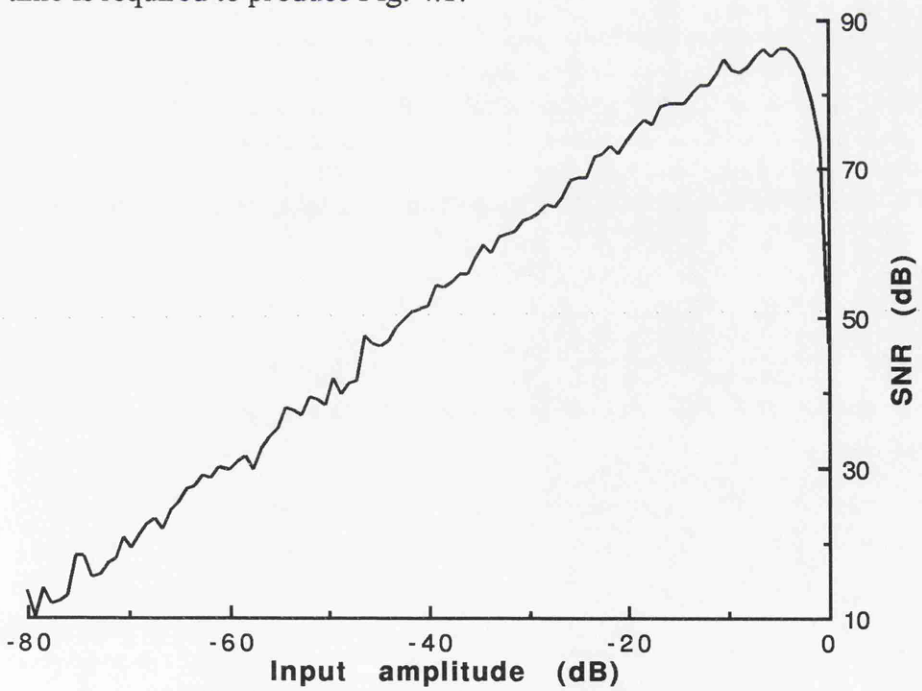


Fig. 4.1 SNR versus relative input amplitude for 2nd-order sigma-delta modulator

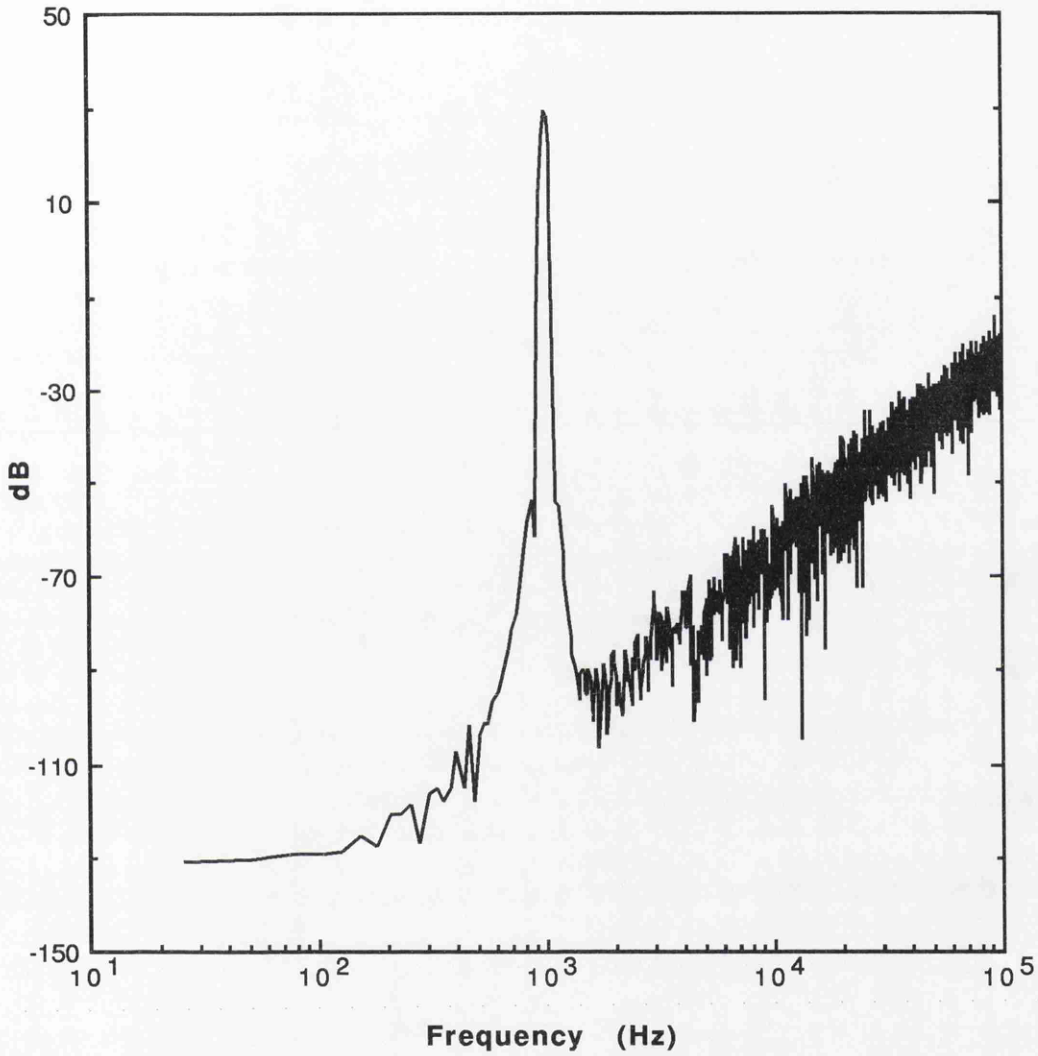


Fig. 4.2 FFT spectrum of the comparator output for 2nd-order sigma-delta modulator

