Analysis of
Analog Sampled Data Circuits

by

Bijan Raahemi

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical Engineering

Waterloo, Ontario, Canada, 1996

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

Complex systems consist of a core of digital signal processing (DSP) buffered from external environments by analog interface circuitry. "Analog sampled data" circuits can be predominantly found in the data conversion portions of these mixed-mode integrated circuits. Delta-Sigma modulator is an example. Analog sampled data circuits also perform signal processing tasks, and are used in many filtering applications. Switched-capacitor, and switched-current circuits are the examples of these applications.

Sampled data circuits are dual time systems that contain a rapidly varying clock and a slowly varying input signal. As a result, most of the simulation effort is dedicated to the transient analysis at switching instants, where information is usually not needed. The important required information is the response of the circuit at the end of the clock period, when it reaches the steady state. Searching for a method that can provide the solution of the circuit at discrete instants of time led to the "sampled data simulation" technique [1].

This thesis looks at different aspects of analog sampled data systems. It extends the idea of sampled data simulation to provide an accurate and efficient method of computing the time domain sensitivity of linear circuits. The method is applied to sensitivity analysis of an important class of sampled data systems, Delta-Sigma modulators. We also provide efficient methods for analysis of switched networks, including the group delay and group delay sensitivity of periodically switched linear networks in general, and harmonic distortion of switched-current circuits in particular. Sources of distortion in switched-current circuits are examined, and both upper and lower bounds are derived on total harmonic distortion of a current memory cell.

Acknowledgements

" All praise is due to Allah, the lord of the worlds.
Thee do we serve and Thee do we beseech for help. "

I would like to thank my supervisors, Professor Ajoy Opal and Professor Jiri Vlach, for their valuable guidances and constant supports. I have been always enjoying their friendly relationship, and useful advice. I am also indebted to them for their partial financial support.

I would like to express my gratitude to Professor Jim Barby for his careful reading of this thesis. I am also grateful to the members of my Ph.D. examination committee: Prof. M. Nakhla, Prof. M.I. Elmasry, and Prof. D. Brodie.

I would like to express my appreciation to the Ministry of Culture and Higher Education of the Islamic Republic of Iran for the financial support and sponsorship.

Finally, my most profound gratitude goes to my mother, who spent a lonely time in Iran, for her unconditional love and support. None of this could have been achieved without her enthusiasm and prayer.

Bijan Raahemi
December 12, 1996
Waterloo, Ontario, Canada

To my wife
for her encouragement, help, and patience

Contents

List of Tables

List of Figures

List of Symbols

$\mathbf{B}$	The matrix that transfers the final conditions of each phase to the initial conditions of the next phase
$\mathbf{C}$	Capacitance matrix
CCO	Current-Controlled Oscillator
CFT	Clock Feed Through
$\mathbf{d}$	Selector vector that determines the output (output = $\mathbf{d}^t\,\mathbf{X}$)
DSM	Delta-Sigma Modulator
$\mathbf{F_C}$	Vector of final conditions
$\mathcal{F}$	Fourier transform
$\mathcal{F}^{-1}$	Inverse of Fourier transform
$\mathbf{G}$	Conductance matrix
GRPSEN	The program for computation of group delay and the group delay sensitivity of periodically switched linear networks
h	Parameter value, element variable
$\mathbf{I}$	Identity matrix
$\mathbf{I_C}$	Vector of initial conditions
$\mathcal{L}^{-1}$	Inverse of Laplace transform
$\mathbf{M}$	Transition matrix for the original network
$\mathbf{M_0}$	Transition matrix for the inconsistent initial condition
$\mathbf{M_s}$	Transition matrix for the sensitivity network
Opamp	Operational amplifier
$\mathbf{P}$	Constant vector for the original network
$\mathbf{P_0}$	Constant vector for the inconsistent initial condition
$\mathbf{P_s}$	Constant vector for the sensitivity network

PSLN	Periodically Switched Linear Network
SC	Switched-Capacitor
SDS	Sampled Data Simulation
SDSEN	The program for transient and sensitivity analysis of linear circuits based on sampled data simulation method
SI	Switched-Current
SNR	Signal to Noise Ratio
T	Time step
$\mathbf{R}$	System matrix ($\mathbf{R} = \mathbf{G} + s\mathbf{C}$)
THD	Total Harmonic Distortion
$\mathbf{W}$	Source vector in Laplace domain
$\mathbf{w}$	Source vector in time domain
$\mathbf{X}$	Nodal vector in Laplace domain for the original network
$\mathbf{x}$	Nodal vector in time domain for the original network
$\mathbf{Z}$	Nodal vector in Laplace domain for the sensitivity network
$\mathbf{z}$	Nodal vector in time domain for the sensitivity network

Chapter 1

Introduction

Analog sampled data systems are used as filters and also as interfaces between analog and digital circuits. Complex systems consist of a core of digital signal processing (DSP), buffered from external environments by analog interface circuitry. It is now becoming common to find a single mixed analog and digital (mixed-mode) integrated circuit that contains both the digital signal processor and all the interface circuits required to interact with the outside world, which is inherently analog (Fig.1.1). It is in the data conversion portions of these mixed-mode chips that analog sampled data circuits can be predominantly found. Analog sampled data systems also perform signal processing tasks. As a result, they are used in many filtering applications, such as anti-aliasing filter in Fig.1.1.

Switched-capacitor circuits, switched-current circuits, and Delta-Sigma modulators are examples of sampled data systems. The switched-capacitor technique eliminates resistors from the design of analog filters. It is based on the idea that a periodically switched capacitor can be used to simulate a resistor (provided that the switching frequency is much higher than the signal frequencies of interest).

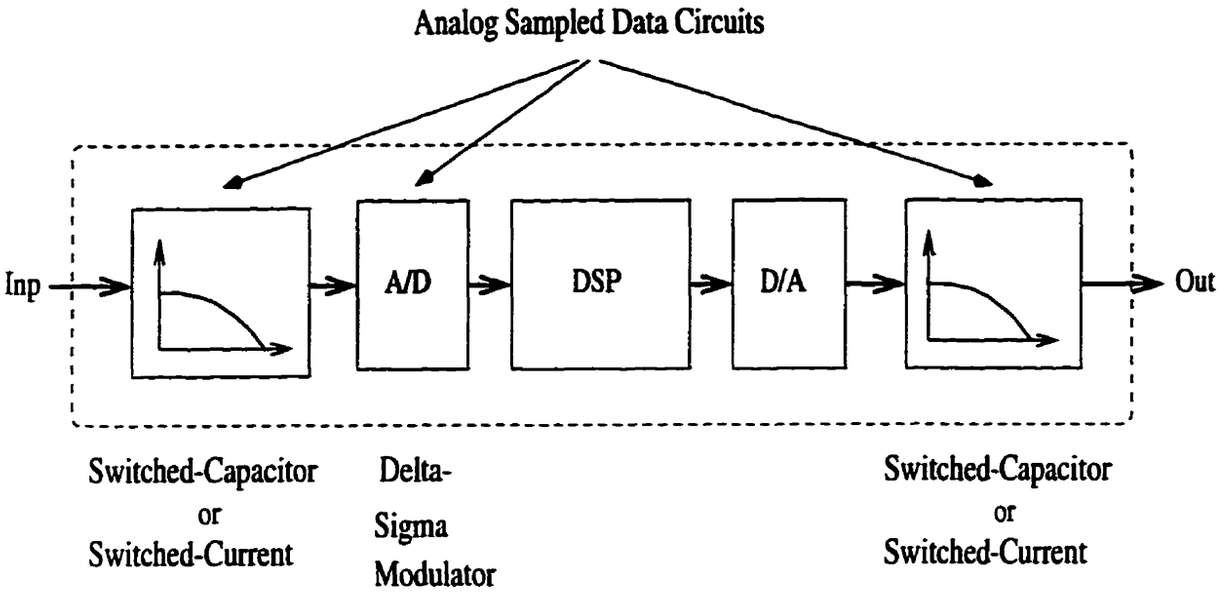


Figure 1.1: A mixed analog and digital (mixed-mode) integrated circuit.

Switched-current circuits do not require linear floating capacitors, and can be integrated in a standard digital CMOS process. This allows integrated circuit manufacturers to implement both digital and analog circuits on the same chip with the existing low-cost CMOS processes. Operating in the current mode, and eliminating the need for a large voltage swing, make the switched-current technique suitable for low voltage applications.

Delta-Sigma modulators are important blocks of oversampled A/D converters. Oversampling A/D converters depend on relatively simple and modest analog circuitry. They are less insensitive to circuit imperfections and component mismatches, since they usually employ a simple two-level quantizer embedded within a feedback loop.

If analog sampled data systems are to be widely used in signal processing and data conversion, they require a suite of computer aided design (CAD) tools to

simulate the circuits, perform the sensitivity analysis, and automate the design processes. General purpose analog simulation tools, such as SPICE [2], can be used for simulation of these circuits. However, these tools are provided for simulation of general nonlinear circuits and do not simultaneously provide both the accuracy and speed necessary for simulation of specialized circuits.

Sampled data circuits are dual time systems that contain a rapidly varying clock and a slowly varying input signal. As a result, most of the simulation effort is dedicated to the transient analysis at switching instants, where information is usually not needed. Sampled data systems are usually working at frequencies that allow the output become stable. The important required information is therefore the response of the circuit at the end of the clock period, when it reaches the steady state between the switching instants. Consider, for instance, a Delta-Sigma modulator for which we collect the output data at the end of the phase, when modulator reaches the steady state, and take FFT of the data to obtain the frequency spectrum. Searching for a method that can provide the solution of the circuit at discrete instants of time led to the "sampled data simulation" technique proposed in [1]. Sampled data simulation (SDS) is an efficient, accurate, and stable method for transient analysis of lumped linear time invariant circuits. It generates some constant matrices before the simulation, and provides the transient solution at each time point by performing only one matrix-vector multiplication. The time points are equally spaced, and can be chosen arbitrarily regardless of the circuit time constants.

We extend the idea of sampled data simulation to compute the time domain sensitivity of linear circuits (switched and unswitched). There are several reasons for the importance of sensitivity in analog circuit design. In addition to providing more insight into the behavior of a physical system, sensitivity function plays an

important part in the design and optimization of reliable circuits. The circuit manufacturing process results in the spread of parameter values, known as the element tolerances. Also, during the lifetime of a manufactured circuit, parameters are subject to change through aging, and environment effects, such as temperature and humidity. A sensitivity analysis is therefore required to find out which circuit parameters are critical, i.e. the network sensitivity with respect to them is very large.

Group delay is another aspect of analog sampled data filters that is analyzed in this thesis. In filter design, the magnitude response requirements are normally considered, and the corresponding phase response is ignored. This is because the reduction of antialiasing by stopband attenuation is the first issue, and phase information can be corrected later. For some applications, such as speech transmission, this consideration is sufficient as the human ear is insensitive to the phase shift. For video applications, however, the distortion caused by system nonlinear phase response is unacceptable. A fast and accurate method for the computation of group delay and the group delay sensitivity of periodically switched linear networks is presented in this thesis.

1.1 Thesis Outline

The objectives of this thesis are: (i) to extend the idea of sampled data simulation to compute the time domain sensitivity of linear circuits (switched and unswitched), and a class of nonlinear circuits– Delta-Sigma modulators, (ii) to provide efficient method for analysis of group delay and its sensitivity in periodically switched linear networks, (iii) to compute the harmonic distortion in switched-current circuits. Fig.1.2 shows the various classes of analog sampled data circuits, and the different

types of analysis considered in this thesis.

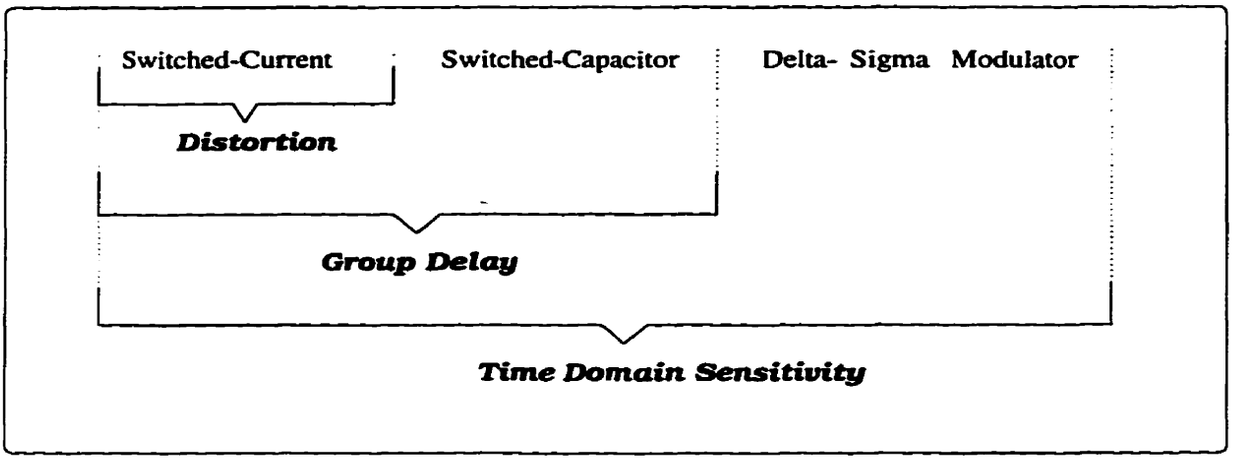


Figure 1.2: The outline of the thesis.

Each chapter of this thesis looks at one aspect of sampled data systems. Therefore, each chapter starts with its own introduction in the first section, continued by new material in subsequent sections as my contribution to each topic.

Chapter 2 is dedicated to time domain sensitivity of linear circuits using sampled data simulation. It presents a new method which is accurate, because no approximation is made, and efficient because some parts of the computations are performed only once, in a pre-processing step before simulation starts. We discuss the applications of the method in both *sensitivity* and *adjoint* networks, and illustrate the theory with some examples. A program, called SDSEN, was written based on the theory developed in this chapter, and applied to some examples. Extension of sampled data simulation to the case of inconsistent initial conditions is another subject presented in this chapter.

Chapter 3 employs the theory implemented in chapter 2 and applies it to an important class of sampled data circuits, Delta-Sigma modulators. This chapter

presents the formulation of sensitivity networks for Delta-Sigma modulators, from which the sensitivity of the output magnitude with respect to any circuit elements is obtained. The method can be applied to all types/configurations of modulators if all elements, except the comparator, are linear. We establish a set of equations for the sensitivity network by taking the differential of the original network with respect to element h, and solve them in parallel to obtain the sensitivity of all nodes with respect to that element value. During the simulation we also need sensitivity of the output of the comparator with respect to its input. We distinguish between the unclocked and clocked comparators to explain their behavior.

Chapter 4 presents a fast and accurate method for the calculation of group delay and group delay sensitivity of periodically switched linear networks. Group delay is an important measure used in the design of precision filters. In a distortionless filter, the magnitude of the transfer function and the group delay must be flat over the passband. Since the method uses the MNA formulation of the circuit, and considers all types of linear elements, it can be used to simulate the switched-capacitor networks when the switches are replaced by their resistances, and the operational amplifiers have frequency-dependent as well as the other nonideal linear characteristics. Switched-current networks are another application, as long as the MOS transistors are modeled by linear components such as dependent sources, capacitors, and resistors. A program, called GRPSN, was written based on the theory developed in this chapter, and applied to some examples.

Chapter 5 introduces some nonfiltering applications of switched-current circuits. Switched-current is a relatively new analog sampled-data technique that promises to overcome the problems associated with switched-capacitor circuits. Although filtering applications have received most of the attention among switched-current circuits, there are other analog signal processing tasks that can be performed with

the same fabrication technology and circuit elements as those used in switched-current filters. In this chapter, we present some non-filtering applications including a current-controlled oscillator, a modulator and a full-wave rectifier. We also propose a switched-current oversampling Delta-Sigma A/D converter.

Chapter 6 focuses on the distortion analysis of switched-current networks. Harmonic distortion in switched-current circuits is more severe than in switched-capacitor networks. Clock feed through and mismatch in the transistor threshold voltage are two major sources of distortion in switched-current circuits. These sources are introduced in this chapter, and their contributions to distortion are examined. We also present a general expression for the total harmonic distortion (THD) of nonlinear circuits. Using this expression, we impose both upper and lower bounds on the THD of a switched-current memory cell.

Chapter 7 summarizes the important aspects of the work performed, and gives future research directions.

The thesis is concluded with three appendices. Appendix-A explains the numerical Laplace inversion and the stepping algorithm used in computation of the constant matrices required in SDSEN. Appendix-B presents the intermediate matrix manipulations used in the computation of group delay and group delay sensitivity. Appendix-C explains some selected switched-current building blocks that are used to implement the non-filtering applications.

Chapter 2

Time Domain Sensitivity of Linear Circuits Using Sampled Data Simulation

In the design of any system, it is important to know the effect of the variations of system parameters on the system's performance. In the case of lumped linear time invariant networks, a precise measure of this effect can be expressed in terms of the *sensitivity* to parameter values. The parameter can be a circuit element, such as a resistor or capacitor, or any other characteristic, such as the gain of the operational amplifier. There are several reasons for the importance of sensitivity in analog circuit design :

1. The study of the network sensitivity enhances insight into circuit behavior. By dividing the circuit parameters into critical and non-critical ones, an effective method is provided to simplify circuit models for efficient circuit analysis.

2. Network sensitivity plays an important part in the design and optimization of reliable circuits. The objective of optimization is to minimize the discrepancy between the actual and the desired circuit behavior.

3. During the lifetime of a manufactured circuit, parameters are subject to change through aging, and environment effects, such as temperature and humidity. A sensitivity analysis is therefore required to find out which circuit parameters are critical.

4. The spread of parameter values resulting from the circuit manufacturing process requires the knowledge of the circuit performance in a certain range of parameter values, known as the tolerance range. This generates the need for tolerance analysis.

5. Knowledge of the network sensitivity can be used as a basis for comparing different circuits. It helps the circuit designer in selecting the proper circuit for a specified application.

Various sensitivity definitions of linear networks were introduced in the frequency domain [3]. In this chapter, we focus on the time domain sensitivity analysis using sampled data simulation method. An accurate and efficient method of computing time domain sensitivity is introduced. The method is accurate because no approximation is made, and is computationally efficient because some parts of the computations are performed only once, in a pre-processing step. It is similar to SDS in the manner of generating some constant matrices before simulation, and then providing the sensitivity at each time point by performing only matrix-vector multiplications.

We apply both *sensitivity network* and *adjoint network* [4–6] approaches to compute the time domain sensitivity. In the sensitivity network approach, the sensi-

tivities of all variables with respect to one element are found at all time points. In the adjoint network approach, the sensitivities of one variable with respect to all elements are calculated at one time point. The application of the sensitivity at one instant of time is in the calculation of error gradients for linear networks, and in the time domain sensitivity analysis of objective functions used in circuit optimization [3,7].

A review of sampled data simulation is given in section 1. Sections 2, 3, and 4 present my contributions to this topic. In section 2, I extend SDS to the case of inconsistent initial conditions. In section 3, I present the derivations for the time domain sensitivity using SDS. In this section, the derivations are for the sensitivity network. The adjoint network is discussed in section 4. A program was written in MATLAB, based on the theories developed in this chapter. Appendix-A shows the algorithms used in this program for computation of the constant matrices in the pre-processing step.

2.1 Sampled Data Simulation of Linear Circuits

Sampled data simulation (SDS) of linear circuits proposed in [1] is an efficient, accurate, stable, and explicit method for the transient analysis of lumped linear time invariant circuits. The method formulates a set of finite difference equations in the time domain. The solution of these equations gives the network response at fixed and equally spaced discrete instants of time. The fixed time interval between each solution can be chosen arbitrarily and does not depend on the circuit time constants. The transient solution at each time point requires only one matrix-vector multiplication. The algorithm is a general computer oriented formulation method that can be applied to any linear circuit.