The commands of in house software to produce FFT and SNR curves are as follows,

sim c_fft 64K 1.6666666667M NUTTALL 0 < sdmout

sim c_snr 64K 1k 1.66666666667M 3.90625E-3 NUTTALL 0 < sdmout

C) 5th-order elliptic lowpass SC filter

SCNAP5 can also analyse periodic sampled-data analogue circuits.

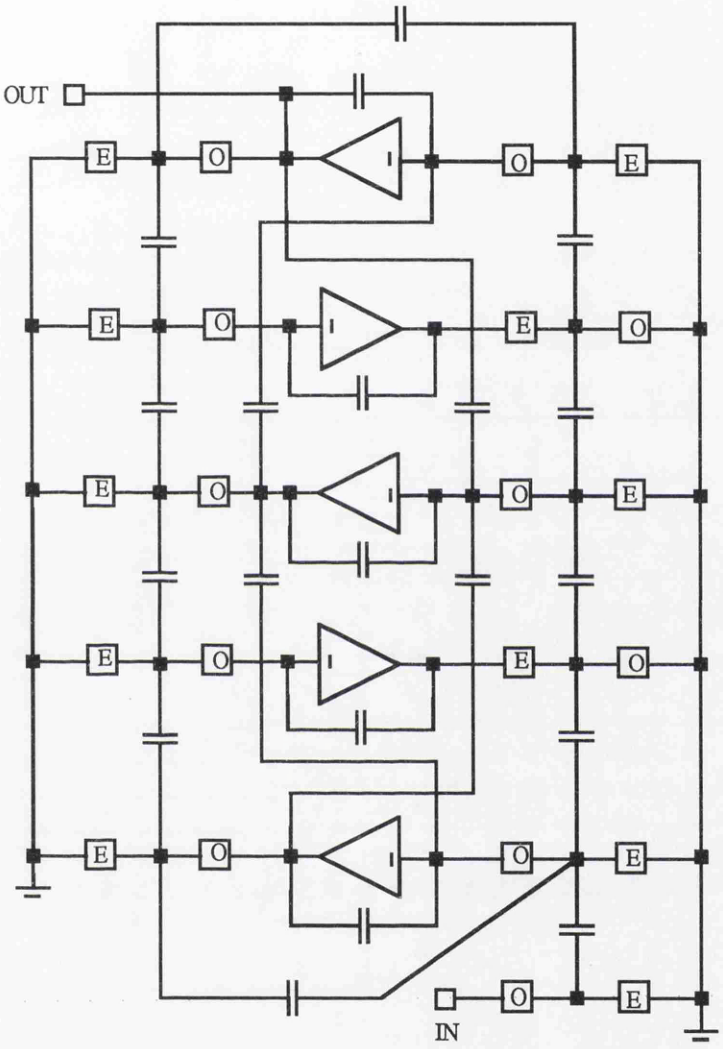


Fig. 5 5th-order elliptic lowpass SC filter

The circuit description file is,

```
5th order elliptic lowpass

.option acct
c1 3 4 27.3p
c2 2 3 30.2p
c3 5 6 79.2p
```

```
c4 4 7 10.9p
c5 8 9 50.6p
c6 3 10 35.7p
c7 7 11 10p
c8 6 12 10p
c9 12 13 146p
c10 5 13 10p
c11 10 14 32.7p
c12 11 15 16.7p
c13 12 21 19p
c14 13 20 15.8p
c15 14 18 29.2p
c16 16 17 59.5p
c17 15 19 10p
c18 20 21 30.8p
c19 18 22 21.1p
c20 19 22 10p

s1 1 2 even
s2 2 0 odd
s3 3 0 odd
s4 10 0 even
s5 14 0 odd
s6 18 0 even
s7 22 0 odd
s8 4 0 odd
s9 7 0 odd
s10 11 0 odd
s11 15 0 odd
s12 19 0 odd
s13 4 5 even
s14 7 8 even
s15 11 12 even
s16 15 16 even
s17 19 20 even
s18 3 6 even
s19 9 10 odd
s20 13 14 even
s21 17 18 odd
s22 22 21 even

e1 0 21 20 0 100k
e2 0 16 17 0 100k
e3 0 13 12 0 100k
e4 0 8 9 0 100k
e5 0 6 5 0 100k

vin 1 0 pulse -1 1 0 8u 8u 984us 2ms

.phase even pwl 0us 5v 8.0us 0v 16.0us 5v
.phase odd  pwl 0us 0v 8.0us 5v 16.0us 0v

.tran 0 3.992m 8u
.plot tran v(20) v(1)

.end
```

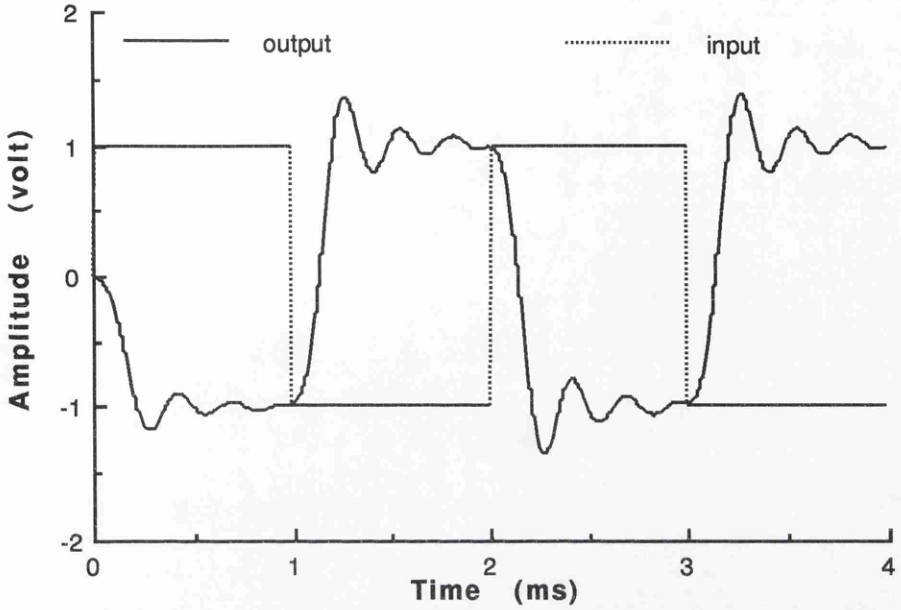


Fig. 5.1 Input and output waveform of 5th order filter

References

[1] Z. Q. Shang and J. I. Sewell, "SCNAP4 User's Guide (version 1.6)", Department of Electronics and Electrical Engineering, University of Glasgow, 1994

[2] "Analog Artist Design System SPICE Socket for Mixed Signal", Analog Artist Version A2.4 Pre-Release, Cadence Design Systems, Inc., May 1, 1991

APPENDIX C: SCNAPDIS USER'S GUIDE

SCNAPDIS User's Guide

Version 1.0

Z Q Shang and J I Sewell

(March 1995)

Department of Electronics and Electrical Engineering

University of Glasgow

Glasgow G12 8LT

1. Introduction

SCNAPDIS is an extension of the time domain analysis facility of SCNAP4 [1] to perform the analysis of dominant nonlinearities in sampled-data analogue networks. Many highly efficient linear techniques which were used effectively in SCNAP4 have been incorporated. With the built-in FFT facility, the dominant nonlinear charactoristics in SC or SI networks can be addressed efficiently. The program is written in C.

2. Element description

The input format of SCNAPDIS is very similar to that of SPICE and is of the free format type. The first line of the input file must be a title line, and the last line must be a **.end** card. The order of the remaining lines is arbitrary. SCNAPDIS is not case-sensitive.

2.1 Resistor (linear or nonlinear)

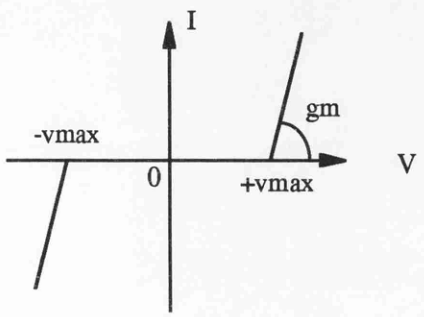
General form:

rxxxxxx node1 node2 val [gm = val vmax = val]

Examples:

r1 1 2 100 gm = 100 vmax=5v

rin 7 3 4.7k



node1 and **node2** are the two nodes to which the element is connected. **val** is the resistance value and may be positive or negative but not zero. If the resistance is greater than 1E30, the conductance will be set to zero. When **gm** and **vmax** were specified, the resistor is considered as a nonlinear resistor with the I-V charactoristic as shown in the figure. It should be stated that the original reisitance value is overriden by a piecewise linear resistance value.

2.2 Capacitor

General form:

cxxxxxx node1 node2 val

Examples:

c1 1 2 100pf
cb 7 3 4.7nf

node1 and **node2** are the two nodes to which the element is connected. **val** is the capacitance value and can be positive, negative or zero.

2.3 Inductor

General form:

lxxxxxx node1 node2 val

Examples:

l1 1 2 100mH

la 2 3 4.55e-3

node1 and **node2** are the two nodes to which the element is connected. **val** is the inductance value and could be positive, negative or zero.

2.4 Switch

General form:

sxxxxxx node1 node2 clknam [ron = val [roff = val]]

Examples:

s1 4 7 even

sa 2 3 clk1 ron = 4.7k

s7 8 5 odd ron = 5.7k roff = 4Meg

node1 and **node2** are the two nodes to which the element is connected. **clknam** is the name of the clock waveform which controls the switch and must be defined elsewhere. **ron/roff** are optional switch on and off resistances respectively. If the values are not specified then default values are assumed. The default values can be specified by using the option card. However, they cannot override the values which have been set in element card.