To explain the method, we assume that the linear circuit is formulated using Modified Nodal Analysis (MNA) [3,8]:

$$\mathbf{G}\mathbf{x} + \mathbf{C}\frac{d\mathbf{x}}{dt} = \mathbf{w}(t), \qquad \mathbf{x}(0^-) = \mathbf{x}_0 \tag{2.1}$$

where $\mathbf{G}$ is the conductance matrix ($m \times m$), $\mathbf{C}$ the capacitance matrix ($m \times m$), $\mathbf{w}(t)$ the input source vector ($m \times 1$), $\mathbf{x}(t)$ the unknown vector ($m \times 1$) containing nodal voltages and some branch currents needed for MNA, and $\mathbf{x}_0$ the initial condition vector ($m \times 1$). The Laplace transform of (2.1) is

$$\mathbf{R}\mathbf{X} = \mathbf{W} + \mathbf{C}\mathbf{x}(0^-), \tag{2.2}$$

where $\mathbf{R} = \mathbf{G} + s\mathbf{C}$ is the system matrix. The formal solution of (2.2) is

$$\mathbf{X} = \mathbf{R}^{-1}\mathbf{W} + \mathbf{R}^{-1}\mathbf{C}\mathbf{x}(0^-). \tag{2.3}$$

The first term on the right-hand side of (2.3) is related to the zero-state response, and the second term to the zero-input response of the linear circuit. Without loss of generality, we assume that there is one input source, which is a complex exponential $w(t) = e^{st}$. If there is more than one input source, the superposition principle is applied to sum the responses due to individual inputs. Taking the inverse Laplace transform of (2.3) and considering the circuit response at the first time point $t = T$,

$$\mathbf{x}(T) = \mathcal{L}^{-1}\left(\mathbf{R}^{-1}\mathbf{W}\right)\Big|_{t=T} + \mathcal{L}^{-1}\left(\mathbf{R}^{-1}\mathbf{C}\right)\Big|_{t=T} \mathbf{x}(0^-) \tag{2.4}$$

where $\mathcal{L}^{-1}(\cdot)$ denotes the inverse Laplace transform. Define the constant matrices $\mathbf{M}$ and $\mathbf{P}$ as

$$\mathbf{P} = \mathcal{L}^{-1}\left(\mathbf{R}^{-1}\mathbf{W}\right)\Big|_{t=T}, \tag{2.5}$$

$$\mathbf{M} = \mathcal{L}^{-1}\left(\mathbf{R}^{-1}\right)\Big|_{t=T} \times \mathbf{C}. \tag{2.6}$$

P is a vector with dimension $m \times 1$, and **M** is an $m \times m$ matrix. Equations (2.5) and (2.6) are formal definitions for the **M** and **P** matrices. In practice, as explained in the next section, these matrices are computed numerically by the solutions of two system of equations in the time domain, i.e. by integrating the equations over the interval $[0, T]$. Rewrite (2.4) using (2.5) and (2.6)

$$\mathbf{x}(T) = \mathbf{P} + \mathbf{M}\mathbf{x}(0^-).$$

Next, consider the circuit response at the second time point $t = 2T$. In lumped linear networks, any time can be selected as the origin by taking the initial conditions into account. These initial conditions "reset" the problem so that the next calculation can start without any reference to previous history. To move the time origin from $t = 0$ to $t = T$, we must consider $\mathbf{x}(T)$ as the initial conditions, and $w(t) = e^{s(t+T)}$ as the input. The response at the end of the second time slot becomes

$$\mathbf{x}(2T) = \mathbf{P}e^{sT} + \mathbf{M}\mathbf{x}(T).$$

In general, by considering any time instant $t = nT + T$, the complete response is

$$\mathbf{x}(nT + T) = \mathbf{P}\ e^{snT} + \mathbf{M}\mathbf{x}(nT), \tag{2.7}$$

where T is the interval between two subsequent discrete time points at which the response is calculated. Eq.(2.7) computes the response of the circuit in a sampled data manner after equal intervals of time as long as the **M** and **P** matrices are known. These matrices are constant and need to be computed only once, in a pre-processing step before simulation starts.

Sampled data simulation can jump over large time steps and still maintain accuracy whereas simulators like SPICE [2, 9] must take many small time steps to maintain accuracy. In addition, SDS is an explicit method that is not slowed down by iterations or time step control algorithms.

The time step T can be chosen arbitrarily and does not depend on the circuit time constants. Thus, if T is larger than the time steps taken by ordinary integration methods to solve (2.1), the sampled data simulation method will be computationally efficient. This is very evident if the response is needed at many time points. Simulation of oversampled Delta-Sigma modulators is an example of such a case.

2.1.1 Numerical Computation of M and P Matrices

As suggested by (2.5), the vector $\mathbf{P}$ is obtained by setting the initial conditions to zero and numerically integrating the circuit equations (2.1) over the time interval $[0, T]$. According to (2.6), the matrix $\mathbf{M}$ is also obtained by integrating (2.1) while the input is an identity matrix (as explained in the next paragraph), over the interval $[0, T]$, and then postmultiplying the result by the matrix $\mathbf{C}$.

The matrix $\mathbf{M}$ can be computed also by turning all sources off, then considering the circuit response due to initial conditions only, and after a time interval T. This results in

$$\mathbf{x}(T) = \mathbf{M}\mathbf{x}(0^-).$$

As explained in [10], if we choose $\mathbf{x}(0^-)$ to be the j-th column of an $m \times m$ identity matrix, then, after integrating over the interval $[0, T]$, we obtain $\mathbf{x}(T)$ equal to the j-th column of $\mathbf{M}$. This suggests that the j-th column of $\mathbf{M}$ can be calculated by setting the input sources to zero and by exciting only the j-th initial condition (by setting the j-th entry of $\mathbf{x}(0^-)$ to unity and all other entries to zero), and numerically integrating the circuit equations (2.1) over the interval $[0, T]$. The complete $\mathbf{M}$ matrix is obtained by repeating this process for each column. The

reader is referred to [11] for more details about computations of **M** and **P** matrices, especially when the circuit is formulated using the MNA method.

The computation of **M** and **P** requires the time domain solution of (2.1), and hence a numerical integration method for the solution of a set of linear differential equations is required. We use numerical Laplace inversion [12–15] because it provides very accurate results and is equivalent to an absolutely stable, very high order integration method. As shown in [13], the numerical Laplace inversion correctly inverts the first $p+1$ terms of the Taylor series expansion of the time domain response of the network, and is equivalent to an integration method of order p (in our work we used $p = 18$). If a small time step h is used at each time step, then the total truncation error is proportional to h^p and can be made arbitrarily small. Other integration methods such as the Backward Differentiation Formula and Runge-Kutta method [3] are normally lower order methods ($p < 5$), and can not provide the same level of accuracy. Furthermore, finite precision arithmetic on a digital computer limits the smallest time step that can be used with low order integration methods and ultimately the accuracy of the computed results.

Computation Cost

A computer program, called MPgen, was written in MATLAB [16] to compute **M** and **P** matrices. The program accepts the circuit matrices **G** and **C**, and the time interval T, then generates the **M** and **P** matrices. MPgen proceeds with the stepping algorithm [3] that is explained in Appendix-A. The matrix inversions indicated in (2.5) and (2.6) are performed by LU decompositions followed by forward-backward substitutions. The computation cost is of order $O(m^3/3)$ [17]. However, the computation can be performed in sparse with the cost of $O(m^{1.1-1.5})$ [18].

The order of integration used in MPgen is 18, and the local truncation error is set to 10^{-9}, equivalent to 180 dB simulation accuracy per time step. The numerical Laplace inversion performed in MPgen requires complex arithmetic as opposed to real arithmetic in ordinary integration methods, and requires the solution of the network at 5 frequency points for each integration time step. As a result, the cost is usually larger than ordinary integration methods. However, this higher cost can be tolerated because of the accuracy we obtain, and because the calculation of the **M** and **P** matrices is needed only once in a pre-processing step, before simulation starts.

2.2 Inconsistent Initial Conditions

In some situations, especially in networks with ideal switches, it may happen that the initial conditions before switching and just after switching are not the same. Consider the network in Fig. 2.1, where the capacitor on the left is initially charged to $2V$ and the capacitor on the right has no charge. When the switch is closed, there are simultaneously two different voltages on the node: $2V$ from the left capacitor and $0V$ from the right one. This is a case of *inconsistent initial conditions* [19, 20].

In this section, we propose an extension to SDS by generating the transition matrices needed to integrate the circuit equations from nT^- to nT^+. In particular, we are looking for the matrix $\mathbf{M}_0$ and the vector $\mathbf{P}_0$ that provide the following relation at each instant $nT, (n = 0, 1, \ldots, N)$:

$$\mathbf{x}(nT^+) = \mathbf{M}_0\ \mathbf{x}(nT^-) + \mathbf{P}_0\ e^{snT}. \tag{2.8}$$

If $\mathbf{M}_0$ and $\mathbf{P}_0$ are computed once at the beginning of the simulation, then the

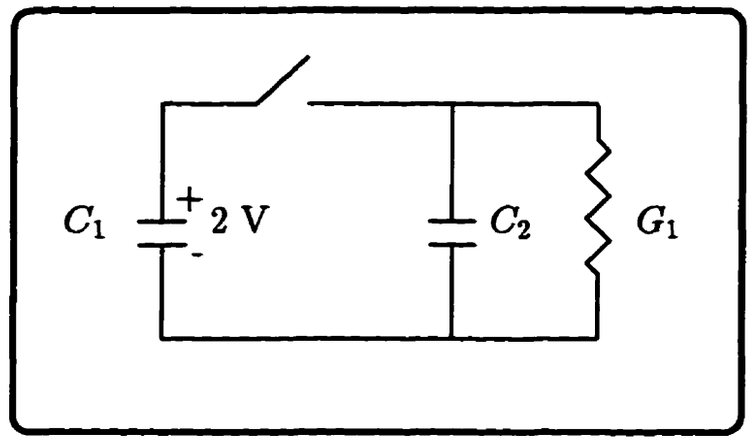


Figure 2.1: Network with inconsistent initial voltages.

inconsistent initial conditions at each instant $nT, (n = 0, 1, \ldots, N)$ can be handled very quickly by performing only a matrix-vector multiplication.

SDS provides the time domain response at either $(0^-, T^-, \ldots, NT^-)$ or $(0^+, T^+, \ldots, NT^+)$. This means that the method can not detect the presence of Dirac impulses at the time instances nT. However, it considers the effect of impulses in the computation of responses at the next time point. To illustrate this, consider the simple RC circuit in Fig. 2.2 with zero initial condition, and the input of Dirac impulses applied at two different time points, $r(t) = \delta(t) + \delta(t - 3T)$. The analytical solution of the circuit provides

$$v_{out}(t) = e^{-t}u(t) + e^{-(t-3T)}u(t - 3T),$$

where $u(\cdot)$ denotes the step function. The output $v_{out}(t)$ at both nT^- and nT^+ are shown in Table 2.1. The effects of Dirac impulses can be seen as the differences between $v_{out}(0^-)$ and $v_{out}(0^+)$, and $v_{out}(3T^-)$ and $v_{out}(3T^+)$.

To simulate the circuit using the sampled data simulation technique, we consider

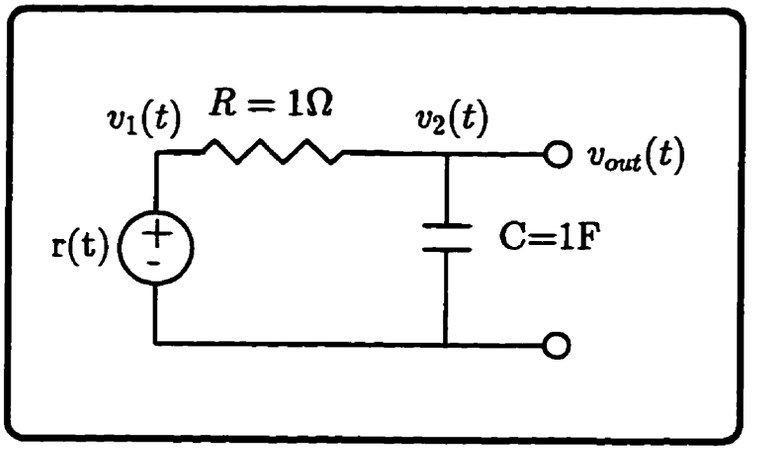


Figure 2.2: RC circuit with Dirac impulse input.

t Time	$v_{out}(t^-)$ Analytical	$v_{out}(t+)$ Analytical	$v_{out}(t)$ SDS
0	0	1	0
T	e^{-T}	e^{-T}	e^{-T}
$2T$	e^{-2T}	e^{-2T}	e^{-2T}
$3T$	e^{-3T}	$e^{-3T}+1$	e^{-3T}
$4T$	$e^{-4T}+e^{-T}$	$e^{-4T}+e^{-T}$	$e^{-4T}+e^{-T}$

Table 2.1: The output of RC circuit at a few time points.

the modified nodal formulation of the circuit, and compute the **M** and **P** matrices.

$$\underbrace{\begin{bmatrix} 1 & -1 & 1 \\ -1 & 1+s & 0 \\ 1 & 0 & 0 \end{bmatrix}}_{\mathbf{R}} \underbrace{\begin{bmatrix} V_1 \\ V_2 \\ I_E \end{bmatrix}}_{\mathbf{X}} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{\mathbf{W}}$$

$$\mathbf{M} = \mathcal{L}^{-1}\left(\mathbf{R}^{-1}\right)\Big|_{t=T} \quad \mathbf{C} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & e^{-T} & 0 \\ 0 & e^{-T} & 0 \end{bmatrix}$$

$$\mathbf{P} = \mathcal{L}^{-1}\left(\mathbf{R}^{-1}\mathbf{W}\right)\Big|_{t=T} = \begin{bmatrix} 0 \\ e^{-T} \\ e^{-T} \end{bmatrix}$$

The time domain response based on (2.7) becomes

$$\begin{bmatrix} v_1(nT+T) \\ v_2(nT+T) \\ i_E(nT+T) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & e^{-T} & 0 \\ 0 & e^{-T} & 0 \end{bmatrix} \begin{bmatrix} v_1(nT) \\ v_2(nT) \\ i_E(nT) \end{bmatrix} + \begin{bmatrix} 0 \\ e^{-T} \\ e^{-T} \end{bmatrix} * g(nT) \qquad (2.9)$$

where, because of the Dirac impulse inputs at $t = 0$ and $t = 3T$,

$$g(nT) = \begin{cases} 1 & \text{if } n = 0 \text{ or } n = 3T \\ 0 & \text{otherwise} \end{cases}$$

Assuming zero initial condition, (2.9) generates the time domain responses listed in the third column of Table 2.1. The response at $t = 0$ is valid for 0^-, not for 0^+. Also the response at $t = 3T$ is valid only for $3T^-$.

A technique, called *two step* method, was proposed in [19] to take into account the effects of Dirac impulses and inconsistent initial conditions. In this technique, we first take a relatively long step forward from 0^- to T^-. At this time point, we have good accuracy for the time domain response, and no Dirac impulse or inconsistent initial conditions are present. Next, we take exactly the same step size backward in time from T^- to 0^+. The error of this step is very low since we are now dealing with a situation without a Dirac impulse or inconsistent initial conditions.

With the combination of the "two step" and "SDS" methods, we generate $\mathbf{M}_0$ and $\mathbf{P}_0$, the matrices needed to integrate the circuit equations from nT^- to nT^+ in

(2.8). First, integrate the circuit equations from nT^- to $(nT+T)$

$$\mathbf{x}(nT+T) = \mathbf{M}\mathbf{x}(nT^-) + \mathbf{P}e^{snT}. \tag{2.10}$$

Then, integrate the equations backward from $(nT+T)$ to nT^+

$$\mathbf{x}(nT^+) = \widehat{\mathbf{M}}\mathbf{x}(nT+T) + \hat{\mathbf{P}}e^{-s(nT+T)}. \tag{2.11}$$

Substituting (2.10) into (2.11) yields

$$\mathbf{x}(nT^+) = \widehat{\mathbf{M}}\mathbf{M}\mathbf{x}(nT^-) + \left[\widehat{\mathbf{M}}\mathbf{P} + \hat{\mathbf{P}}e^{-s(2nT+T)}\right] e^{snT}. \tag{2.12}$$

Comparing (2.12) with (2.8) gives

$$\mathbf{M}_0 = \widehat{\mathbf{M}}\,\mathbf{M} \tag{2.13}$$

$$\mathbf{P}_0 = \widehat{\mathbf{M}}\,\mathbf{P}e^{sT} + \hat{\mathbf{P}}e^{-s(2nT+T)} \tag{2.14}$$

The matrices $\mathbf{M}$, $\mathbf{P}$, $\widehat{\mathbf{M}}$, and $\hat{\mathbf{P}}$ are constant. The matrix $\mathbf{M}_0$ is therefore constant and computed once. Only the last term of $\mathbf{P}_0$ in (2.14) depends on the time point. This is not a big concern because it introduces only a vector-vector addition per time point. In the special case, when the input is a unit step function, $\mathbf{P}_0$ also becomes a constant vector

$$\mathbf{P}_0 = \widehat{\mathbf{M}}\,\mathbf{P} + \hat{\mathbf{P}}.$$

A program was written in MATLAB based on the above equations to generate $\mathbf{M}_0$ and $\mathbf{P}_0$ for switched linear networks. The program was tested on several circuits among which is the circuit in Fig. 2.3. This network was considered in [3] and [19]. It has two switches and two equal phases, the elements have unit values, and the switching frequency is 10 Hz. During the first phase, switch $S2$ is closed and $S1$ is open. In the second phase, the positions of the switches are reversed. A unit

step input is applied. The time domain response is given in Fig.2.4. At the start of phase 1 ($t = 0.1, 0.2, \ldots$), the network has a loop of capacitors and thus a jump occurs at the output. To detect these jumps, we use the $\mathbf{M_0}$ and $\mathbf{P_0}$ matrices. After that, we proceed with $\mathbf{M}$ and $\mathbf{P}$. For instance, at $t = 0.1\ s$, to detect the jump, we compute

$$\mathbf{x}(0.1^+) = \mathbf{M_0}\ \mathbf{x}(0.1^-) + \mathbf{P_0}.$$

Then, to integrate for the rest of the phase, we compute

$$\mathbf{x}(0.15) = \mathbf{M}\ \mathbf{x}(0.1^+) + \mathbf{P}.$$

The solid line in Fig.2.4 shows the response when using the $\mathbf{M_0}$ and $\mathbf{P_0}$ matrices all each switching instants. The dotted line shows the results when we proceed without using $\mathbf{M_0}$ and $\mathbf{P_0}$ (i.e. when we discard inconsistent initial conditions). Since the use of $\mathbf{M_0}$ and $\mathbf{P_0}$ costs only one matrix-vector multiplication, it is advisable to use them to obtain more reliable answers.

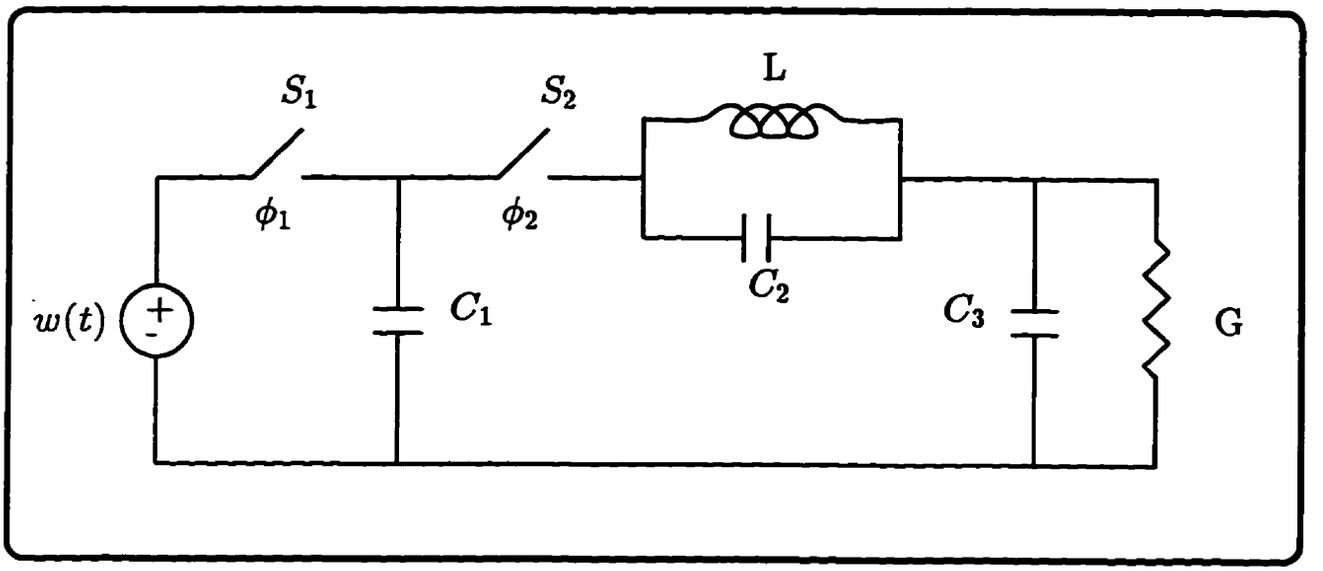


Figure 2.3: Switched network with inconsistent initial conditions.

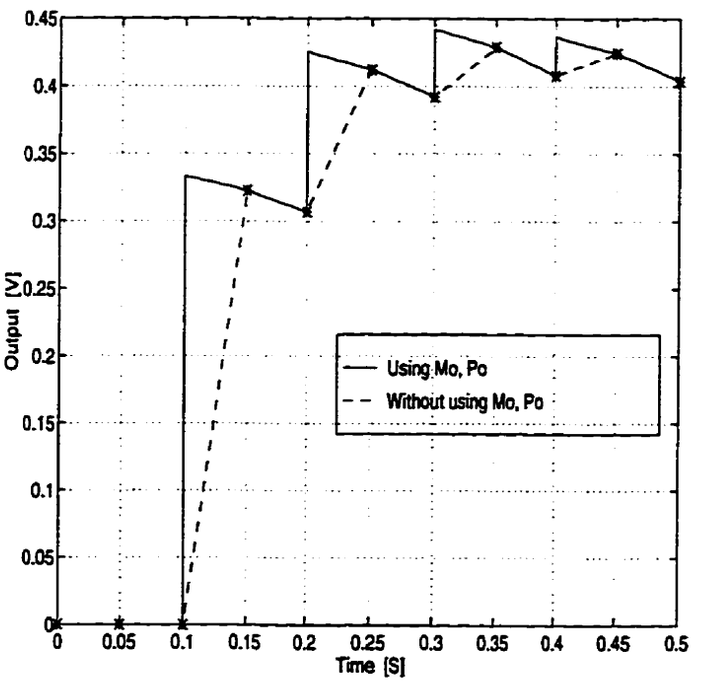


Figure 2.4: Unit step response of the network in Fig.2.3 with and without using $\mathbf{M_0}$ and $\mathbf{P_0}$ matrices.

2.3 Time Domain Sensitivity Using Sensitivity Network

This section presents a new method for computation of the time domain sensitivity of linear networks. It is similar to sampled data simulation in the manner of generating some constant matrices before the simulation, and then providing the sensitivity at each time point by performing only matrix-vector multiplications. The method is accurate because no approximation is made, and efficient because most of the computation is performed only once, in a pre-processing step.

To evaluate the sensitivity of all components of the vector $\mathbf{x}(t)$ with respect to a single parameter h, we start from the MNA formulation (2.1), and differentiate it with respect to h.

$$\frac{\partial \mathbf{G}}{\partial h}\mathbf{x}(t) + \mathbf{G}\frac{\partial \mathbf{x}(t)}{\partial h} + \frac{\partial \mathbf{C}}{\partial h}\frac{d\mathbf{x}(t)}{dt} + \mathbf{C}\frac{\partial}{\partial h}\frac{d\mathbf{x}(t)}{dt} = \mathbf{0}, \qquad \frac{\partial \mathbf{x}(0^-)}{\partial h} = \frac{\partial \mathbf{x}_0}{\partial h} \tag{2.15}$$

We assume that h does not depend on the frequency variable s or the time step T, thus $\frac{\partial \mathbf{w}(t)}{\partial h} = \frac{\partial e^{snT}}{\partial h} = 0$. Define

$$\mathbf{z}(t) = \frac{\partial \mathbf{x}(t)}{\partial h}$$

and substitute it in (2.15)

$$\mathbf{G}\mathbf{z}(t) + \mathbf{C}\frac{d\mathbf{z}(t)}{dt} = -\left[\frac{\partial \mathbf{G}}{\partial h}\mathbf{x}(t) + \frac{\partial \mathbf{C}}{\partial h}\frac{d\mathbf{x}(t)}{dt}\right]. \tag{2.16}$$

In (2.16) there are the differentials of $\mathbf{z}(t)$ and $\mathbf{x}(t)$ with respect to time. Taking the Laplace transform of this equation, and considering the initial conditions of the variables $\mathbf{z}(t)$ and $\mathbf{x}(t)$ give

$$(\mathbf{G} + s\mathbf{C})\,\mathbf{Z} = -\left[\frac{\partial \mathbf{G}}{\partial h} + s\frac{\partial \mathbf{C}}{\partial h}\right]\mathbf{X} + \frac{\partial \mathbf{C}}{\partial h}\mathbf{x}(0^-) + \mathbf{C}\mathbf{z}(0^-),$$

or

$$\mathbf{RZ} = -\frac{\partial \mathbf{R}}{\partial h}\mathbf{X} + \frac{\partial \mathbf{C}}{\partial h}\mathbf{x}(0^-) + \mathbf{C}\mathbf{z}(0^-). \tag{2.17}$$

Eq. (2.17) describes the *sensitivity network* that is the same as the original network but with a different right-hand side. The input of the sensitivity network is composed of the solution of the original network($\mathbf{X}$) and its initial condition ($\mathbf{x}(0^-)$). Substituting $\mathbf{X}$ from (2.3) into (2.17) gives

$$\begin{aligned}
\mathbf{RZ} &= -\frac{\partial \mathbf{R}}{\partial h}\left[\mathbf{R}^{-1}\mathbf{C}\mathbf{x}(0^-) + \mathbf{R}^{-1}\mathbf{W}\right] + \frac{\partial \mathbf{C}}{\partial h}\mathbf{x}(0^-) + \mathbf{C}\mathbf{z}(0^-), \\
\mathbf{RZ} &= -\frac{\partial \mathbf{R}}{\partial h}\mathbf{R}^{-1}\mathbf{W} + \left[-\frac{\partial \mathbf{R}}{\partial h}\mathbf{R}^{-1}\mathbf{C} + \frac{\partial \mathbf{C}}{\partial h}\right]\mathbf{x}(0^-) + \mathbf{C}\mathbf{z}(0^-), \\
\mathbf{Z} &= \underbrace{-\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial h}\mathbf{R}^{-1}\mathbf{W}}_{\text{Provides } P_s} + \underbrace{\left[-\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial h}\mathbf{R}^{-1}\mathbf{C} + \mathbf{R}^{-1}\frac{\partial \mathbf{C}}{\partial h}\right]\mathbf{x}(0^-)}_{\text{Provides } M_s} \\
&\quad + \underbrace{\mathbf{R}^{-1}\mathbf{C}}_{\text{Provides } M}\ \mathbf{z}(0^-).
\end{aligned} \tag{2.18}$$

The first term in (2.18) corresponds to the zero-state response of the sensitivity network, and its integral over the interval $[0, T]$ provides the vector $\mathbf{P_s}$

$$\mathbf{P_s} = \mathcal{L}^{-1}\left(-\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial h}\mathbf{R}^{-1}\mathbf{W}\right)\Bigg|_{t=T}. \tag{2.19}$$

The second and third terms in (2.18) are related to the zero-input response of the sensitivity network, and their integral over the interval $[0, T]$ provide the $\mathbf{M_s}$ and $\mathbf{M}$ matrices

$$\mathbf{M_s} = \mathcal{L}^{-1}\left(-\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial h}\mathbf{R}^{-1}\mathbf{C} + \mathbf{R}^{-1}\frac{\partial \mathbf{C}}{\partial h}\right)\Bigg|_{t=T}, \tag{2.20}$$

$$\mathbf{M} = \mathcal{L}^{-1}\left(\mathbf{R}^{-1}C\right)\Big|_{t=T}. \tag{2.21}$$

In the time domain, (2.18) can be written in sampled data manner as

$$\mathbf{z}(nT+T) = \mathbf{M}_s\mathbf{x}(nT) + \mathbf{M}\mathbf{z}(nT) + \mathbf{P}_s e^{snT} \tag{2.22}$$

Eq.(2.22) is the core algorithm used in the sensitivity analysis described in this section. It can be used to compute the time domain sensitivity in a sampled data manner as long as the $\mathbf{M}_s$ and $\mathbf{P}_s$ matrices are known. The computation of $\mathbf{P}_s$ and $\mathbf{M}_s$, based on (2.19) and (2.20), can be performed simultaneously with the computation of $\mathbf{P}$ and $\mathbf{M}$, based on (2.5) and (2.6). These matrices are constant, and need to be computed only once. It is therefore reasonable to spend extra care in their calculations. The reader is referred to Appendix-A for the algorithm of calculating $\mathbf{M}_s$ and $\mathbf{P}_s$ using the numerical Laplace transform inversion. A program called MPMPSgen, was written in MATLAB to generate these matrices.

The accuracy and cost of performing (2.22) are the same as discussed in [1] for computing (2.7). If the matrices $\mathbf{M}$, $\mathbf{M}_s$, and $\mathbf{P}_s$ are known, then the only operations required in the sensitivity computation given in (2.22) are two matrix-vector multiplications and one vector addition for every sample of T seconds.

To simplify (2.19) and (2.20), we start from the following identity, and differentiate it with respect to h:

$$\begin{aligned} \mathbf{R}\,\mathbf{R}^{-1} &= \mathbf{I} \\ \tfrac{\partial \mathbf{R}}{\partial h}\mathbf{R}^{-1} + \mathbf{R}\tfrac{\partial \mathbf{R}^{-1}}{\partial h} &= \mathbf{0} \\ \tfrac{\partial \mathbf{R}^{-1}}{\partial h} &= -\mathbf{R}^{-1}\tfrac{\partial \mathbf{R}}{\partial h}\mathbf{R}^{-1} \end{aligned} \tag{2.23}$$

Substituting (2.23) into (2.19) provides

$$\mathbf{P}_s = \mathcal{L}^{-1}\left(\frac{\partial \mathbf{R}^{-1}}{\partial h}\mathbf{W}\right)\Bigg|_{t=T} = \mathcal{L}^{-1}\left(\frac{\partial}{\partial h}\left(\mathbf{R}^{-1}\mathbf{W}\right)\right)\Bigg|_{t=T}$$

$$= \frac{\partial}{\partial h}\mathcal{L}^{-1}\left(\mathbf{R}^{-1}\mathbf{W}\right)\Big|_{t=T} = \frac{\partial \mathbf{P}}{\partial h}. \tag{2.24}$$

We took the differential operator $\frac{\partial}{\partial h}$ out of the bracket because h does not depend on the frequency s or the time step T. The source vector $\mathbf{W}$ is also assumed to be independent of h, thus $\frac{\partial \mathbf{W}}{\partial h} = 0$. The matrix $\mathbf{M_s}$ can be similarly simplified by substituting (2.23) into (2.20)

$$\begin{aligned}\mathbf{M_s} &= \mathcal{L}^{-1}\left(\frac{\partial \mathbf{R}^{-1}}{\partial h}\mathbf{C} + \mathbf{R}^{-1}\frac{\partial \mathbf{C}}{\partial h}\right)\Big|_{t=T} = \mathcal{L}^{-1}\left(\frac{\partial}{\partial h}\left(\mathbf{R}^{-1}\mathbf{C}\right)\right)\Big|_{t=T} \\ &= \frac{\partial}{\partial h}\mathcal{L}^{-1}\left(\mathbf{R}^{-1}\mathbf{C}\right)\Big|_{t=T} = \frac{\partial \mathbf{M}}{\partial h}.\end{aligned} \tag{2.25}$$

Considering (2.24) and (2.25) in (2.22) results in

$$\mathbf{z}(nT+T) = \frac{\partial \mathbf{M}}{\partial h}\mathbf{x}(nT) + \mathbf{M}\mathbf{z}(nT) + \frac{\partial \mathbf{P}}{\partial h}e^{snT}. \tag{2.26}$$

It is interesting to note that (2.26) could have been directly derived from (2.7) by differentiating (2.7) with respect to h, and considering $\frac{\partial e^{snT}}{\partial h} = 0$. Of course, we still need to compute the $\mathbf{P_s}$ and $\mathbf{M_s}$ matrices by (2.19) and (2.20).

2.3.1 Example

We illustrate the procedures developed in the previous section with a simple example. In each step, the analytical solutions in the time domain will be given. Of course, for any nontrivial network, solutions of $\mathbf{M_s}$, $\mathbf{P_s}$, and time domain sensitivities must be obtained numerically with the program MPMPSgen, or the algorithms in Appendix-A and the formula presented in the previous section. Consider the RC

circuit shown in Fig.2.5 with the unit step input and zero initial conditions. We calculate the time domain sensitivity of the output $v_2(t)$ with respect to G_1 and C.

First, we find the analytical solutions. The system equation is

$$\begin{bmatrix} G_1 + sC & -sC \\ -sC & G_2 + sC \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{s} \\ 0 \end{bmatrix}, \quad \begin{bmatrix} v_1(0) \\ v_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

The Laplace domain output is

$$V_2(s) = \frac{1}{(G_1 + G_2)\left[s + \frac{G_1 G_2}{C(G_1+G_2)}\right]}.$$

The exact time domain response is

$$v_2(t) = \frac{1}{(G_1 + G_2)} e^{-\frac{G_1 G_2 t}{C(G_1+G_2)}}.$$

The derivative with respect to G_1 is

$$\frac{\partial v_2(t)}{\partial G_1} = \frac{-1}{(G_1 + G_2)^2}\left[1 + \frac{G_2^2 t}{C(G_1 + G_2)}\right] e^{-\frac{G_1 G_2 t}{C(G_1+G_2)}}.$$

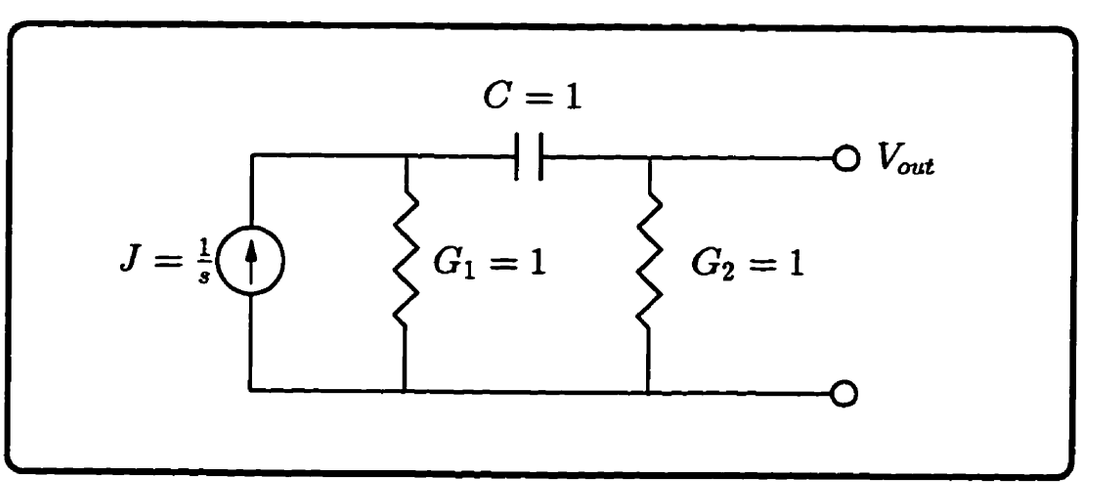


Figure 2.5: RC circuit with unit step input.

The derivative with respect to C is

$$\frac{\partial v_2(t)}{\partial C} = \frac{G_1 G_2 t}{C^2(G_1+G_2)^2} e^{-\frac{G_1 G_2 t}{C(G_1+G_2)}}.$$

For the element values indicated on Fig.2.5, and assuming the step size of $t = T$

$$v_2(nT) = \frac{1}{2}e^{-t/2}\,|_{t=nT} = \frac{1}{2}e^{-nT/2} \tag{2.27}$$

$$\frac{\partial v_2(nT)}{\partial G_1} = -\frac{1}{4}(1+\frac{t}{2})e^{-t/2}\,|_{t=nT} = -\frac{1}{4}(1+\frac{nT}{2})e^{-nT/2} \tag{2.28}$$

$$\frac{\partial v_2(nT)}{\partial C} = \frac{1}{4}te^{-t/2}\,|_{t=nT} = \frac{1}{4}nTe^{-nT/2} \tag{2.29}$$

where $n = 0, 1, \ldots, N$ is the number of time steps.

Next, we compute $v_2(nT)$ and its sensitivities using $\mathbf{M}$, $\mathbf{P}$, $\mathbf{M_s}$, and $\mathbf{P_s}$ matrices and the recurrence equations of (2.7) and (2.22).

$$\mathbf{R} = (\mathbf{G}+s\mathbf{C}) = \begin{bmatrix} 1+s & -s \\ -s & 1+s \end{bmatrix}$$

$$\mathbf{R}^{-1} = \begin{bmatrix} \frac{1+s}{1+2s} & \frac{s}{1+2s} \\ \frac{s}{1+2s} & \frac{1+s}{1+2s} \end{bmatrix}$$

$$\mathbf{M} = \mathcal{L}^{-1}\left(\mathbf{R}^{-1}\mathbf{C}\right)\Big|_{t=T} = \begin{bmatrix} \frac{1}{2}e^{-t/2} & -\frac{1}{2}e^{-t/2} \\ -\frac{1}{2}e^{-t/2} & \frac{1}{2}e^{-t/2} \end{bmatrix}_{t=T}$$

$$= \begin{bmatrix} \frac{1}{2}e^{-T/2} & -\frac{1}{2}e^{-T/2} \\ -\frac{1}{2}e^{-T/2} & \frac{1}{2}e^{-T/2} \end{bmatrix}$$

$$\mathbf{P} = \mathcal{L}^{-1}\left(\mathbf{R}^{-1}\mathbf{W}\right)\Big|_{t=T} = \mathcal{L}^{-1}\begin{bmatrix} \frac{1+s}{(1+2s)s} \\ \frac{1}{1+2s} \end{bmatrix}_{t=T}$$

$$= \begin{bmatrix} -\frac{1}{2}e^{-T/2}+1 \\ \frac{1}{2}e^{-T/2} \end{bmatrix}$$

To get the time domain response of the circuit at discretized points nT, we use (2.7)

$$\begin{bmatrix} v_1(nT+T) \\ v_2(nT+T) \end{bmatrix} = \begin{bmatrix} \frac{1}{2}e^{-T/2} & -\frac{1}{2}e^{-T/2} \\ -\frac{1}{2}e^{-T/2} & \frac{1}{2}e^{-T/2} \end{bmatrix} \begin{bmatrix} v_1(nT) \\ v_2(nT) \end{bmatrix} + \begin{bmatrix} -\frac{1}{2}e^{-T/2}+1 \\ \frac{1}{2}e^{-T/2} \end{bmatrix} \quad (2.30)$$

It is clear that (2.30) generates the same response as indicated in (2.27) at time points $nT, (n = 0, 1, \ldots, N)$.

To find the sensitivity with respect to G_1, we first calculate $\frac{\partial \mathbf{R}}{\partial G_1}$, and then substitute it in (2.19) and (2.20) to obtain the corresponding $\mathbf{P_s}$ and $\mathbf{M_s}$ matrices.

$$\frac{\partial \mathbf{R}}{\partial G_1} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{P_s} = \mathcal{L}^{-1}\left(-\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial G_1}\mathbf{R}^{-1}\mathbf{W}\right)\Big|_{t=T} = \mathcal{L}^{-1}\begin{bmatrix} -\frac{(1+s)^2}{(1+2s)^2 s} \\ -\frac{1+s}{(1+2s)^2} \end{bmatrix}_{t=T}$$

$$= \begin{bmatrix} \frac{1}{8}Te^{-T/2} + \frac{3}{4}e^{-T/2} - 1 \\ \frac{1}{8}Te^{-T/2} - \frac{1}{4}e^{-T/2} \end{bmatrix},$$

$$\mathbf{M_s} = \mathcal{L}^{-1}\left(-\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial G_1}\mathbf{R}^{-1}\mathbf{C}\right)\Big|_{t=T} = \mathcal{L}^{-1}\begin{bmatrix} -\frac{1+s}{(1+2s)^2} & \frac{1+s}{(1+2s)^2} \\ -\frac{1+s}{(1+2s)^2} & \frac{1+s}{(1+2s)^2} \end{bmatrix}_{t=T}$$

$$= \begin{bmatrix} -\frac{1}{8}Te^{-T/2} - \frac{1}{4}e^{-T/2} & \frac{1}{8}Te^{-T/2} + \frac{1}{4}e^{-T/2} \\ \frac{1}{8}Te^{-T/2} - \frac{1}{4}e^{-T/2} & -\frac{1}{8}Te^{-T/2} + \frac{1}{4}e^{-T/2} \end{bmatrix}.$$

To calculate the time domain sensitivity of all nodes with respect to G_1, rewrite (2.22) as follows

$$\begin{bmatrix} \frac{\partial v_1(nT+T)}{\partial G_1} \\ \frac{\partial v_2(nT+T)}{\partial G_1} \end{bmatrix} = \begin{bmatrix} -\frac{1}{8}Te^{-T/2} - \frac{1}{4}e^{-T/2} & \frac{1}{8}Te^{-T/2} + \frac{1}{4}e^{-T/2} \\ \frac{1}{8}Te^{-T/2} - \frac{1}{4}e^{-T/2} & -\frac{1}{8}Te^{-T/2} + \frac{1}{4}e^{-T/2} \end{bmatrix} \begin{bmatrix} v_1(nT) \\ v_2(nT) \end{bmatrix}$$

$$+ \begin{bmatrix} \frac{1}{2}e^{-T/2} & -\frac{1}{2}e^{-T/2} \\ -\frac{1}{2}e^{-T/2} & \frac{1}{2}e^{-T/2} \end{bmatrix} \begin{bmatrix} \frac{\partial v_1(nT)}{\partial G_1} \\ \frac{\partial v_2(nT)}{\partial G_1} \end{bmatrix} +$$

$$\begin{bmatrix} \frac{1}{8}Te^{-T/2} + \frac{3}{4}e^{-T/2} - 1 \\ \frac{1}{8}Te^{-T/2} - \frac{1}{4}e^{-T/2} \end{bmatrix}. \tag{2.31}$$

Assuming zero initial conditions for the circuit in Fig. 2.5 and for its sensitivity network, Eq.(2.31) generates the same results as that of (2.28) at discretized time points $nT, (n = 0, 1, \ldots, N)$.