2.5 Dependent Sources

SCNAPDIS allows circuits to contain linear or nonlinear dependent sources. Basically they are characterised by any of the four following equations

I = G*V V = E*V I = F*I V = H*I

where G, E, F and H are constants representing transconductance, voltage gain, current gain and transresistance, respectively.

2.5.1 Voltage-Controlled Current Source

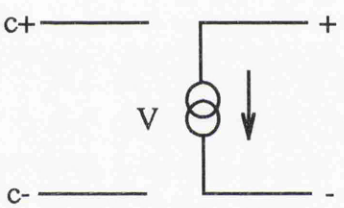
General form:

gxxxxxx cnode+ cnode- node+ node- val [ipos = val [ineg = val]]

Examples:

g1 2 0 5 0 0.1mmho

g2 1 2 3 0 50mho ipos=5mA



node+ and **node-** are positive and negative source nodes, respectively. Current flow is through the source from the positive node to the negative node. **cnode+** and **cnode-** are the positive and negative controlling nodes, respectively. **val** is the transconductance value. **ipos** and **ineg** are two current limiting parameters which can be used to describe the nonlinear characteristic of transconductance.

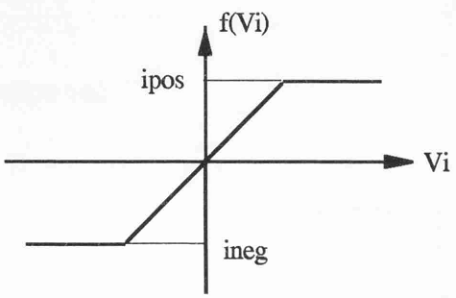


Fig. 1 Nonlinear transconductance

2.5.2 Voltage-Controlled Voltage Source

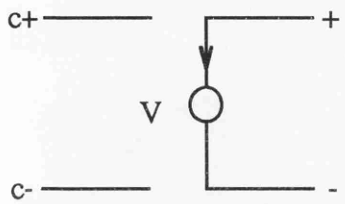
General form:

exxxxxx cnode+ cnode- node+ node- val [vpos = val [vneg = val]]

Examples:

e8 0 5 4 0 1k vpos = 5v vneg = -3v

e1 2 3 5 1 100k



node+ and **node-** are positive and negative source nodes, respectively. **cnode+** and **cnode-** are the positive and negative controlling nodes, respectively. **val** is the voltage gain. **vpos** and **vneg** are two voltage limiting parameters which can be used to describe nonlinear voltage gain as shown in Fig. 2

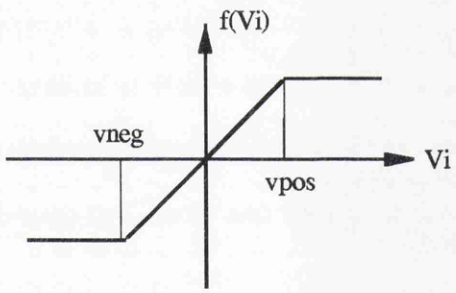


Fig. 2 Nonlinear voltage gain

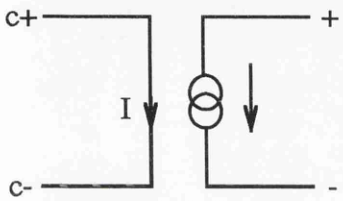
2.5.3 Linear Current-Controlled Current Source

General form:

fxxxxxx node+ node- vname val

Examples:

f1 4 7 vz1 5.0



node+ and **node-** are positive and negative source nodes, respectively. Current flow is through the source from the positive node to the negative node. **vname** is the name of controlling source through which the controlling current flows. The controlling source can be a voltage source, or a voltage-controlled voltage source, or a current-controlled voltage source or an inductor. The direction of positive controlling current is from the positive controlling node, through the source, to the negative controlling node. **val** is the current gain.

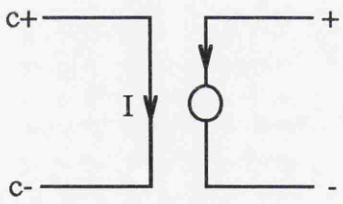
2.5.4 Linear Current-Controlled Voltage Source

General form:

hxxxxxx node+ node- vname val

Examples:

h1 5 17 vz1 0.5k



node+ and **node-** are positive and negative source nodes, respectively. **vname** is the name of controlling source through which the controlling current flows. The controlling source can be a voltage source, or a voltage-controlled voltage source, or a current-controlled voltage source or an inductor. The direction of positive controlling current is from the positive controlling node, through the source, to the negative controlling node. **val** is the transresistance value.

2.6 Independent Sources

General form:

vxxxxxx node+ node- [source function]

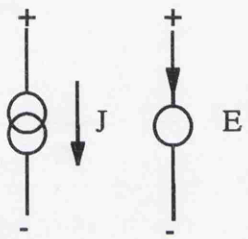
ixxxxxx node+ node- [source function]

Examples:

vin 1 0 dc 5.0v

vs 7 3 step 1.0v

isrc 4 9 sin 0.0 100mA 5kHz



node+ and **node-** are positive and negative source nodes, respectively. For a voltage source, positive current is assumed to flow from the positive node, through the source and out from the negative node. A current source of positive value will force current to flow from the **node+**, through the source and from the **node-**.

2.7 Source function

For time-domain analysis, no restrictions are put on the number of independent sources. A source function must be specified for each source and four of these are available in SCNAPDIS:

2.7.1 Step function

General form: **step val**

Examples:

vin 1 0 step 0.1v

isrc 3 7 step 10mA

2.7.2 Impulse function

General form: **pulse val**

Examples:

vs 1 0 pulse 3v

is 3 7 pulse 120mA

2.7.3 DC source

General form: **dc val**

Examples:

vin 1 0 dc 5.0v

isrc 3 7 dc 3mA

2.7.4 Sinusoidal

General form: **sin vo va freq**

Examples:

vs 1 0 sin 0.1v 1.0v 10kHz

isrc 3 7 sin 0 20mA 98kHz

The function is defined as **vo + va*sin(two*pi*freq*time)**

3. The clock description

To describe the clock signals that control the switches of the network, the **.phase** card is used.

General form:

.phase clknam pwl time0 v0 [[[time1 v1] time2 v2] ...]

Examples:

.phase clk1 pwl 0us 0v 5us 1v 10us 0v

.phase clk2 pwl 0us 1v 10us 0v 20us 1v

The **clknam** is the clock name that controls switches. **pwl** is a key word which means that the clock waveform is periodically repeated and described in a piecewise linear manner. It is defined by pairs of numbers (time and value) for each of the breakpoints in a waveform. Each waveform need only be described over one complete period, irrespective of the periods of other clocks of the whole system. The value associated with each time is the value of the waveform at the instant immediately after that time.

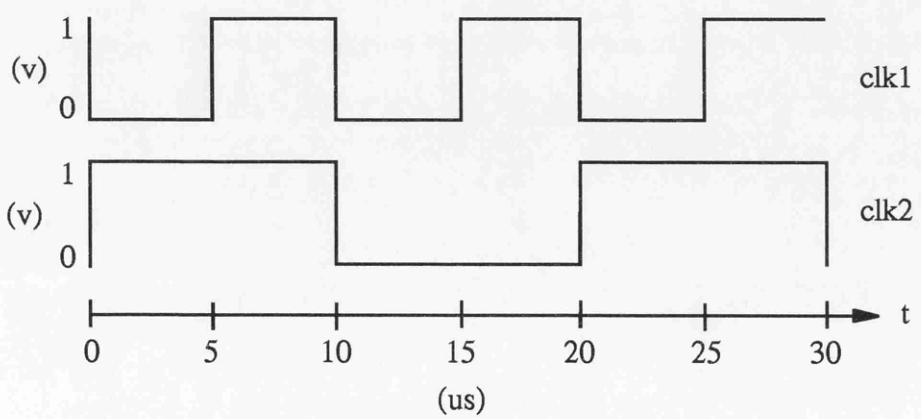


Fig. 3 Clock wave form definition

4. Subcircuit

SCNAPDIS provides a powerful subcircuit facility. A subcircuit can be formed with any elements supported by SCNAPDIS.

4.1 Subcircuit definition

General form:

.subckt subnam node1 [node2 node3 ...] [(parm1 [parm2 parm3 ...])]

Examples:

.subckt opamp 1 2 3 4 (rin, rout, gain)

Subcircuit definition starts from **.subckt** card. **subnam** is the name of the subcircuit; **node1, node2**... are subcircuit's external nodes; **parm1, parm2**... are parameters of the subcircuit and separated by space or comma. Subcircuits having the same circuit structure but different parameters only need one subcircuit definition. After **.subckt** card, there are a series of element cards which describe the subcircuit structure. Subcircuit should use **.ends** card to finish its definition. (See following) No control cards and options cards are allowed within a subcircuit definition. However nested subcircuit definition is possible. Any nodes in subcircuit definition except those in **.subckt** card are considered as local. Ground node is always global.

4.2 Subcircuit end card

General form:

.ends subnam

Example:

.ends opamp

This card should be the last line of any subcircuit definition. **subnam** denotes the subcircuit name.

4.3 Subcircuit call

General form:

xyyyyyy node1 [node2 node3 ...] subnam [(parm1 [parm2 parm3 ...])]

Examples:

x1 3 0 2 4 opamp (800k 100, 1e5)
xa 1 2 3 block1

SCNAPDIS treats subcircuit as a pseudo element. **node1, node2**... are nodes which connect to the subcircuit. **subnam** represents the subcircuit name, followed by a series of parameter values. The node and parameter sequence should be the same as defined in corresponding **.subckt** card. If the user wants to reference an element in a subcircuit, the element name should be written as follows; from left to right, first the element name, followed by an underline, then the name of subcircuit call. For example, **c1** is in a subcircuit named **opamp** and the subcircuit call is **x1**. To reference **c1**, one should write it as **c1_x1**. If there are nested subcircuits, then write all subcircuit calls from inside to outside with an underline between each. eg. **e1_xa_xb_xc** .

5. Analyse control description

SCNAPDIS supports time-domain analysis for both switched capacitor circuits, switched current circuits and continuous time circuits.