Without going into the details, and following the same procedure, we find the following equations for the time domain sensitivity of the network with respect to C.

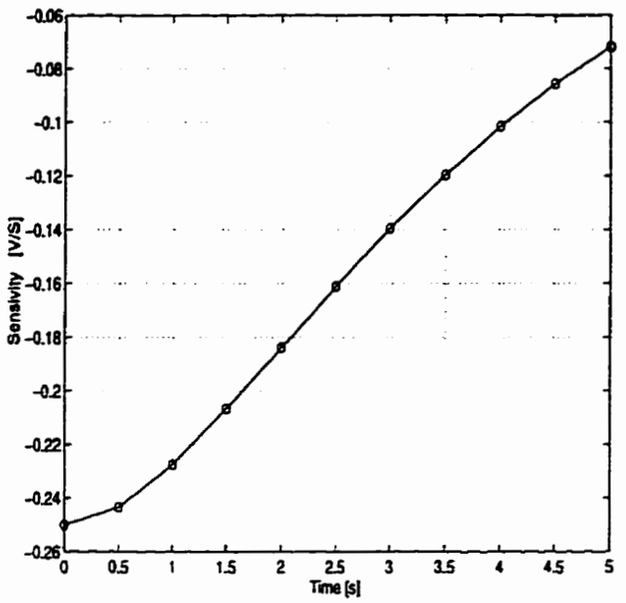
$$\frac{\partial \mathbf{R}}{\partial \mathbf{C}} = \begin{bmatrix} s & -s \\ -s & s \end{bmatrix}$$

$$\mathbf{P_s} = \begin{bmatrix} -\frac{1}{4}Te^{-T/2} \\ \frac{1}{4}Te^{-T/2} \end{bmatrix}$$

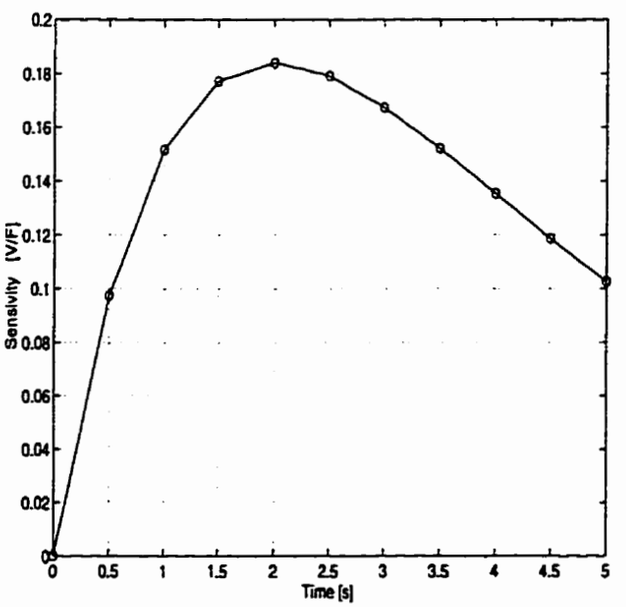
$$\mathbf{M_s} = \begin{bmatrix} -\frac{1}{4}Te^{-T/2} & -\frac{1}{4}Te^{-T/2} \\ -\frac{1}{4}Te^{-T/2} & \frac{1}{4}Te^{-T/2} \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial v_1(nT+T)}{\partial C} \\ \frac{\partial v_2(nT+T)}{\partial C} \end{bmatrix} = \begin{bmatrix} -\frac{1}{4}Te^{-T/2} & -\frac{1}{4}Te^{-T/2} \\ -\frac{1}{4}Te^{-T/2} & \frac{1}{4}Te^{-T/2} \end{bmatrix} \begin{bmatrix} v_1(nT) \\ v_2(nT) \end{bmatrix}$$
$$+ \begin{bmatrix} \frac{1}{2}e^{-T/2} & -\frac{1}{2}e^{-T/2} \\ -\frac{1}{2}e^{-T/2} & \frac{1}{2}e^{-T/2} \end{bmatrix} \begin{bmatrix} \frac{\partial v_1(nT)}{\partial C} \\ \frac{\partial v_2(nT)}{\partial C} \end{bmatrix} + \begin{bmatrix} -\frac{1}{4}Te^{-T/2} \\ \frac{1}{4}Te^{-T/2} \end{bmatrix} \tag{2.32}$$

Again, assuming zero initial conditions for the original and the sensitivity networks, Eq.(2.32) generates exactly the same results as (2.29) at time points nT. Fig. 2.6 shows the simulation results for the sensitivity of the output with respect to G_1 and C at a few time points.



(a)



(b)

Figure 2.6: Simulation results for the sensitivity of $v_2(t)$ (a) with respect to G_1, and (b) with respect to C.

2.4 Time Domain Sensitivity Using Adjoint Network

The time domain sensitivity of *one* variable, which is usually the single output ϕ, with respect to *many* variable elements h_i at *one* instant of time ($t = t_f$) is considered in this section. The application of sensitivity at one instant of time is in the calculation of error gradients for linear networks, and in the time domain sensitivity analysis of objective functions used in circuit optimization [3,5,7]. Another application is in the time domain sensitivity analysis of switched networks at the end of each phase when the circuit does not reach the steady state. Transient analysis of rise/fall time in digital circuits and transmission lines are the other types of applications.

In the frequency domain, the application of adjoint network (sometimes called transpose network) in the calculation of sensitivity is well known [3,5]. The procedure for computing frequency domain sensitivity using an adjoint network is summarized as follows:

I- Solve the given system of linear equations $\mathbf{RX} = \mathbf{W}$.

II- Solve the adjoint network defined by $\mathbf{R}^t\mathbf{X}^a = -\mathbf{d}$ where $\mathbf{d}$ is a constant vector that relates the output ϕ to the nodal vector $\mathbf{X}$, $\phi = \mathbf{d}^t\,\mathbf{X}$.

III- For each parameter h_i, form $\frac{\partial \mathbf{R}}{\partial h_i}$, and insert this in the following equation to compute $\frac{\partial \phi}{\partial h_i}$

$$\frac{\partial \phi}{\partial h_i} = (\mathbf{X}^a)^t \frac{\partial \mathbf{R}}{\partial h_i}\mathbf{X} \tag{2.33}$$

As the vector $\mathbf{X}$ and $\mathbf{X}^a$ are independent of the parameters h_i, Eq.(2.33) requires the solution of only two sets of algebraic equations (in steps I and II), irrespective of the number of parameters h_i.

In the time domain, the sensitivity is computed by the procedure explained in [7] as follows:

1- Perform a transient analysis of the original network N for the time interval $t = [0, t_f]$. Obtain $i(t)$ or $v(t)$ for resistive , $\dot{v}(t)$ for capacitive, and $\dot{i}(t)$ for inductive branches.

2- Construct the adjoint network $\widehat{N}$ according to the fact that the adjoint network of R, L, and C are unchanged, and the adjoint network of dependent sources are dependent sources with a new configuration and transfer function given in [7]. Set all initial conditions and all independent sources equal to zero, and apply a current source of $-\delta(t)$ between the output nodes.

3- Perform a transient analysis of the adjoint network $\widehat{N}$ for the time interval $\tau = [0, t_f]$, where $\tau = t_f - t$ (time reversal). Obtain $\hat{i}(\tau)$ or $\hat{v}(\tau)$ for resistive branches, $\hat{v}(\tau)$ for capacitive, and $\hat{i}(\tau)$ for inductive branches.

4- Evaluate the following equations to find the sensitivity of the output with respect to R, and C (for other elements refer to [7]).

$$\frac{\partial v_{out}(t_f)}{\partial R} = \int_0^{t_f} \left[\hat{i}_R(\tau) i_R(t)\right]_{\tau=t_f-t} dt$$

$$\frac{\partial v_{out}(t_f)}{\partial C} = -\int_0^{t_f} \left[\hat{v}_C(\tau) \dot{v}_C(t)\right]_{\tau=t_f-t} dt$$

2.4.1 Adjoint Network in Time Versus Frequency Domain

In the time domain, the adjoint network is obtained from a more general principle known as Tellegen's theorem [7,21,22], whereas in the frequency domain, the adjoint network can be obtained after some matrix manipulations [3]. The applications of the adjoint network in these two domains were derived separately without making any explicit link between them. We establish here a one-by-one correspondence between these two applications as shown in Table 2.2. Based on the relations shown in this table, the adjoint network can be established in either the time or frequency domain, and finally, can be transformed to the other one, if necessary. Considering

Frequency Domain	Time Domain
step I: Solving the original system	step 1: Solving the original system
step II : Creating the transpose system of $\mathbf{R}^t\mathbf{X}^a = -\mathbf{d}$	step 2 : Constructing the adjoint network $\widehat{N}$ and applying a source of $-\delta(t)$
step III : Multiplication in the frequency domain	step 3 : The solution of the transpose system **reversed** in time, needed for the convolution involved in step 4 step 4 : Convolution of $i_R(t)$ and $\hat{i}_R(t)$ for resistive elements,and convolution of $\dot{v}_C(t)$ and $\hat{v}_C(t)$ for capacitive elements
step III : If the parameter h is a frequency dependent element, $\frac{\partial \mathbf{R}}{\partial h}$ produces a s	step 4 : If the parameter h is a frequency dependent element, use the derivative of the voltage, $\dot{v}(t)$

Table 2.2: Relationship between the adjoint methods in time and frequency domains.

this, and by applying sampled data simulation, we introduce two approaches for time domain sensitivity of linear networks using an adjoint network.

2.4.2 Approach I : Convolution

The formulation of the adjoint network in the frequency domain is straightforward. It performs a multiplication in the frequency domain (step III) which can be translated to a convolution in the time domain. Since the sampled data simulation is an accurate and efficient method for computing the circuit response in the time domain, we transform the frequency domain sensitivity analysis into the time domain as follows :

1- Solve the original system of equations $\mathbf{G}\mathbf{x}(t)+\mathbf{C}\frac{d\mathbf{x}(t)}{dt} = \mathbf{w}(t)$ in the time domain using $\mathbf{P}$ and $\mathbf{M}$ matrices

$$\mathbf{x}(nT+T) = \mathbf{M}\mathbf{x}(nT) + \mathbf{P}e^{snT} \quad , \quad n = 0, 1, \ldots, N$$

where $\mathbf{P}$ and $\mathbf{M}$ are given by (2.5) and (2.6). This step is equivalent to solving $\mathbf{R}\mathbf{X} = \mathbf{W}$ in frequency domain.

2- Solve the system of equations $\mathbf{G}\mathbf{y}(t) + \mathbf{C}\frac{d\mathbf{y}(t)}{dt} = \frac{d\mathbf{w}(t)}{dt}$ in the time domain to get the solution of $\mathbf{y}(t) = \frac{d\mathbf{x}(t)}{dt} = \dot{\mathbf{x}}(t)$

$$\dot{\mathbf{x}}(nT+T) = \mathbf{M}\dot{\mathbf{x}}(nT) + \dot{\mathbf{P}}\ e^{snT} \quad , \quad n = 0, 1, \ldots, N$$

where $\mathbf{M}$ is the same as in the previous step, and

$$\dot{\mathbf{P}} = \mathcal{L}^{-1}\left(s\mathbf{R}^{-1}\mathbf{W}\right)\Big|_{t=T}$$

We need $\dot{\mathbf{x}}(t)$ to compute the sensitivity with respect to frequency dependent elements like capacitors and inductors.

3- Solve the transpose system of $\mathbf{G}^t\mathbf{x}^a(t) + \mathbf{C}^t\frac{d\mathbf{x}^a(t)}{dt} = -\mathbf{d}\delta(t)$ in the time domain which is equivalent to the solution of the transpose system $\mathbf{R}^t\mathbf{X}^a = -\mathbf{d}$ in the frequency domain.

$$\mathbf{x}^a(nT+T) = \mathbf{M}^t\mathbf{x}^a(nT) + \mathbf{P}^a g(nT) \quad , \quad n = 0, 1, \ldots, N$$

where, because of the Dirac impulse input,

$$g(nT) \;=\; \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{otherwise} \end{cases}$$

and $\mathbf{M}^t$ and $\mathbf{P}^a$ are given by

$$\mathbf{M}^t \;=\; \mathcal{L}^{-1}\left((\mathbf{R}^t)^{-1}\right)\Big|_{t=T} = \mathcal{L}^{-1}\left((\mathbf{R}^{-1})^t\right)\Big|_{t=T} = \text{transpose of } M$$

$$\mathbf{P}^a \;=\; \mathcal{L}^{-1}\left(-(\mathbf{R}^t)^{-1}\mathbf{d}\right)\Big|_{t=T}$$

4- Evaluate the equation $F(s) = \frac{\partial\phi}{\partial h_i} = (\mathbf{X}^a)^t\frac{\partial\mathbf{R}}{\partial h_i}\mathbf{X}$ in the time domain using discrete-time convolution. For instance, if h_i is a conductance appearing at column #1 and row #1 of $\mathbf{R}$

$$\begin{aligned} F(s) &= X_1^a\, X_1 \\ f(NT) &= x_1^a(NT) \oplus x_1(NT) = \sum_{m=0}^{N} x_1^a(mT)x_1(NT-mT). \end{aligned}$$

If h_i is a capacitor appearing at column #1 and row #1 of $\mathbf{R}$

$$\begin{aligned} F(s) &= X_1^a\, s\, X_1 \\ f(NT) &= x_1^a(NT) \oplus \dot{x}_1(NT) = \sum_{m=0}^{N} x_1^a(mT)\dot{x}_1(NT-mT). \end{aligned}$$

Since we are interested in the sensitivity at one instant in time, the above convolutions are evaluated at only that time point (NT). The computation cost depends on N. If we do not consider the steady state, and concentrate only on the transient response of the circuit and its sensitivity, N is normally a small number. In switched networks, if one is interested in the time domain analysis within each phase, N could be less than 10.

The drawback of this approach is that we are convolving the samples of two signals x_1 and x_1^a to generate the time domain equivalent of the multiplication $X_1^a(s)\ X_1(s)$ in the Laplace domain. This is accurate only if the original analog signals are sampled at a rate that is twice the maximum frequency component of the signals. Otherwise, some errors are encountered in the discrete-time convolution due to sampling. Another minor drawback may be the need for memory storage for the circuit response from 0 to NT for the purpose of convolution.

2.4.3 Approach II : Numerical Laplace Transform Inversion

This approach is basically a translation of the adjoint method from the frequency domain into the time domain. Assume $\mathbf{X}$ to be the solution of the original network in the frequency domain

$$\mathbf{R}\,\mathbf{X} = \mathbf{W},$$

and $\mathbf{X}^a$ to be the solution of the adjoint network

$$\mathbf{R}^t\,\mathbf{X}^a = -\mathbf{d},$$

where $\mathbf{d}$ is a selector vector defining the output ϕ

$$\phi = \mathbf{d}^t\,\mathbf{X}.$$

Also assume m to be the size of the system matrix $\mathbf{R}$, and l the number of parameters h_i with respect to which the sensitivities are computed. Define matrix $\mathbf{Q}$ as

$$\mathbf{Q} = \begin{bmatrix} (\mathbf{X}^a)^t \frac{\partial \mathbf{R}}{\partial h_1} \\ (\mathbf{X}^a)^t \frac{\partial \mathbf{R}}{\partial h_2} \\ \vdots \\ (\mathbf{X}^a)^t \frac{\partial \mathbf{R}}{\partial h_l} \end{bmatrix}_{l \times m},$$

Since we are interested in the sensitivities at only one time point, $t = t_f$, they can be computed using the numerical Laplace inversion

$$\mathbf{P}_{Adj} = \mathcal{L}^{-1}(\mathbf{Q}\ \mathbf{X})\Big|_{t=t_f}$$

The numerical Laplace inversion has to be performed separately for each row of $\mathbf{Q}\ \mathbf{X}$. Each entry i in $\mathbf{P}_{Adj}$ now contains the time domain sensitivity of the output ϕ with respect to each parameter h_i at a fixed time point $t = t_f$.

Comparing to Approach I, the above proceeds continuously in time, and does not include discrete time convolution. In addition, all computations are performed in the forward direction in time. This may reduce the memory space requirement because there is no need to store the whole response of the circuit (or its adjoint) from 0 to t_f.

Chapter 3

Analysis and Sensitivity of Delta-Sigma Modulator

An obvious application of the method developed in chapter 2 is in the sensitivity analysis of Delta-Sigma Modulators (DSM). Due to the presence of a comparator, which is a nonlinear element, the sensitivity analysis of DSM can not be performed directly in the frequency domain. Instead, we apply the method of chapter 2 (sensitivity network) to do a fast sensitivity analysis in the time domain, and finally transfer the results into the frequency domain by means of the FFT.

A brief review of DSM is given in section 1. The reader is referred to [23–28] for more details. Analysis of a second-order DSM using sampled data simulation is discussed in section 2. My contribution to this topic, the sensitivity analysis of a DSM, is presented in section 3, where I derive the sensitivity network for the DSM, and distinguish between unclocked and clocked comparators to explain the sensitivity of the comparator output with respect to its input. A discussion about the incremental and differential sensitivity is also given in section 3. The

derivations, although explained on a specific circuit, are general and can be applied to all types/configurations of modulators if all elements, except the comparator, are linear.

3.1 Oversampled Delta-Sigma A/D Converters

With the emergence of digital signal processing applications, there is an increasing demand for high resolution on-chip A/D converters. Conventional A/D converters, which sample and quantize the input signal at the Nyquist rate, have attributes that make it difficult to implement in fine-line VLSI technology. They need sharp cutoff analog filters, high precision analog components, and they suffer from increased noise levels due to high circuit densities. Oversampling A/D converters, on the other hand, depend on relatively simple and modest analog circuitry. They combine high sampling rates with negative feedback in order to trade off resolution in time for resolution in amplitude. Oversampled A/D converters are insensitive to circuit imperfections and component mismatch, since they usually employ a simple two-level quantizer embedded within a feedback loop [26, 27].

The basic structure of the oversampled Delta-Sigma A/D converter consists of four blocks: the input anti-aliasing analog filter, the Delta-Sigma modulator, the decimator, and a digital low pass filter. The input anti-aliasing filter is a non-critical low-order passive filter whose cutoff frequency is set at some frequency far above the Nyquist rate. The Delta-Sigma modulator performs two important functions: one is to modulate the band-limited analog input signal into a one-bit digital code at a frequency much higher than the Nyquist rate. The other function is to noise-shape the quantization noise and transfer most of its energy to high frequencies. The decimator converts the low resolution high bit rate signal to the high resolution low

bit rate signal. Finally, the digital filter removes out-of-band quantization noise. The decimator and the digital filter are usually combined into a single digital circuit, where the digital filter comes first to suppress the high frequency noise before doing decimation.

A second-order modulator is shown in Fig. 3.1-(a). The input analog signal x(t) is sampled at the sampling frequency, f_s, much higher than the signal Nyquist rate. The ratio of the sampling frequency to Nyquist rate is called the oversampling ratio M

$$M = f_s/f_N.$$

A quantizer with only two levels is employed to avoid the distortion generated by step-size mismatch in multibit quantizers. The integrators force the average of the error signal $e(t)$ to be zero. This error signal results from subtraction of the output signal from the input.