General form:

.tran tstart tstop tstep

Examples:

.tran 0 5ms 0.03ms
.tran 0 3200us 16us

tstart is the time at which the analysis is to start and must be greater than or equal to zero. **tstop** is the time at which the analysis is to stop and must be greater than **tstart**. **tstep** is the printing or plotting increment. **tstep** should be greater than zero.

6. Option facilities

General form:

.options opt1 opt2... (or opt=optval...)

Examples:

.options ron = 10 ohm

The above card means that all switch-on resistances are set to 10 ohm. SCNAPDIS allows user to select following options in an arbitrary order.

options	function
order = x	specify approximation order of input source for transient analysis (default is 5)
roff = x	set switch-off resistances which have not been specified in switch cards. (default 1E10 ohm)
ron = x	set switch-on resistances which have not been specified in switch cards. (default 1 ohm)
acct	print out the run time statistics.
fft_size	number of points for FFT (must be power of two)
fft_fc	sampling frequency
fft_fd	fundamental frequency of intrest
fft_warmup	number of points at the beginning which are ignored. The number remaining must be a power of two

7. Sample and hold

General form:

.sample input clknam

.sample output clknam

Examples:

.sample output clk2

.sample input clk1

SCNAPDIS allows users to study sample and hold effects for both input signal and output response. **input** and **output** are key words and refer to signal type. **clknam** is the clock name which describes sample and hold time intervals. These time intervals

correspond to certain basic time slots. For input signals, it means that input is only supplied in defined clock phases. Similarly, for output sample and hold effect, the outputs are only observed in defined clock phases.

8. Plot card

General form:

.plot pltype ov1 < ov2...>

Example,

.plot tran v(2) v(3) i(vin#branch)

After each successful run, an output file named rawfile will be created automatically. It is in the same format as produced by SPICE3. So it directly fits into SPICE post-processor NUTMEG. **pltype** denotes the required analysis type. (transient in this case) For transient analysis, the output variable types are node voltage and branch current. To reference a branch current, the branch name should be formed correctly. For most of the elements, it is the combination of an element name, which produces the branch current, and "#branch" . For current-controlled voltage source, "#contbranch" is needed to reference its controlling branch.

9. Examples

Two circuits are used to show applications of SCNAPDIS.

(A) A phase-compaensated positive-gain amplifier

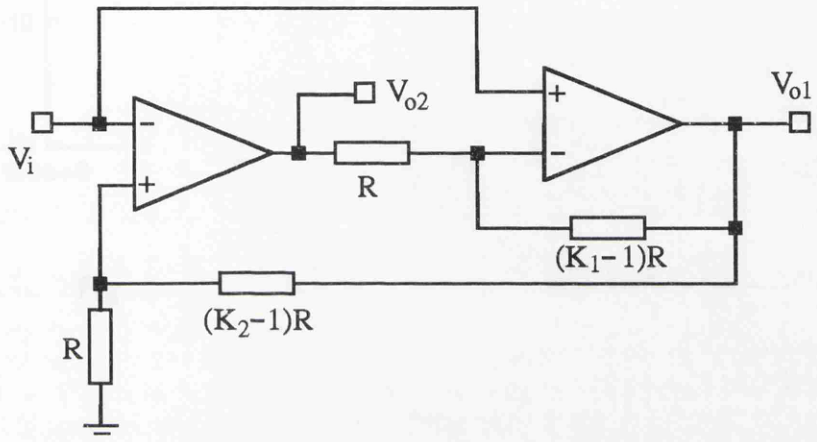


Fig.4 A phase-compensated positive-gain amplifier

The circuit description file is,

```
phase compensated positive-gain amplifier

.option acct

r1 2 0 1k
r2 3 4 1k
r3 2 5 2k
r4 4 5 2k

.subckt amp 1 2 3
 e1 2 1 3 0 1e4 vpos=15
.ends amp

x1 1 2 3 amp
x2 4 1 5 amp

vin 1 0 sin 0 6.0 10k

.tran 0 1e-4 1e-6
.plot tran v(5) v(3) v(1)
.end
```

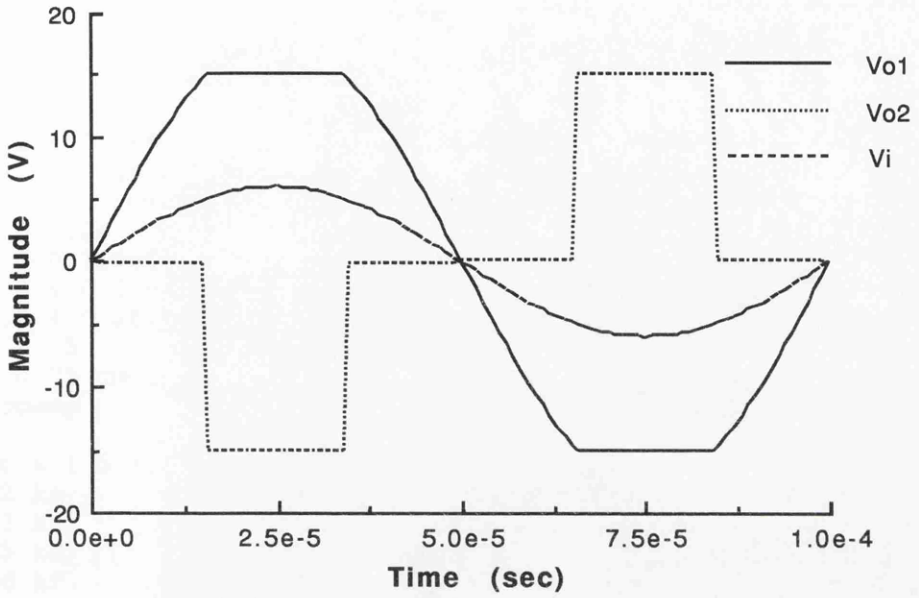


Fig. 5 Simulated waveforms for the circuit in Fig. 4

(B) A 7th-order Chebyschev lowpass filter

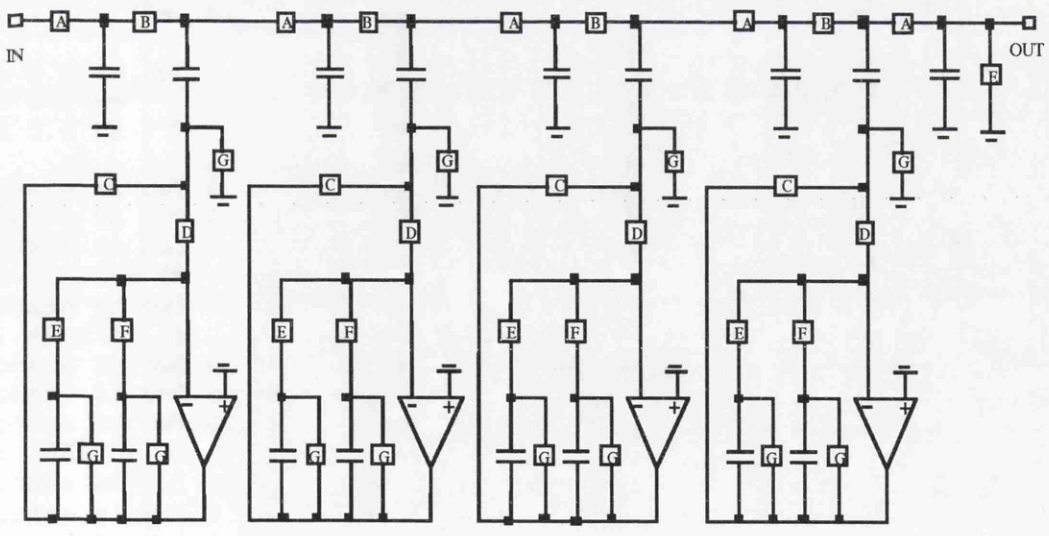


Fig. 6 7th-order chebtschev SC filter

The netlist is given as follows,

```
title '7th order chebyschev lpf (fettweiss)'
```

```
.option acct

vin 1 0 sin 0 7 500

.subckt opamp 1 2 4
* e2 1 2 4 0 1e4 vpos = 5 vneg = -3
 e2 1 2 4 0 1e4 vpos = 5
* g1 1 2 3 0 50mho ipos=5mA
* ri 3 0 100k
* ci 3 0 0.318uf
* e2 0 3 4 0 2 vpos=7 vneg=-6.5
* g2 3 0 4 0 26.7mmho
* ro 4 0 75
* ro 4 0 75 gm=100 vmax=5
.ends opamp

.subckt x 1 3 (ca,cb cc cd)
 s1 1 2 ka
 s2 2 3 kb
 s3 4 5 kd
 s4 5 6 kf
 s5 7 8 kg
 s6 4 0 kg
 s7 4 8 kc
 s8 5 7 ke
 s9 6 8 kg
 c1 2 0 ca
 c2 3 4 cb
 c3 7 8 cc
 c4 6 8 cd
 x1 0 5 8 opamp
.ends x
*
x1 1 2 x (26.5pf,8.32pf 6.355pf 7.524pf)
x2 2 3 x (78.262pf,5.341pf 5.0pf 5.032pf)
x3 3 4 x (87.022pf,5.341pf 5.032pf 5.0pf)
x4 4 5 x (78.262pf,8.324pf 7.52pf 6.335pf)
*
c1 6 0 26.5pf
s1 6 0 ke
s2 5 6 ka
*
.phase ka pwl 0us 1v 13.88889us 0v 41.66667us 1v
.phase kb pwl 0us 0v 20.83333us 1v 34.72222us 0v 41.66667us 0v
.phase kc pwl 0us 0v 6.944444us 1v 13.88889us 0v 27.77778us 1v
34.72222us 0v 41.66667us 0v
.phase kd pwl 0us 1v 6.944444us 0v 20.83333us 1v 27.77778us 0v
41.66667us 1v
.phase ke pwl 0us 0v 20.83333us 1v 41.66667us 0v
.phase kf pwl 0us 1v 20.83333us 0v 41.66667us 1v
.phase kg pwl 0us 0v 13.88889us 1v 20.83333us 0v 34.72222us 1v
41.66667us 0v
.phase kh pwl 0us 0v 13.88889us 1v 20.83333us 0v 41.66667us 0v
.phase ki pwl 0us 1v 13.88889us 0v 20.83333us 1v 41.66667us 1v
.phase kj pwl 0us 0v 13.88889us 1v 20.83333us 0v 41.66667us 0v
*
.option fft_size=1040 fft_fc=24k, fft_fd=0.5k fft_warmup=16
.tran 0 4.175m 41.67u
.sample output kj
.plot tran  v(6)
.plot ac vdb(6)
.end
```