To analyze the operation of the second-order SDM,we assume that the quantization noise is uncorrelated with the input signal. In such a case, the modulator can be linearized as shown in Fig. 3.1-(b). In this model, each integrator is replaced by a z-domain Forward Euler non-inverting integrator, Q(z) is the additive quantization noise, and the delay cell in the feedback path represents the latch in the output of the quantizer. In this model,

$$Y(z) = F_W(z)W(z) + F_Q(z)Q(z)$$

where $F_W(z)$ is the signal transfer function, and $F_Q(z)$ is the noise transfer function. If the gains of the integrators are K_1 and K_2, and the gain of DAC is K_3, the following expressions can be derived for $F_W(z)$ and $F_Q(z)$:

$$F_W(z) = \frac{K_1K_2H^2(z)}{K_1K_2K_3z^{-1}H^2(z) + K_2K_3z^{-1}H(z) + 1}$$

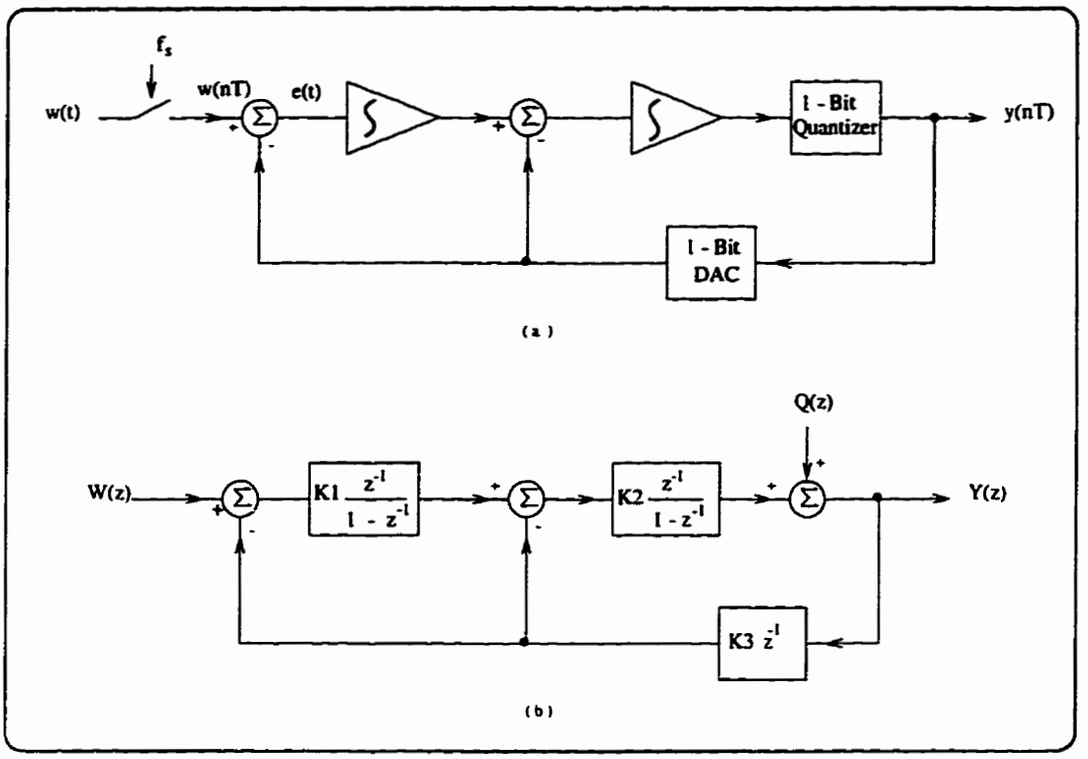


Figure 3.1: (a) Block diagram of a second-order Delta-Sigma modulator, (b) Linearized model of modulator.

$$F_Q(z) = \frac{1}{K_1K_2K_3z^{-1}H^2(z) + K_2K_3z^{-1}H(z) + 1}$$

where $H(z) = \frac{z^{-1}}{1-z^{-1}}$. In practice, $F_W(z)$ is a lowpass filter with unity gain in the passband. In contrast, $F_Q(z)$ has a highpass behavior, effectively attenuating the quantization noise in the modulator passband at the expense of amplifying the quantization noise at higher frequencies.

The maximum in-band signal-to-noise ratio (SNR_{max}) of an ideal second-order DSM is approximately expressed by [28] :

$$SNR_{max} = 15Log_2(M) - 13 \quad (dB) \tag{3.1}$$

where M is the oversampling ratio. For an input signal of 10kHz (f_N is 20kHz),

and with an oversampling ratio of 128 ($f_S = 2.56MHz$), the SNR is 92dB, equivalent to 15 bits of resolution. In theory, the SNR can be increased without limit by increasing the oversampling ratio. However, the higher sampling frequency is limited in practice by the circuit frequency response. In addition, the total noise is the summation of the in-band quantization noise and in-band noise coming from other error sources, such as thermal noise, flicker noise, and clock feed through. The non-linearity of the circuit also limits the signal dynamic range, thus limiting the SNR.

3.2 Sampled Data Simulation of Delta-Sigma Modulator

As explained in [1], sampled data simulation, although developed for linear circuits, can be applied to a restricted set of nonlinear elements whose characteristics change only at switching instants. The change in characteristics could include a change of the value (resistance, capacitance etc.), or a change of topology. Examples of such nonlinear elements are single and multibit quantizers that are used as analog to digital converters in oversampled DSMs. These circuits are externally clocked, and their output changes only at the switching instants based on their input at that time.

A second-order continuous-time oversampled DSM [29] is shown in Fig. 3.2. It is partitioned into the linear and nonlinear blocks. Sampled data simulation proceeds by calculating the output of the linear block after one clock cycle, updating the state of the comparator based on the output of the linear block and then repeating the process for the next clock cycle. The output of the linear block is needed only

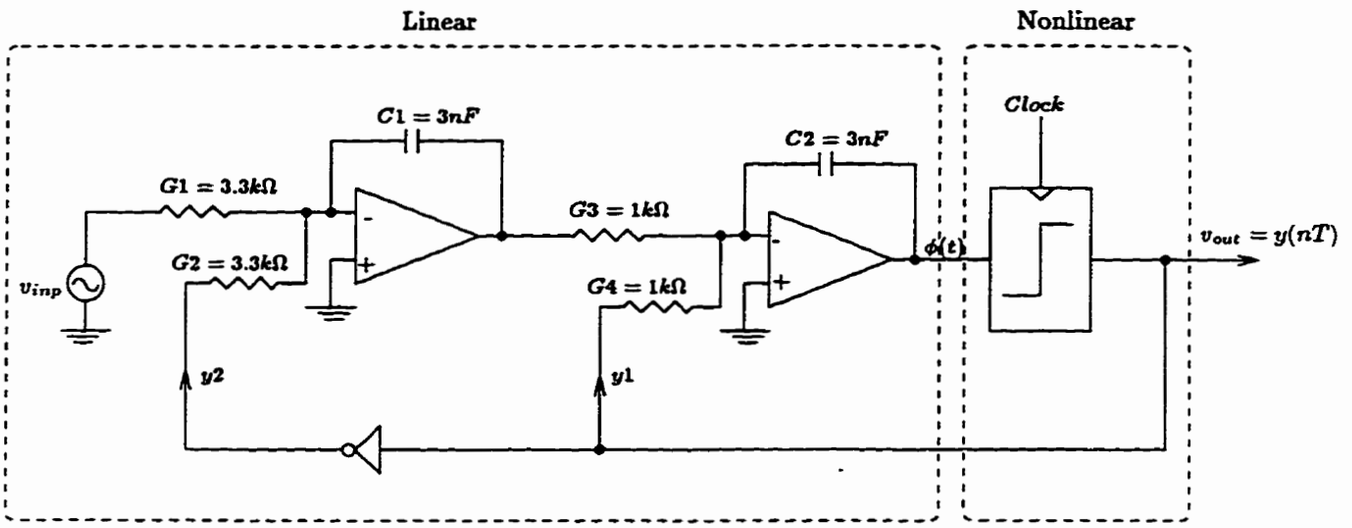


Figure 3.2: Second-order continuous-time Delta-Sigma modulator

at the clock edges when the quantizer samples its input. In between the clock edges, the output of the linear block changes, but does not affect the operation of the overall circuit. Since the output of the linear block is needed only at the clock instants, sampled data simulation is used. During simulation we have two types of inputs to the linear block: the primary input, which is the signal to be converted, and the feedback from the quantizer. For theoretical studies, the primary input is considered to be a sinusoidal function. The feedback signals are constant over a clock cycle and can be treated as a step input for the clock duration.

The MNA formulation of the linear block in Fig.3.2 provides

$$\mathbf{R}\,\mathbf{X} = \mathbf{W}, \text{ where } \quad \mathbf{R} = \mathbf{G} + s\mathbf{C}.$$

The linear block is stimulated by two inputs: the primary input $A_0 cos(\omega_0 t)$, and the feedback from quantizer (which is a step function). These inputs are considered in separate source vectors $\mathbf{W}_{inp}$, and $\mathbf{W}_1$. The $\mathbf{M}$ and $\mathbf{P}$ matrices needed for sampled

data simulation are

$$\mathbf{M} = \mathcal{L}^{-1}\left(\mathbf{R}^{-1}\,\mathbf{C}\right)\Big|_{t=T},$$

$$\mathbf{P}_{inp} = \mathcal{L}^{-1}\left(\mathbf{R}^{-1}\mathbf{W}_{inp}\right)\Big|_{t=T},$$

$$\mathbf{P}_{y1} = \mathcal{L}^{-1}\left(\mathbf{R}^{-1}\mathbf{W}_{1}\right)\Big|_{t=T},$$

where T is the sampling period. The linear block operates on the primary input and the feedback signals, and provides the output $\phi(nT)$ to be applied to the comparator. The output of the comparator $y(nT)$ changes only at the rising (or falling) edges of the external clock, and remains at the same level until the next edge of the clock.

$$y(nT) = \text{sign}\left(\phi(nT)\right) = \begin{cases} 1 & \text{if} \phi(nT) \geq 0 \\ -1 & \text{if} \phi(nT) < 0 \end{cases}$$

The following sampled data equation determines the time domain response of the linear block

$$\mathbf{x}(nT+T) = \mathbf{M}\mathbf{x}(nT) + Real\left(\mathbf{P}_{inp}\, e^{j\omega_0 nT}\right) + \mathbf{P}_{y1}\, y(nT). \tag{3.2}$$

After each clock cycle, the state of the comparator is updated based on the output of the linear block. The comparator then generates the feedback signals, and the process is repeated for the next clock cycle.

The Delta-Sigma circuit in Fig. 3.2 was simulated considering a clock frequency of 1MHz with equal phase widths, and the input frequency of 1kHz with 0.6V peak to peak amplitude. It took 3 minutes CPU execution time on a SPARC-10 Sun workstation to simulate the modulator for 74k clock cycles. The spectrum at

the output of the modulator is shown in Fig. 3.3. The results were obtained by simulating the circuit for 74k clock cycles, discarding the first 10k data points to remove any circuit transients, and performing a Fast Fourier Transform(FFT) on the remaining 64k data points. Discarding the first 10K data points ensures that all the transients are passed, because it is equivalent to ignoring the circuit response from 0 to 10mSec while the circuit time constants are around μSec.

For comparison, the circuit of Fig.3.2 was also simulated using HSPICE. To make the circuit similar to what was simulated using sampled data simulation, all elements were considered ideal, and defined by behavioral models. The latch circuit

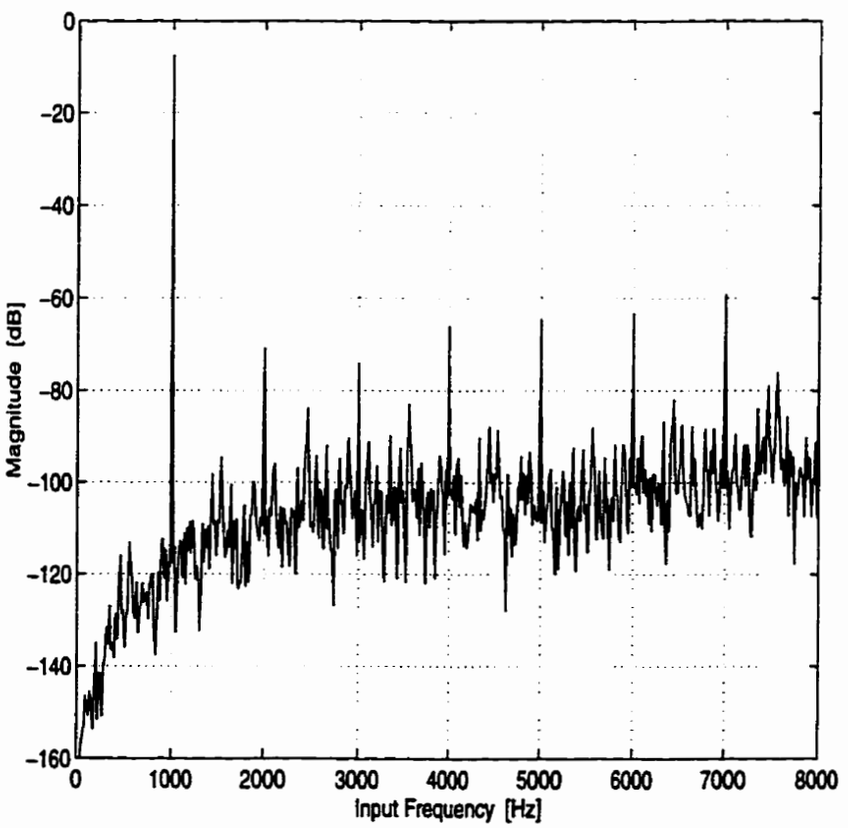


Figure 3.3: DSM output spectrum for a 1 kHz sinusoidal input.

in the quantizer was also designed using a few ideal switches. It took 14 hours CPU execution time on the same machine to simulate the modulator for 74k clock cycles.

The complete SNR curve vs. input amplitude is shown in Fig. 3.4 for 20 different input amplitudes. A 4 kHz bandwidth was assumed for SNR calculations. This SNR curve needs 20 simulations of the Delta-Sigma modulator one for each input amplitude. Each simulation is for 74k clock cycles. This is an expensive task for general purpose simulators like SPICE [2, 9]. Since sampled data simulation is a very fast method, it can provide the SNR curve in less than 45 minutes CPU execution time on a SPARC-10 Sun workstation.

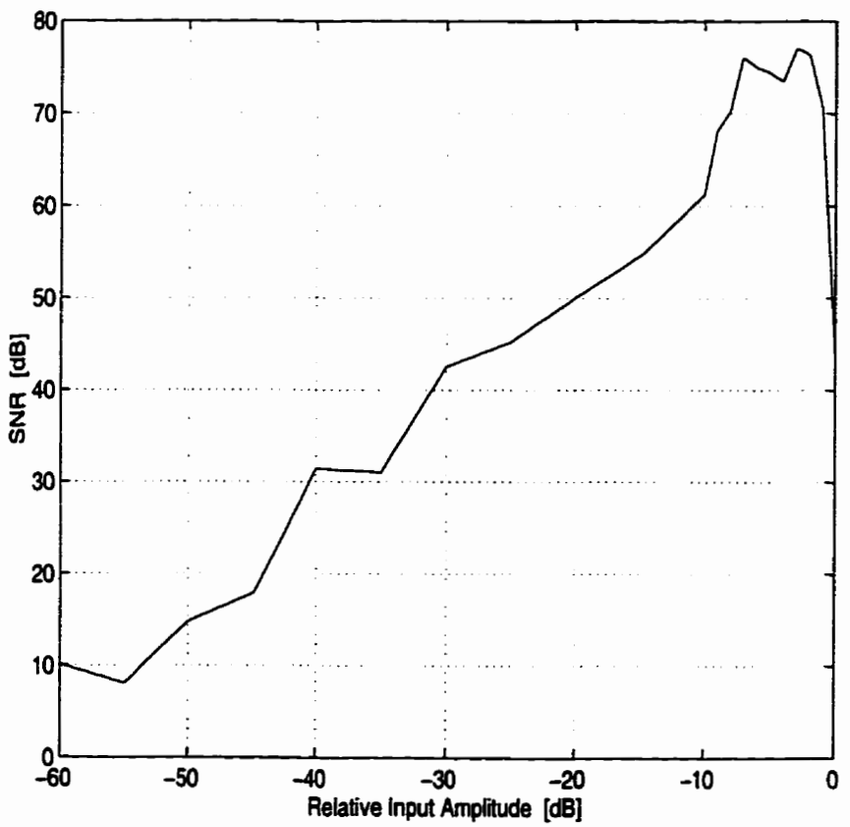


Figure 3.4: Signal to noise ratio vs. input amplitude for the DSM in Fig. 3.2.

3.3 Sensitivity of Delta-Sigma Modulators

Sampled data simulation provides a fast and accurate method of calculating the SNR curve for a Delta-Sigma modulator when all elements are linear except the comparator. The method allows us to repeat the simulation of the circuit several times, each time with an increment in one of the element values. The new circuit obtained from changing one of the element values in the original circuit is called the perturbed circuit. By simulating several perturbed circuits, we can investigate the effect of the change in element values on the output magnitude. This type of analysis is called *incremental sensitivity analysis*. It is in contrast with *differential sensitivity analysis* in which we take the differential of the circuit equations, and solve it simultaneously with the circuit equations without perturbing the circuit itself.

As an example of incremental sensitivity, we simulate the circuit in Fig. 3.2 for different values of G1. The value of G1 is changed form 0.3 mS down to 0.1 mS, and the simulation is repeated from the beginning for each value. After taking the FFT of the output, the magnitude of the tone at the input frequency is plotted versus G1 in Fig 3.5. The slope of the curve is the sensitivity, i.e. $\frac{\partial \mid V_{out} \mid}{\partial G1}$. Considering nominal values of $\mid V_{out} \mid = 0.6$ and $G1 = \frac{1}{3.3k\Omega}$, the normalized sensitivity is about 1.

$$S_{G1}^{|V_{out}|} = \frac{G1}{\mid V_{out} \mid} \frac{\partial \mid V_{out} \mid}{\partial G1} \approx 1$$

This result can be verified by looking at the modulator circuit in Fig. 3.2. Since the opamps are ideal, G1 is connected to the virtual ground at one end, and the input signal at the other. Any change in G1 is converted to a proportional change in the current through G1, which is integrated by the first integrator. The current through G1 can be also controlled by the input signal. Therefore, instead

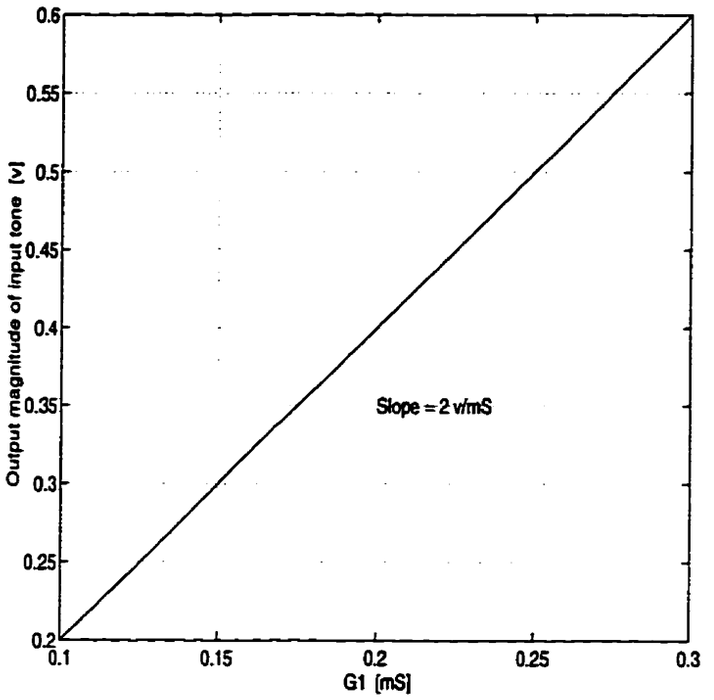


Figure 3.5: The DSM output magnitude of input tone for different values of G1.

of changing G1, we may change the amplitude of the input signal. With this interpretation, the normalized sensitivity of the modulator output with respect to G1 is equivalent to the normalized sensitivity of the output with respect to the input, which is 1 if the opamps do not saturate and the modulator remains in its linear operating region.