Amplifier model for asymmetrical saturation effects

```
.subckt opamp 1 2 4
 e2 1 2 34 0 1e4 vpos = 7 vneg = -3
.ends opamp
```

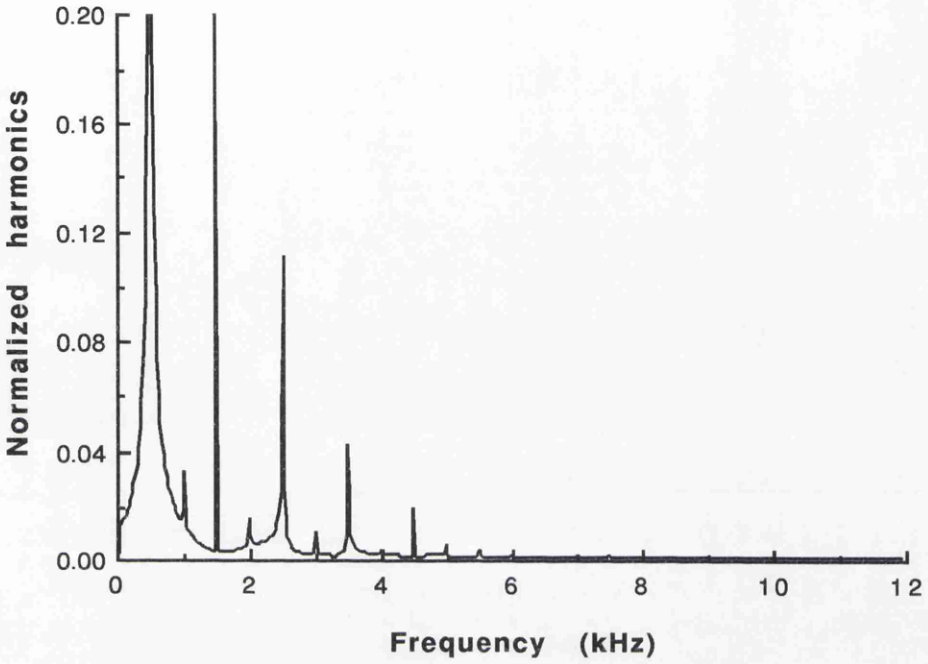


Fig. 7 FFT spectrum for 7th order lowpass SC filter,
with asymmetric amplifier saturation: 500Hz excitation

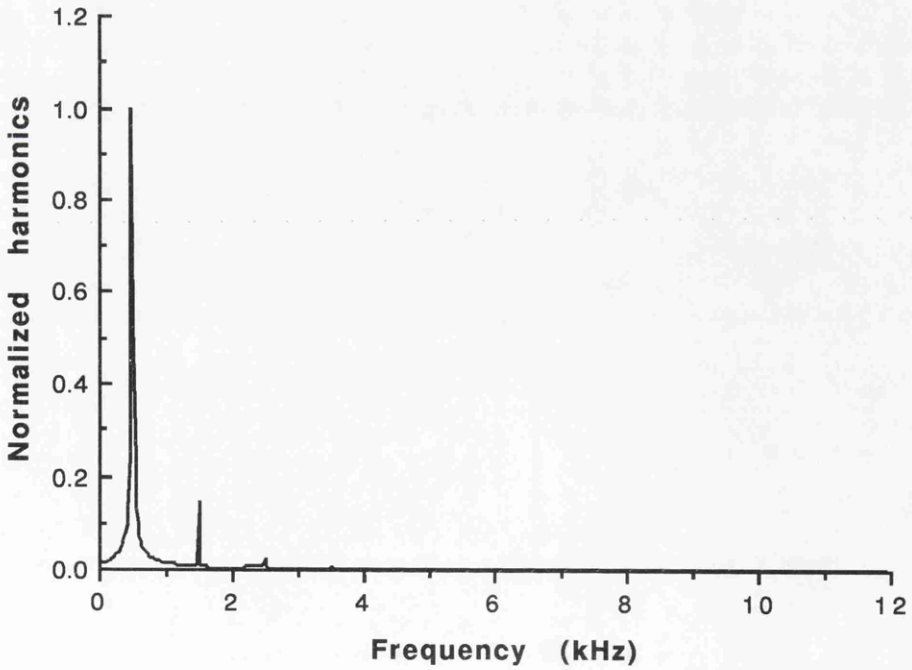


Fig. 8 FFT spectrum for 7th order lowpass SC filter with finite slew-rate

Amplifier model for finite slew-rate effects

```
.subckt opamp 1 2 4
  g1 1 2 3 0 50mho ipos = 5mA
  ri 3 0 100k
  ci 3 0 0.318uf
  g2 3 0 4 0 26.7mmho
  ro 4 0 75
.ends opamp
```

References

[1] Z.Q. Shang and J. I. Sewell, "SCNAP4 User's Guide (version 1.6)", Department of Electronics and Electrical Engineering, University of Glasgow, 1994