After understanding that the normalized sensitivity of the output with respect to G1 is about 1, let us make a very small change in G1. This is equivalent to holding G1 fixed but making a very small change in the amplitude of the input signal. If this change is smaller than the resolution of the Delta-Sigma modulator, its output on average (after taking FFT and considering the magnitude of the signal at the

input frequency) does not change. This produces the same output (on average) as in the nominal case. This can be seen in Fig. 3.6 which shows a magnified area of Fig. 3.5 (assuming that the input magnitude is changed instead of G1). In fact, the curve in Fig. 3.5 is not a straight line. Some parts have the slope of zero resulting in a sensitivity of zero, and some other parts have larger slopes. But overall, the curve has a slope that results in a normalized sensitivity of 1.

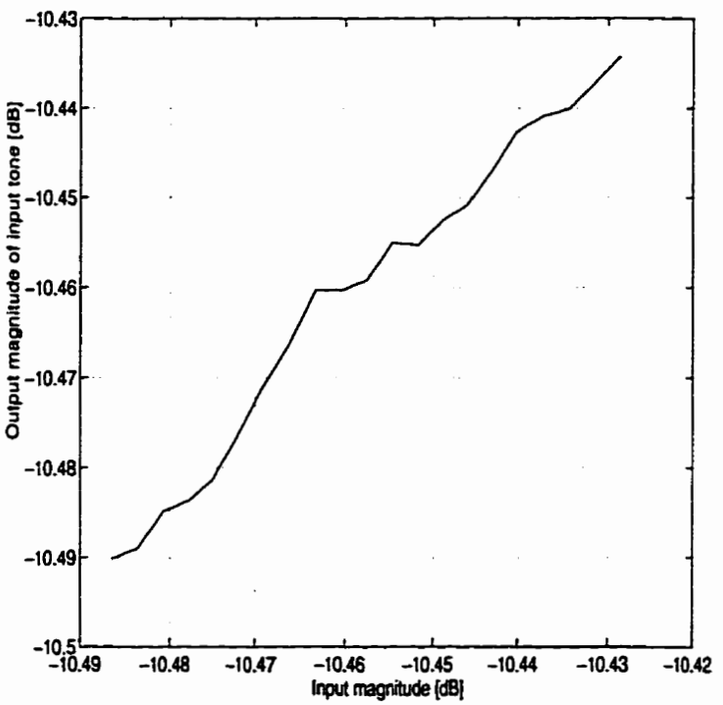


Figure 3.6: The magnified area of the curve in Fig. 3.5 (assuming that the input magnitude is changed instead of G1).

3.3.1 Sensitivity Network

Changing the element values in the original circuit and repeating the simulation of the perturbed circuit from the beginning is not usually the best method for the sensitivity analysis. Instead, a set of equations can be derived and simultaneously solved with the original circuit equations. For instance, in chapter 2 we explained how to establish a sensitivity network from an original network, and solve them in parallel to get the sensitivity of all nodes with respect to an element value. The system matrix of the sensitivity network is the same as that of the original network. It has only a different right-hand side. The question that arises here is "Is there any possibility to obtain the sensitivity of a Delta-Sigma modulator using the sensitivity network, without perturbing the original circuit and repeating the simulation from the beginning?" In this section we try to answer this question.

Due to the presence of the comparator, which is a nonlinear element, the sensitivity analysis of Delta-Sigma modulators can not be performed directly in the frequency domain. Instead, we do the analysis in the time domain, and finally transform the results into the frequency domain by means of the FFT.

Taking the differential of (3.2) with respect to element h gives

$$\begin{aligned}\frac{\partial \mathbf{x}(nT+T)}{\partial h} &= \frac{\partial \mathbf{M}}{\partial h}\mathbf{x}(nT) + \mathbf{M}\frac{\partial \mathbf{x}(nT)}{\partial h} + Real\left(\frac{\partial \mathbf{P}_{inp}}{\partial h}e^{j\omega_0 nT}\right) \\ &+ \frac{\partial \mathbf{P}_{yl}}{\partial h}y(nT) + \mathbf{P}_{yl}\frac{\partial y(nT)}{\partial h} \end{aligned} \tag{3.3}$$

Since h does not depend on the frequency ω_0 or the time period T, $\frac{\partial}{\partial h}e^{j\omega_0 nT} = 0$. Eq.(3.3) can be used to compute the time domain sensitivity in a sampled data manner. The matrices $\frac{\partial \mathbf{M}}{\partial h}$, $\frac{\partial \mathbf{P}_{inp}}{\partial h}$, and $\frac{\partial \mathbf{P}_{yl}}{\partial h}$ are constant, and computed once before simulation starts. These matrices can be computed simultaneously with $\mathbf{M}$ and $\mathbf{P}$ as explained in Appendix-A.

The only term in (3.3) which needs more investigation is $\frac{\partial y(nT)}{\partial h}$, the sensitivity of the output of the comparator. For more explanations, we distinguish between unclocked and clocked comparators.

Sensitivity of The Unclocked Comparator

Assume $y(t)$ to be the output of the unclocked comparator in Fig. 3.7. $y(t)$ represents the sign of $\phi(t)$, and can be written as any one of the following equations

$$\begin{aligned} y(t) &= \begin{cases} 1 & \text{if} \phi(t) \geq 0 \\ -1 & \text{if} \phi(t) < 0 \end{cases} \\ y(t) &= \text{sign}(\phi(t)) \\ y(t) &= \frac{\phi(t)}{\mid \phi(t) \mid} \end{aligned}$$

The derivative of $y(t)$ with respect to $\phi(t)$ becomes

$$\frac{\partial y(t)}{\partial \phi(t)} = 2\ \delta(\phi(t))$$

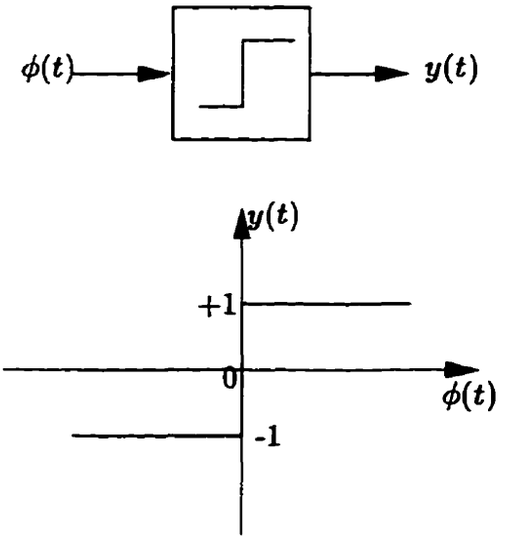


Figure 3.7: The comparator and its input-output characteristic.

which is nonzero only when $\phi(t) = 0$. This is shown in Fig. 3.8, where the signals $\phi(t)$, $y(t)$, and the derivatives of $y(t)$ are plotted versus time. Applying the chain

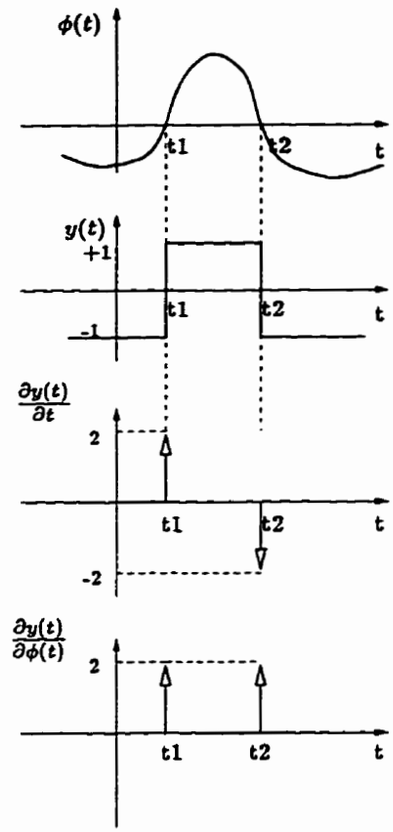


Figure 3.8: The input and output of the unclocked comparator.

rule gives

$$\frac{\partial y(t)}{\partial h} = \frac{\partial y(t)}{\partial \phi(t)} \frac{\partial \phi(t)}{\partial h} = 2\, \delta(\phi(t)) \frac{\partial \phi(t)}{\partial h}. \tag{3.4}$$

Eq.(3.4) relates the sensitivity of the output of the comparator to the sensitivity of its input with respect to h. This equation can be derived also by the *limiting approach*: we approximate the characteristic equation of the comparator with a parametric continuous function, then change the parameter to make the function

the same as that of the ideal comparator. One approximation is based on an exponential curve with a time constant of $\frac{1}{K}$ as the parameter.

$$y(t) = \left(1 - e^{-K\phi(t)}\right) u(\phi(t)) - \left(1 - e^{K\phi(t)}\right) u(-\phi(t))$$

where $u(\cdot)$ denotes the step function. This curve is shown in Fig. 3.9 for $K = 1$ and $K = 10$.

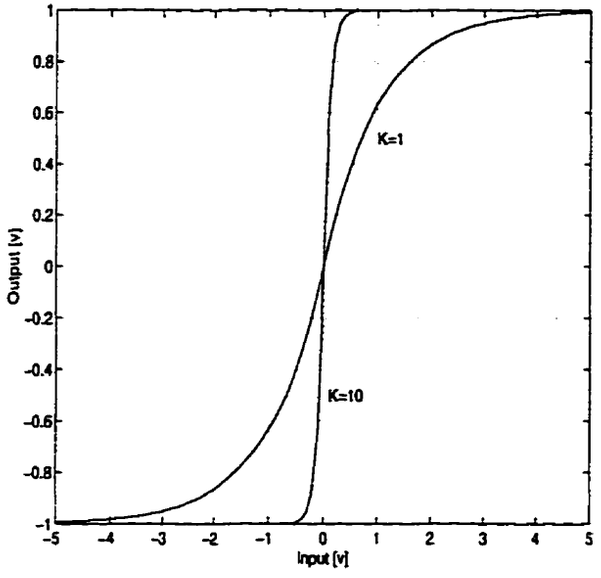


Figure 3.9: Exponential approximation of the comparator.

Another approximation is based on a trigonometric function,

$$y(t) = \frac{2}{\pi} \arctan\left(K\phi(t)\right). \tag{3.5}$$

Fig. 3.10 shows $y(t)$ for two values of K. Taking the differential of (3.5) with respect to h, and applying the chain rule

$$\frac{\partial y(t)}{\partial h} = \frac{\partial y(t)}{\partial \phi(t)} \frac{\partial \phi(t)}{\partial h} = \left(\frac{2K/\pi}{1 + K^2\phi(t)^2}\right) \frac{\partial \phi(t)}{\partial h}. \tag{3.6}$$

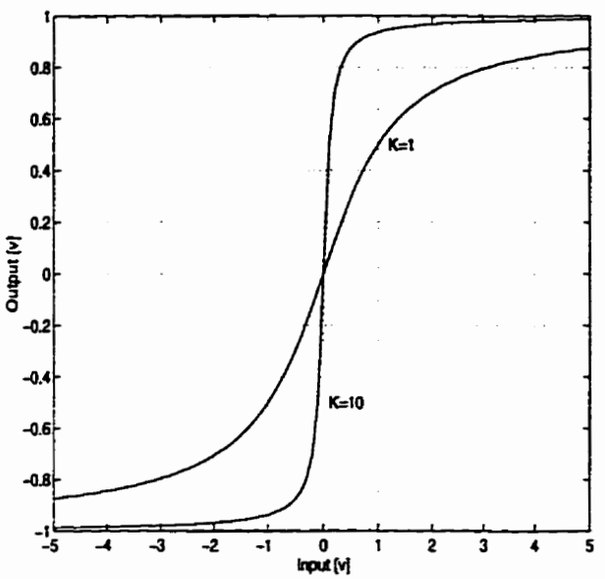


Figure 3.10: Trigonometric approximation of the comparator.

When K goes to infinity, the curve in Fig. 3.10 approaches the characteristic of an ideal comparator. On the other hand, when K goes to infinity

$$\lim_{K\to\infty} \frac{\partial y(t)}{\partial h} = \lim_{K\to\infty} \left(\frac{2K/\pi}{1+K^2\phi(t)^2} \right) \frac{\partial \phi(t)}{\partial h} = \begin{cases} 0 & \text{if} \phi(t) \neq 0 \\ \frac{\infty}{\infty \times 0} & \text{if} \phi(t) = 0 \end{cases} \tag{3.7}$$

where the sensitivity of the input of the comparator $\frac{\partial \phi(t)}{\partial h}$ is assumed to be finite. As K goes to infinity, the magnitude of $\frac{\partial y(t)}{\partial \phi(t)}$ at $\phi(t) = 0$ goes to infinity too (This is shown in Fig. 3.11 for $K = 1$ and $K = 10$). But the area under the curve is always 2 :

$$\text{The area of } \frac{\partial y(t)}{\partial \phi(t)} = \int_{-\infty}^{+\infty} \left(\frac{2K/\pi}{1+K^2\phi(t)^2} \right) d\phi = \frac{2}{\pi} \arctan(K\phi(t)) \Big|_{-\infty}^{+\infty} = \frac{2}{\pi}(\frac{\pi}{2} + \frac{\pi}{2}) = 2$$

Therefore, (3.7) can be written as

$$\lim_{K\to\infty} \frac{\partial y(t)}{\partial h} = 2\delta(\phi(t)) \frac{\partial \phi(t)}{\partial h}$$

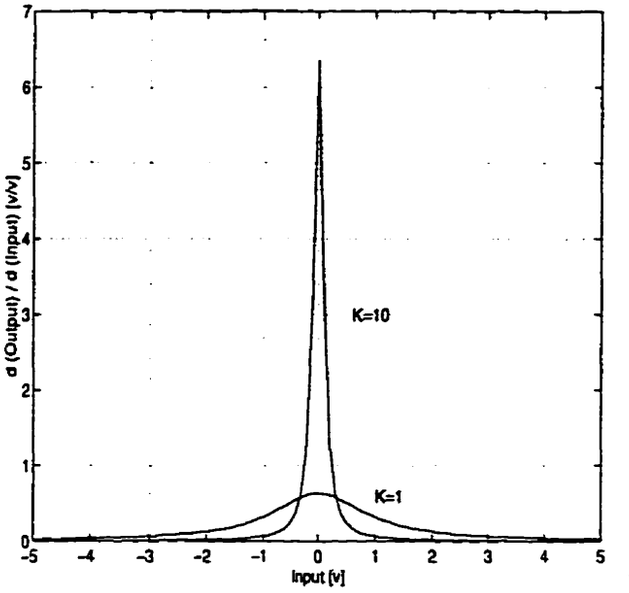


Figure 3.11: Differential of the trigonometric approximation of the comparator.

Sensitivity of The Clocked Comparator

The output of the clocked comparator changes only at the rising (or falling) edge of the clock and remains at the same level until the next edge. As shown in Fig. 3.12, the output changes at discrete instants of time, nT, and only when $\phi(nT)$ has a different polarity compared to $\phi(nT-T)$. If the input of the comparator crosses zero several times but returns back to the same polarity at the edge of the clock, the output of the comparator does not change, and $\frac{\partial y(nT)}{\partial \phi(nT)} = 0$. This is shown in Fig. 3.12 at instants $(nT-2T)$and $(nT-T)$. The output changes only if the input crosses zero (at any arbitrary time between the two time points) and also changes its polarity at the next edge of the clock. This happens at $t = nT$ in Fig. 3.12. Note that even though the signal $\phi(nT)$ crosses zero at an arbitrary time between $(nT-T)$ and (nT), the output of the comparator changes exactly at $t = nT$.

For the clocked comparator, the sensitivity $\frac{\partial y(nT)}{\partial h}$ does not contain any Dirac

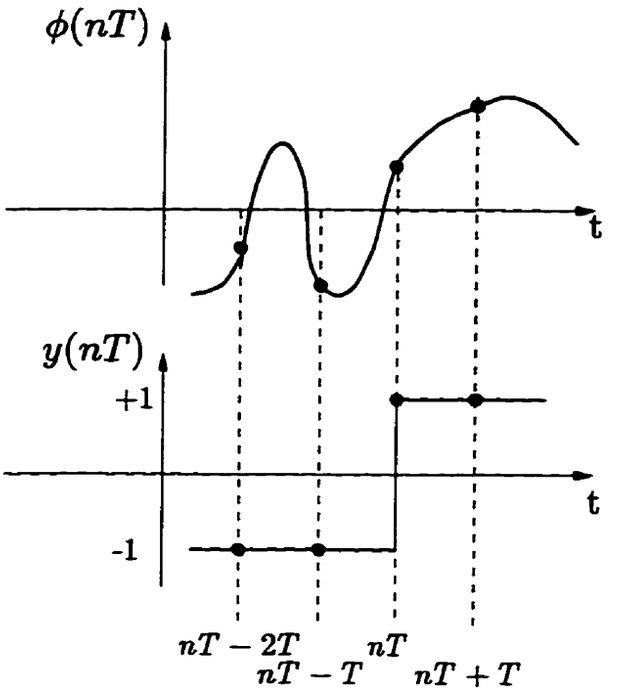


Figure 3.12: The input and output of the clocked comparator.

impulses. It is either zero or a pulse with the width of T and a limited amplitude. This is illustrated on Fig. 3.13. Assume we change one of the circuit elements to make a small change in the input of the comparator. This perturbed input is called $\phi_{pert}(nT)$. The corresponding output is called $y_{pert}(nT)$. Define the following increments

$$\begin{aligned} \Delta\phi(nT) &= \phi(nT) - \phi_{pert}(nT) \\ \Delta y(nT) &= y(nT) - y_{pert}(nT) \end{aligned}$$

If both $\phi(t)$ and $\phi_{pert}(t)$ have the same polarity, the output of the comparator does not change, and

$$\Delta y(nT) = 0$$

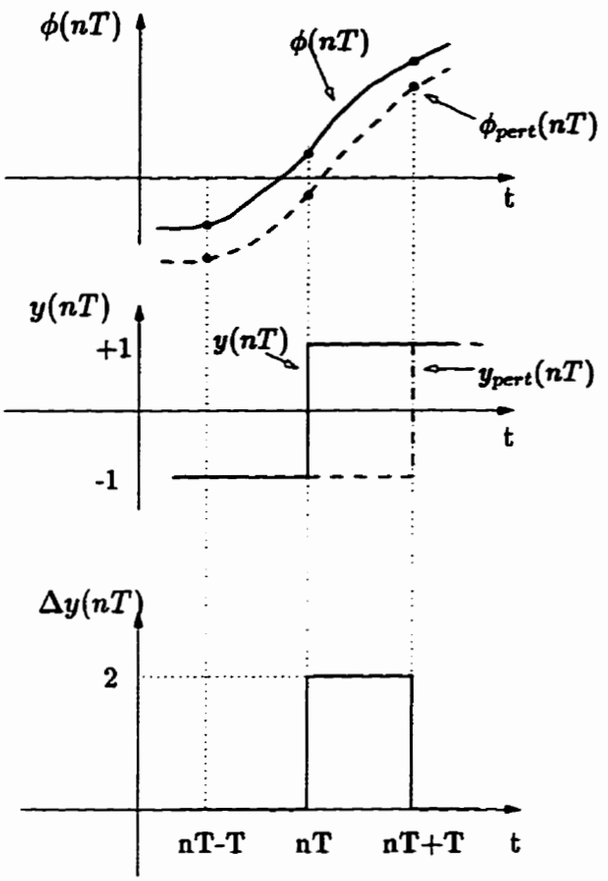


Figure 3.13: The change in the output of the clocked comparator due to the change in its input.

The output changes only if $\phi(t)$ and $\phi_{pert}(t)$ have different polarities when the edge of the clock comes. This happens at $t = nT$ in Fig.3.13. In this case, $\Delta y(nT)$ becomes 2 (or -2), and remains at the same level until the next edge of the clock.

$$\Delta y(nT) = \pm 2$$

Hence, the terms $\Delta y(nT)$, $\frac{\Delta y(nT)}{\Delta \phi(nT)}$, and $\frac{\Delta y(nT)}{\Delta h}$ are no longer Dirac impulses. They are pulses with the width of T, and a limited amplitude. In the case of the clocked

comparator, Eq.(3.4) becomes approximately

$$\begin{aligned}
\frac{\partial y(nT)}{\partial h} &\approx \frac{\Delta y(nT)}{\Delta h} = \frac{\Delta y(nT)}{\Delta \phi(nT)} \frac{\Delta \phi(nT)}{\Delta h} \\
\frac{\partial y(nT)}{\partial h} &\approx \begin{cases} 2\frac{\partial \phi(nT)}{\partial h} & \text{if } \phi(nT) \text{ changes polarity at } t = nT \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{3.8}$$

We will discuss the approximations made in (3.8) in the next section. Substituting (3.8) in (3.3) gives

$$\begin{aligned}
\frac{\partial \mathbf{x}(nT+T)}{\partial h} &= \frac{\partial \mathbf{M}}{\partial h}\mathbf{x}(nT) + \mathbf{M}\frac{\partial \mathbf{x}(nT)}{\partial h} + Real\left(\frac{\partial \mathbf{P}_{inp}}{\partial h} e^{j\omega_0 nT}\right) \\
&+ \frac{\partial \mathbf{P}_{y1}}{\partial h} y(nT) + 2\mathbf{P}_{y1}\frac{\partial \phi(nT)}{\partial h}
\end{aligned} \tag{3.9}$$

where $\phi(nT)$ is one of the elements in the vector $\mathbf{x}(nT)$, and $\frac{\partial \phi(nT)}{\partial h}$ in the vector $\frac{\partial \mathbf{x}(nT)}{\partial h}$. The last term in (3.9) is nonzero only if $\phi(nT)$ changes polarity at $t = nT$. This term exists only for one period, then becomes zero. Eq.(3.9) defines the *sensitivity network* for the second-order Delta-Sigma modulator in Fig. 3.2. The same equation can be derived for any other types/configurations of DSM. After generating the time domain sensitivity by (3.9), we take the FFT of the data points, and measure the magnitude of the component at the input frequency.

3.3.2 Approximations

There are two approximations in our derivations. The first approximation is in the condition specified in (3.8). It says if $\phi(nt)$ crosses zero, a pulse with the magnitude of $\frac{\partial \phi(nt)}{\partial h}$ is applied to the sensitivity network. This is not always true. There are some situations that even $\phi(nt)$ crosses zero but no $\Delta y(nT)$ is generated, and no stimulation is applied to the sensitivity network. It happens when the sensitivity of the input to the comparator, $\frac{\partial \phi(nt)}{\partial h}$, is zero or a very small number. In this case,

the term $\frac{\partial y(nT)}{\partial h}$ must be zero, but we consider it to be a small nonzero value of $2\frac{\partial \phi(nT)}{\partial h}$. We couldn't exclude these exemptions from the condition in (3.8), because it is not known how small $\frac{\partial \phi(nt)}{\partial h}$ should be in order no change is produced at the output of the comparator.

The second approximation is in ignoring the effect of $\Delta\phi(nT)$ in (3.8). This effect can be considered by multiplying $\frac{\partial y(nT)}{\partial h}$ by a factor λ :

$$\frac{\partial y(nT)}{\partial h} = 2\ \lambda \frac{\partial \phi(nT)}{\partial h}$$

We did not consider this factor because we do not have any estimate about the amount of perturbation needed in $\phi(nT)$ to produce a change in $y(nT)$.

More investigations of these approximations, and the methods to improve them are left for future research.

3.3.3 Example

The second-order Delta-Sigma modulator of Fig. 3.2 was simulated for its sensitivity. First, to get a feeling about the circuit sensitivity, we calculated the incremental sensitivity by perturbing the element values by 1%, and simulating the circuit several times for each new element value. The second column in Table-3.1 shows the magnitude of this incremental sensitivity. Next, we computed the differential sensitivity using the sensitivity network in (3.9). The third column in Table-3.1 shows the magnitude of the results.

Although the differential sensitivity is not always close to the incremental one, which is due to the approximations made in (3.8), it still conveys useful information about the circuit. For instance, the sensitivity of the modulator with respect to G3 and G4 is much less than the sensitivity with respect to G1 and G2. This

information is provided without the excessive computational cost associated with the incremental sensitivity.

Element	|Incremental Sensitivity|	|Differential Sensitivity|
G1	1.02	1.00
G2	1.54	1.00
G3	0.005	0.06
G4	0.003	0.05

Table 3.1: Sensitivities of the second-order continuous-time DSM (The incremental sensitivity was calculated by perturbing the element values by 1%, and the differential sensitivity was calculated using (3.9)).

Chapter 4

Group Delay and Group Delay Sensitivity of Periodically Switched Linear Networks

Group delay (sometimes called envelope delay) is an important measure used in the design of precision filters. If a band-limited signal is passed through a filter having a flat amplitude response over the bandwidth of that signal, one might expect the signal to be passed without distortion; however, this is not the case unless the filter also has a linear phase response over the signal bandwidth. In this case, all the components of the input signal in the passband are magnified with the same amplification factor, and delayed by the same amount of time.

In filter design, the magnitude response requirements are normally considered, and the corresponding phase response is ignored. This is because the reduction of antialiasing by stopband attenuation is the first issue, and phase information can be corrected later. For some applications, such as speech transmission, this con-

sideration is sufficient as the human ear is insensitive to the phase shift. For video applications, however, the distortion caused by system nonlinear phase response is unacceptable [30]. As a result, the subjects of "phase equalization" and "maximally flat group delay" are considered in several filter design texts [30–33]. The process of equalization consists of building up the passband delay of a filter to its peak level by the addition of all-pass networks. Linearizing the phase reduces the impulse and step response overshoots, and makes them more symmetric [33].

This chapter presents a fast and accurate method for the computation of group delay and the group delay sensitivity of periodically switched linear networks (PSLN). These networks consist of linear resistors, capacitors, inductors, independent sources, all four types of dependent sources, and frequency dependent amplifiers. Switches are modeled by the resistors that have arbitrary values (including zero and infinity) when they are closed or open. Examples of these networks include switched-capacitor (SC), switched-current (SI), and, frequency modulator and demodulator circuits. The method is accurate because no approximation is made, and efficient because some parts of the computations are performed only once, in a pre-processing step before simulation starts. The method can be used for analysis of SC filters where the resistances of the switches, or frequency-dependence and nonideal characteristics of the amplifiers can not be neglected. The other application is in the design of SI filters as long as their MOS transistors are linearized around the DC operating points.

Analysis and simulation of SC circuits, in time and frequency domains, are already well established, and we refer the interested reader to [34–42]. Several simulators were produced for simulation of SC circuits both at the behavioral and circuit levels. The circuit level SC simulators mainly assume that the opamps and the switches are ideal. The only linear elements allowed in the circuits are capac-

itors and voltage-controlled voltage sources. To consider nonideal switches, and nonideal opamps with offset voltages, input and output impedances, and frequency dependent open-loop gains, the simulator should be able to accept all types of dependent and independent sources as well as resistors. Some other simulators were therefore introduced in [43–45] for simulation of general linear switched networks. However, they do not provide the group delay and group delay sensitivity.

In this chapter we present an efficient method for the calculation of group delay in PSLN containing all types of linear elements. Furthermore, by computing the group delay sensitivity, we examine how a change in the element value h influences the group delay. This study can be utilized for the computerized optimization of switched capacitor and switched current filters.

A brief review of the frequency domain analysis of a PSLN is given in section 1. Section 2 to 5 present my contributions to this topic. I explain the computation of group delay in section 2, its sensitivity in section 3, and the computer algorithm in section 4. In case of sinusoidal inputs, a more efficient method of computing the vectors $\mathbf{P}$ and $\frac{\partial \mathbf{P}}{\partial \omega}$ is presented in section 5. A program was written in MATLAB based on the theories developed in this chapter. Appendix-B shows the detailed algorithm and intermediate matrix manipulations used in the program.

4.1 Frequency Domain Analysis of Periodically Switched Linear Networks

The frequency domain analysis of PSLN is explained in [3, 43]. We review it here to provide a proper background for the derivations presented in the next sections.

General switched networks may have more than two phases. We use N for the

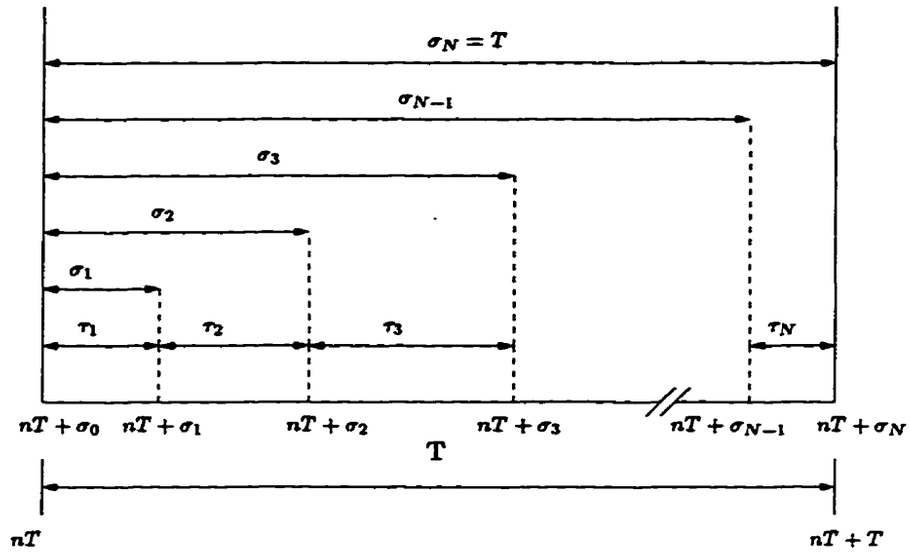


Figure 4.1: Timing definitions for an N-phase switched network.

total number of phases, and the subscript k for the kth phase. The various time slots, and related definitions are shown in Fig. 4.1. It is also shown that

$$\sigma_k = \sum_{i=1}^{k} \tau_i$$

In addition, we define

$$\sigma_0 = 0, \qquad \sigma_N = T$$

where T is the switching period. For the kth phase the network is described by the system of differential equations

$$\mathbf{G}_k \mathbf{v}_k(t) + \mathbf{C}_k \frac{d}{dt} \mathbf{v}_k(t) = \mathbf{g}_k w(t) \tag{4.1}$$

where $\mathbf{G}_k$ is the conductance matrix, $\mathbf{C}_k$ the capacitance matrix, and $\mathbf{v}_k(t)$ the unknown nodal vector in phase k. $\mathbf{g}_k$ is a vector defining the connection of the input $w(t)$ to the circuit. This equation can be decomposed into two sets of equations as

explained in [43]: one set of equations is valid inside the phase where switching does not occur, and the other set is valid across the switching instant. The following is a brief description of these equations.

1 - Discrete-time equations, valid only at the switching instants,

$$\mathbf{v}_k(nT+\sigma_k) = \mathbf{M}_k\mathbf{v}_{k-1}(nT+\sigma_{k-1}) + \mathbf{P}_k e^{j\omega_0(nT+\sigma_{k-1})}, k = 1,2,\ldots,N \quad (4.2)$$

where the input signal is assumed to be a complex exponential $e^{j\omega_0 t}$, $\mathbf{v}_k$ is the nodal vector in phase k, and the matrices $\mathbf{M}_k$ and $\mathbf{P}_k$ are obtained by integration, as explained in chapter 2. $\mathbf{M}_k$ is a real matrix that involves the zero input response of the circuit, and is independent of the input signal frequency ω_0. $\mathbf{P}_k$ is the zero state response of the network in the kth phase, and is a complex vector which depends on the input signal. The numerical computations of these matrices are discussed in Appendix-A. Eq.(4.2) relates the nodal vector in phase k to the nodal vector in the previous phase $k-1$, and to the input. We also define

$$\mathbf{v}_0 = \mathbf{v}_N$$

2 - Continuous-time equations, valid inside each phase,

$$\mathbf{G}_k\mathbf{v}_k(t) + \mathbf{C}_k\frac{d}{dt}\mathbf{v}_k(t) = \mathbf{g}_k w(t)\xi_k(t) + \mathbf{I_C} - \mathbf{F_C} \quad (4.3)$$

where $\mathbf{I_C}$ represents the initial conditions, and $\mathbf{F_C}$ the final conditions in each phase [45]. Initial conditions in phase k are determined by the final nodal voltages in the previous phase. Matrix $\mathbf{B}_k$ transforms the final conditions of the previous phase into initial conditions of the next phase as

$$\mathbf{I_C} = \mathbf{B}_k\mathbf{v}_{k-1}(nT+\sigma_{k-1})\ \delta(t-nT-\sigma_{k-1})$$

The final conditions in phase k are given by the nodal voltages at the end of the phase

$$\mathbf{F_C} = \mathbf{C}_k\mathbf{v}_k(nT+\sigma_k)\ \delta(t-nT-\sigma_k)$$

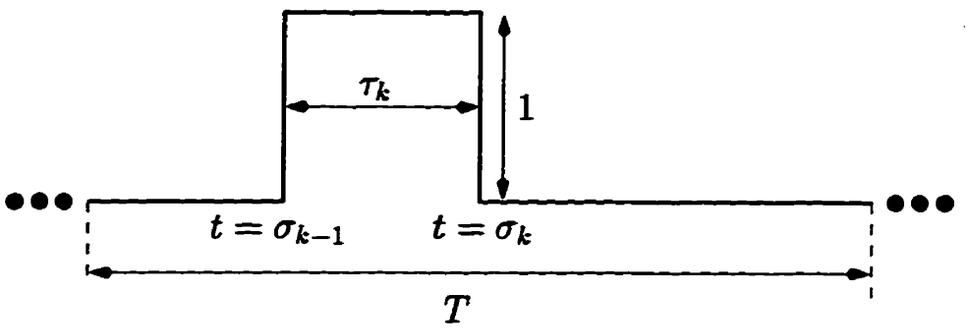


Figure 4.2: Definition of the kth window function $\xi_k(t)$.

where $\mathbf{C}_k$ is the matrix in (4.1). The function $\xi_k(t)$, called the window function, is periodic with the same period T. The kth window function $\xi_k(t)$ is sketched in Fig.4.2. It is nonzero only in the interval from $(nT + \sigma_{k-1})$ to $(nT + \sigma_k)$, and its height is unity. This function ensures that each system equation is valid in only one interval.

For frequency domain analysis, (4.2) and (4.3) must be expressed by their Fourier transforms. We first apply the Fourier transform to (4.2),

$$\mathcal{F}[\mathbf{v}_k(nT + \sigma_k)] = e^{j\omega\sigma_k}\widetilde{\mathbf{V}}_k, \quad k = 1, 2, \ldots, N$$

where the tilde over the variable denotes its Fourier transform. The Fourier transform of the windowed input signal is

$$\mathcal{F}\left[e^{j\omega_0(nT+\sigma_{k-1})}\right] = e^{j\omega_0\sigma_{k-1}}\frac{2\pi}{T}\sum_{n=-\infty}^{+\infty}\delta(\omega - \omega_0 - n\omega_s).$$

The infinite sums indicate that the result is valid only at frequencies $\omega = \omega_0 + n\omega_s$. Keeping this in mind, we drop the infinite sums and consider all frequency variables as $\omega = \omega_0$. The coefficient 2π generated by the Fourier transform is also dropped. The Fourier transform of (4.2) therefore becomes

$$e^{j\omega\sigma_k}\widetilde{\mathbf{V}}_k - e^{j\omega\sigma_{k-1}}\mathbf{M}_k\widetilde{\mathbf{V}}_{k-1} = \frac{1}{T}e^{j\omega\sigma_{k-1}}\mathbf{P}_k, \quad k = 1, 2, \ldots, N \qquad (4.4)$$

The equations for all phases can be written in matrix form

$$\begin{bmatrix} \mathbf{1}e^{j\omega\sigma_1} & \mathbf{0} & \mathbf{0} & \cdots & -\mathbf{M}_1 \\ -\mathbf{M}_2 e^{j\omega\sigma_1} & \mathbf{1}e^{j\omega\sigma_2} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & -\mathbf{M}_3 e^{j\omega\sigma_2} & \mathbf{1}e^{j\omega\sigma_3} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{M}_N e^{j\omega\sigma_{N-1}} & \mathbf{1}e^{j\omega T} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{V}}_1 \\ \widetilde{\mathbf{V}}_2 \\ \widetilde{\mathbf{V}}_3 \\ \cdots \\ \widetilde{\mathbf{V}}_N \end{bmatrix} = \frac{1}{T} \begin{bmatrix} \mathbf{P}_1 \\ e^{j\omega\sigma_1}\mathbf{P}_2 \\ e^{j\omega\sigma_2}\mathbf{P}_3 \\ \cdots \\ e^{j\omega\sigma_{N-1}}\mathbf{P}_N \end{bmatrix} \quad (4.5)$$

where $\mathbf{1}$ denotes the identity matrix. This system of equations provides $\widetilde{\mathbf{V}}_k$, the frequency response of the discrete-time system. We rewrite (4.5) with the following notation for future reference

$$\widehat{\mathbf{R}}\,\widehat{\mathbf{V}}_P = \widehat{\mathbf{W}} \quad (4.6)$$

where $\widehat{\mathbf{R}}$ denotes the system matrix, $\widehat{\mathbf{V}}_P$ the unknown vector, and $\widehat{\mathbf{W}}$ the right-hand side of (4.5).

Next, we apply Fourier transform to (4.3). Assuming a continuous exponential signal as the input, $w(t) = e^{j\omega_0 t}$

$$\mathcal{F}[w(t)] = W(\omega) = 2\pi\delta(\omega - \omega_0). \quad (4.7)$$

The Fourier transform of the product of a periodic function $\xi_k(t)$ and an arbitrary function $w(t)$ is the convolution of the respective transforms:

$$\begin{aligned} \mathcal{F}[w(t)\xi_k(t)] &= \frac{1}{2\pi}\int_{-\infty}^{+\infty} W(u)\Xi(\omega - u)du \\ &= \frac{1}{2\pi}\int_{-\infty}^{+\infty} W(u)2\pi\left[\sum_{n=-\infty}^{+\infty}\theta_{k,n}\delta[(\omega - n\omega_s) - u]\right]du \\ &= \sum_{n=-\infty}^{+\infty}\theta_{k,n}W(\omega - n\omega_s) \end{aligned} \quad (4.8)$$

where

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

is the switching frequency, and $\theta_{k,n}$ are the Fourier series coefficients of the window function $\xi_k(t)$

$$\theta_{k,n} = \frac{1}{T}\int_0^T \xi_k(t)e^{-jn\omega_s t}dt = \begin{cases} \frac{\tau_k}{T} & \text{for } n=0 \quad \text{(baseband)} \\ e^{-jn\omega_s\sigma_{k-1}}\ \frac{1-e^{-jn\omega_s\tau_k}}{jn\omega_s T} & \text{otherwise} \end{cases} \tag{4.9}$$

Substituting (4.7) in (4.8) gives

$$\mathcal{F}[w(t)\xi_k(t)] = 2\pi \sum_{n=-\infty}^{+\infty} \theta_{k,n}\delta(\omega - \omega_0 - n\omega_s).$$

As before, the infinite sums and the coefficient 2π are dropped, and all frequency variables are considered as $\omega = \omega_0$. We also denote Fourier transforms of the final conditions by $\widetilde{\mathbf{V}}_k$:

$$\widetilde{\mathbf{V}}_k = \mathcal{F}\left[\mathbf{v}_k(nT + \sigma_k\ \delta(t - nT - \sigma_k)\right].$$

The Fourier transform of (4.3) therefore becomes

$$(\mathbf{G}_k + j\omega\mathbf{C}_k)\,\mathbf{V}_k \;=\; \mathbf{g}_k\theta_{k,n} + \mathbf{B}_k\widetilde{\mathbf{V}}_{k-1} - \mathbf{C}_k\widetilde{\mathbf{V}}_k, \quad k = 1, 2, \ldots, N$$

The equation for all phases in matrix form is

$$\begin{bmatrix} \mathbf{G}_1 + j\omega\mathbf{C}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_2 + j\omega\mathbf{C}_2 & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{G}_N + j\omega\mathbf{C}_N \end{bmatrix} \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ \cdots \\ \mathbf{V}_N \end{bmatrix} = \begin{bmatrix} \mathbf{g}_1\theta_1 + \mathbf{B}_1\widetilde{\mathbf{V}}_N - \mathbf{C}_1\widetilde{\mathbf{V}}_1 \\ \mathbf{g}_2\theta_2 + \mathbf{B}_2\widetilde{\mathbf{V}}_1 - \mathbf{C}_2\widetilde{\mathbf{V}}_2 \\ \cdots \\ \mathbf{g}_N\theta_N + \mathbf{B}_N\widetilde{\mathbf{V}}_{N-1} - \mathbf{C}_N\widetilde{\mathbf{V}}_N \end{bmatrix} \tag{4.10}$$

or

$$\mathbf{R}\ \mathbf{V_P} = \mathbf{W} \tag{4.11}$$

The solution of (4.10) provides the frequency domain nodal voltages in each phase. The complete nodal vector is the summation of the vectors corresponding to each phase:

$$\mathbf{V} = \sum_{k=1}^{N} \mathbf{V}_k.$$

4.2 Group delay

Assume that the output of the filter is related to the nodal vector $\mathbf{V}$ by the selector vector $\mathbf{d}$

$$\phi = \mathbf{d}^t \mathbf{V}. \tag{4.12}$$

$\phi(j\omega)$ is a complex variable with magnitude $\mid \phi(j\omega) \mid$ and phase $\varphi(\omega)$

$$\phi(j\omega) = \mid \phi(j\omega) \mid e^{j\varphi(\omega)}, \tag{4.13}$$

and its group delay $\tau(\omega)$ is defined as

$$\tau(\omega) = -\frac{\partial \varphi(\omega)}{\partial \omega}.$$

Taking the logarithm of (4.13) and differentiating with respect to ω

$$\frac{1}{\phi}\frac{\partial \phi}{\partial \omega} = \frac{1}{\mid \phi \mid}\frac{\partial \mid \phi \mid}{\partial \omega} + j\frac{\partial \varphi(\omega)}{\partial \omega},$$

the group delay can be defined also as

$$\tau(\omega) = -\mathrm{Im}\left[\frac{1}{\phi}\frac{\partial \phi}{\partial \omega}\right]. \tag{4.14}$$

We therefore need to compute the derivatives of the frequency domain nodal vectors $\mathbf{V}_k$ with respect to frequency ω. First, differentiate (4.4) with respect to ω to

obtain $\frac{\partial \widetilde{\mathbf{V}}_k}{\partial \omega}$

$$j\sigma_k e^{j\omega\sigma_k}\widetilde{\mathbf{V}}_k + e^{j\omega\sigma_k}\frac{\partial \widetilde{\mathbf{V}}_k}{\partial \omega} - j\sigma_{k-1}e^{j\omega\sigma_{k-1}}\mathbf{M}_k\widetilde{\mathbf{V}}_{k-1} - e^{j\omega\sigma_{k-1}}\mathbf{M}_k\frac{\partial \widetilde{\mathbf{V}}_k}{\partial \omega} = \frac{1}{T}j\sigma_{k-1}e^{j\omega\sigma_{k-1}}\mathbf{P}_k + \frac{1}{T}e^{j\omega\sigma_{k-1}}\frac{\partial \mathbf{P}_k}{\partial \omega}, \quad k = 1, 2, \ldots, N \quad (4.15)$$

Since the vectors $\mathbf{P}_k$ depend on the input signal, their derivatives with respect to ω are not zero. An efficient method of numerical calculation of $\frac{\partial \mathbf{P}_k}{\partial \omega}$ is explained in Section 4.5. In matrix form, (4.15) becomes

$$\begin{bmatrix} \mathbf{1}e^{j\omega\sigma_1} & \mathbf{0} & \mathbf{0} & \cdots & -\mathbf{M}_1 \\ -\mathbf{M}_2e^{j\omega\sigma_1} & \mathbf{1}e^{j\omega\sigma_2} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & -\mathbf{M}_3e^{j\omega\sigma_2} & \mathbf{1}e^{j\omega\sigma_3} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{M}_Ne^{j\omega\sigma_{N-1}} & \mathbf{1}e^{j\omega T} \end{bmatrix} \begin{bmatrix} \frac{\partial \widetilde{\mathbf{V}}_1}{\partial \omega} \\ \frac{\partial \widetilde{\mathbf{V}}_2}{\partial \omega} \\ \frac{\partial \widetilde{\mathbf{V}}_3}{\partial \omega} \\ \cdots \\ \frac{\partial \widetilde{\mathbf{V}}_N}{\partial \omega} \end{bmatrix} =$$

$$\begin{bmatrix} -j\sigma_1e^{j\omega\sigma_1}\widetilde{\mathbf{V}}_1 \\ j\sigma_1e^{j\omega\sigma_1}\mathbf{M}_2\widetilde{\mathbf{V}}_1 - j\sigma_2e^{j\omega\sigma_2}\widetilde{\mathbf{V}}_2 \\ j\sigma_2e^{j\omega\sigma_2}\mathbf{M}_3\widetilde{\mathbf{V}}_2 - j\sigma_3e^{j\omega\sigma_3}\widetilde{\mathbf{V}}_3 \\ \cdots \\ j\sigma_{N-1}e^{j\omega\sigma_{N-1}}\mathbf{M}_N\widetilde{\mathbf{V}}_{N-1} - jTe^{j\omega T}\widetilde{\mathbf{V}}_N \end{bmatrix} + \frac{1}{T}\begin{bmatrix} \frac{\partial \mathbf{P}_1}{\partial \omega} \\ j\sigma_1e^{j\omega\sigma_1}\mathbf{P}_2 + e^{j\omega\sigma_1}\frac{\partial \mathbf{P}_2}{\partial \omega} \\ j\sigma_2e^{j\omega\sigma_2}\mathbf{P}_3 + e^{j\omega\sigma_2}\frac{\partial \mathbf{P}_3}{\partial \omega} \\ \cdots \\ j\sigma_{N-1}e^{j\omega\sigma_{N-1}}\mathbf{P}_N + e^{j\omega\sigma_{N-1}}\frac{\partial \mathbf{P}_N}{\partial \omega} \end{bmatrix}. \quad (4.16)$$

This is the same system as (4.5), only with a different right-hand side. The components $\widetilde{\mathbf{V}}_k$ at the right-hand side are provided by the solution of (4.5).

We next differentiate (4.10) with respect to ω to obtain $\frac{\partial \mathbf{V}_k}{\partial \omega}$. Since $\theta_{k,n}$ depends only on the switching frequency ω_s, its derivative with respect to the input frequency $\omega = \omega_0$ is zero.

$$(\mathbf{G}_k + j\omega\mathbf{C}_k)\frac{\partial \mathbf{V}_k}{\partial \omega} = \mathbf{B}_k\frac{\partial \widetilde{\mathbf{V}}_{k-1}}{\partial \omega} - \mathbf{C}_k\frac{\partial \widetilde{\mathbf{V}}_k}{\partial \omega} - j\mathbf{C}_k\mathbf{V}_k, \; k = 1, \ldots, N \quad (4.17)$$

This is also the same system as (4.10), only with a different right-hand side. The components $\mathbf{V}_k$ and $\frac{\partial \widetilde{\mathbf{V}}_k}{\partial \omega}$ are provided by the solutions of (4.10) and (4.16), respectively. The solution of (4.17) provides the derivatives $\frac{\partial \mathbf{V}_k}{\partial \omega}$ required for the computation of group delay. The algorithm for the group delay calculation is:

1. Prepare the matrices $\mathbf{M}_k$, $\mathbf{P}_k$, and $\frac{\partial \mathbf{P}_k}{\partial \omega}$ in each phase, $k = 1, 2 \ldots, N$.
2. Solve (4.5) to obtain $\widetilde{\mathbf{V}}_k$, the discrete-time nodal vector in the frequency domain.
3. Solve (4.10) to obtain $\mathbf{V}_k$, the continuous-time nodal vector in the frequency domain.
4. Add all $\mathbf{V}_k$ to get the complete nodal vector $\mathbf{V}$.
5. Using the solution of (4.5) construct the right-hand side of (4.16), and solve it to obtain $\frac{\partial \widetilde{\mathbf{V}}_k}{\partial \omega}$.
6. Using the solutions of (4.10) and (4.16) construct the right-hand side of (4.17), and solve it to obtain $\frac{\partial \mathbf{V}_k}{\partial \omega}$.
7. Add all $\frac{\partial \mathbf{V}_k}{\partial \omega}$ to get the complete nodal vector $\frac{\partial \mathbf{V}}{\partial \omega} = \sum_{k=1}^{N} \frac{\partial \mathbf{V}_k}{\partial \omega}$.
8. Specify the selector vector $\mathbf{d}$ which determines the output of interest in (4.12), and use (4.14) to calculate the output group delay.

4.2.1 Example

The above algorithm is applied on the simple circuit in Fig.4.3. Since the switches S1 and S2 are alternatively ON, the circuit is equivalent to an un-switched RC

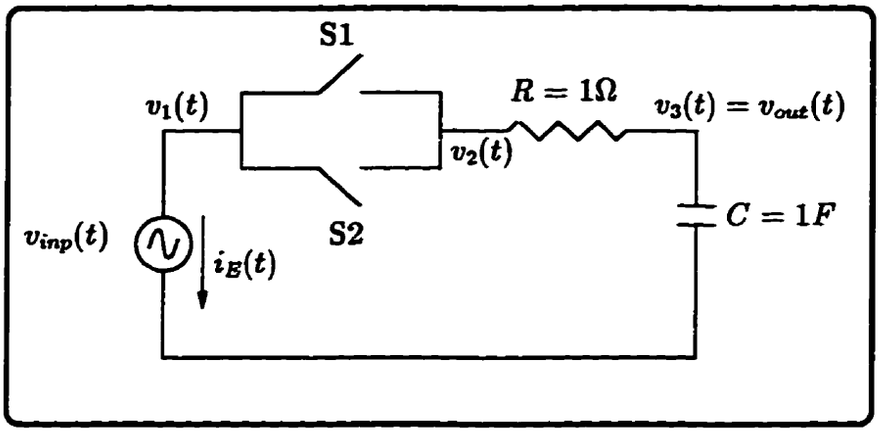


Figure 4.3: Switched RC circuit.

circuit whose group delay is $\tau(\omega) = \frac{1}{1+\omega^2}$. This allows us to verify the results produced by the algorithm. The number of phases is two, $k = 1, 2$. They are equally spaced, so $\tau_1 = \tau_2 = \frac{T}{2}$. If we are considering only the baseband, $n = 0$, and $\theta_{1,n} = \frac{\tau_k}{T} = \frac{1}{2}$. Define $\mathbf{W} = \mathbf{g}_1\theta_{1,0} = \mathbf{g}_2\theta_{2,0} = [0, 0, 0, \frac{1}{2}]^t$. Switches are modeled by the conductances g_{s1} and g_{s2} with zero values when they are open, and large values (10^6) when closed. The system MNA formulation is

$$\begin{bmatrix} g_{s1}+g_{s2} & -g_{s1}-g_{s2} & 0 & 1 \\ -g_{s1}-g_{s2} & g_{s1}+g_{s2}+G & -G & 0 \\ 0 & -G & G+Cs & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ I_E \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{2} \end{bmatrix}. \tag{4.18}$$

The discrete-time equation of (4.5) becomes

$$\begin{bmatrix} \mathbf{1}e^{j\omega T/2} & -\mathbf{M}_1 \\ -\mathbf{M}_2 e^{j\omega T/2} & \mathbf{1}e^{j\omega T} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{V}}_1 \\ \widetilde{\mathbf{V}}_2 \end{bmatrix} = \frac{1}{T} \begin{bmatrix} \mathbf{P}_1 \\ e^{j\omega_0 T/2}\mathbf{P}_2 \end{bmatrix}. \tag{4.19}$$

Since we consider the switches as conductances, the circuit topology is the same in both phases, and $\mathbf{B}_k = \mathbf{C}_k$, $k = 1, 2$. The continuous-time equation of (4.10)

becomes

$$\begin{bmatrix} \mathbf{G}_1 + j\omega\mathbf{C}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_2 + j\omega\mathbf{C}_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{W} + \mathbf{C}_1 \left(\widetilde{\mathbf{V}}_2 - \widetilde{\mathbf{V}}_1\right) \\ \mathbf{W} + \mathbf{C}_2 \left(\widetilde{\mathbf{V}}_1 - \widetilde{\mathbf{V}}_2\right) \end{bmatrix}. \tag{4.20}$$

The total vector $\mathbf{V}$ is equal to $\mathbf{V}_1 + \mathbf{V}_2$. To obtain $\frac{\partial \mathbf{V}}{\partial \omega}$, we first solve the following system of equation obtained from (4.16)

$$\begin{bmatrix} 1e^{j\omega T/2} & -\mathbf{M}_1 \\ -\mathbf{M}_2 e^{j\omega T/2} & 1e^{j\omega T} \end{bmatrix} \begin{bmatrix} \frac{\partial \widetilde{\mathbf{V}}_1}{\partial \omega} \\ \frac{\partial \widetilde{\mathbf{V}}_2}{\partial \omega} \end{bmatrix} = \begin{bmatrix} -1jT/2e^{j\omega T/2} & \mathbf{0} \\ jT/2e^{j\omega T/2}\mathbf{M}_2 & -1jTe^{j\omega T} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{V}}_1 \\ \widetilde{\mathbf{V}}_2 \end{bmatrix} \tag{4.21}$$

and consider its solution in the following equations obtained from (4.17)

$$\begin{aligned} \begin{bmatrix} \mathbf{G}_1 + j\omega\mathbf{C}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_2 + j\omega\mathbf{C}_2 \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{V}_1}{\partial \omega} \\ \frac{\partial \mathbf{V}_2}{\partial \omega} \end{bmatrix} &= \begin{bmatrix} -j\mathbf{C}_1 & \mathbf{0} \\ \mathbf{0} & -j\mathbf{C}_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{C}_1 \left(\frac{d\widetilde{\mathbf{V}}_2}{d\omega} - \frac{d\widetilde{\mathbf{V}}_1}{d\omega}\right) \\ \mathbf{C}_2 \left(\frac{d\widetilde{\mathbf{V}}_1}{d\omega} - \frac{d\widetilde{\mathbf{V}}_2}{d\omega}\right) \end{bmatrix}. \end{aligned} \tag{4.22}$$

The total vector $\frac{\partial \mathbf{V}}{\partial \omega}$ is equal to $\frac{\partial \mathbf{V}_1}{\partial \omega} + \frac{\partial \mathbf{V}_2}{\partial \omega}$. The output group delay is calculated by

$$\tau(\omega) = -\mathrm{Im}\left[\frac{1}{\mathbf{V}(3)} \frac{\partial \mathbf{V}(3)}{\partial \omega}\right]. \tag{4.23}$$

Assuming unit values for the elements in Fig. 4.3, and $T = 0.01$, the group delay is computed as shown in Fig. 4.4. The curve is in agreement with the analytical expression of the group delay for this simple RC circuit, $\tau(\omega) = \frac{1}{\omega^2+1}$.

4.3 Group Delay Sensitivity

In this section, we see how the change in the element value h influences the group delay. This study can be utilized in the computerized optimization of switched-capacitor and switched-current filters. The group delay sensitivity is calculated by

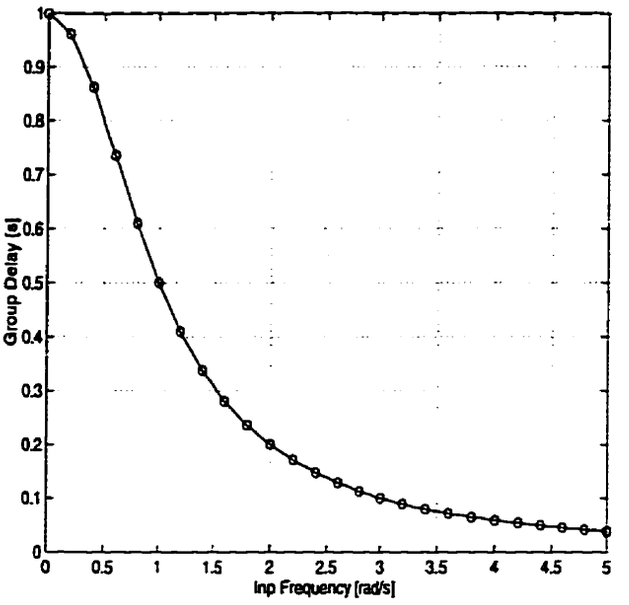


Figure 4.4: Group delay of the output in Fig. 4.3

differentiating (4.14) with respect to element h

$$\frac{\partial \tau(\omega)}{\partial h} = \mathrm{Im}\left[\frac{1}{\phi^2}\frac{\partial \phi}{\partial h}\frac{\partial \phi}{\partial \omega} - \frac{1}{\phi}\frac{\partial^2 \phi}{\partial h \partial \omega}\right] \tag{4.24}$$

Differentiating (4.6) with respect to ω and h, separately, gives

$$\widehat{\mathbf{R}}\frac{\partial \widehat{\mathbf{V}}_P}{\partial \omega} = -\frac{\partial \widehat{\mathbf{R}}}{\partial \omega}\widehat{\mathbf{V}}_P + \frac{\partial \widehat{\mathbf{W}}}{\partial \omega} \tag{4.25}$$

$$\widehat{\mathbf{R}}\frac{\partial \widehat{\mathbf{V}}_P}{\partial h} = -\frac{\partial \widehat{\mathbf{R}}}{\partial h}\widehat{\mathbf{V}}_P + \frac{\partial \widehat{\mathbf{W}}}{\partial h} \tag{4.26}$$

Differentiate either (4.25) with respect to h or (4.26) with respect to ω to obtain

$$\widehat{\mathbf{R}}\frac{\partial^2 \widehat{\mathbf{V}}_P}{\partial h \partial \omega} = -\frac{\partial \widehat{\mathbf{R}}}{\partial h}\frac{\partial \widehat{\mathbf{V}}_P}{\partial \omega} - \frac{\partial^2 \widehat{\mathbf{R}}}{\partial h \partial \omega}\widehat{\mathbf{V}}_P - \frac{\partial \widehat{\mathbf{R}}}{\partial \omega}\frac{\partial \widehat{\mathbf{V}}_P}{\partial h} + \frac{\partial^2 \widehat{\mathbf{W}}}{\partial h \partial \omega} \tag{4.27}$$

The discrete-time equations (4.25) to (4.27) have the same system matrix $\widehat{\mathbf{R}}$, and different right-hand sides. To generate the terms at the right-hand sides, we need to precompute $\mathbf{P}_k$, $\frac{\partial \mathbf{P}_k}{\partial \omega}$, $\frac{\partial \mathbf{P}_k}{\partial h}$, and $\frac{\partial^2 \mathbf{P}_k}{\partial h \partial \omega}$ vectors, and the matrix $\mathbf{M}_k$ with the same

derivatives. Refer to Section 4.5 and also Appendix-A for the numerical computations of these matrices. Solutions of the discrete-time equations are used to construct the continuous-time equations. Considering the continuous-time equation (4.11), and differentiating it with respect to ω and h

$$\mathbf{R}\frac{\partial \mathbf{V_P}}{\partial \omega} = -\frac{\partial \mathbf{R}}{\partial \omega}\mathbf{V_P} + \frac{\partial \mathbf{W}}{\partial \omega} \tag{4.28}$$

$$\mathbf{R}\frac{\partial \mathbf{V_P}}{\partial h} = -\frac{\partial \mathbf{R}}{\partial h}\mathbf{V_P} + \frac{\partial \mathbf{W}}{\partial h} \tag{4.29}$$

$$\mathbf{R}\frac{\partial^2 \mathbf{V_P}}{\partial h \partial \omega} = -\frac{\partial \mathbf{R}}{\partial h}\frac{\partial \mathbf{V_P}}{\partial \omega} - \frac{\partial^2 \mathbf{R}}{\partial h \partial \omega}\mathbf{V_P} - \frac{\partial \mathbf{R}}{\partial \omega}\frac{\partial \mathbf{V_P}}{\partial h} + \frac{\partial^2 \mathbf{W}}{\partial h \partial \omega} \tag{4.30}$$

where

$$\frac{\partial \mathbf{W}}{\partial \omega} = \begin{bmatrix} \mathbf{B}_1\frac{\partial \tilde{\mathbf{V}}_N}{\partial \omega} - \mathbf{C}_1\frac{\partial \tilde{\mathbf{V}}_1}{\partial \omega} \\ \mathbf{B}_2\frac{\partial \tilde{\mathbf{V}}_1}{\partial \omega} - \mathbf{C}_2\frac{\partial \tilde{\mathbf{V}}_2}{\partial \omega} \\ \cdots \\ \mathbf{B}_N\frac{\partial \tilde{\mathbf{V}}_{N-1}}{\partial \omega} - \mathbf{C}_N\frac{\partial \tilde{\mathbf{V}}_N}{\partial \omega} \end{bmatrix},$$

$$\frac{\partial \mathbf{W}}{\partial h} = \begin{bmatrix} \frac{\partial \mathbf{B}_1}{\partial h}\tilde{\mathbf{V}}_N - \frac{\partial \mathbf{C}_1}{\partial h}\tilde{\mathbf{V}}_1 \\ \frac{\partial \mathbf{B}_2}{\partial h}\tilde{\mathbf{V}}_1 - \frac{\partial \mathbf{C}_2}{\partial h}\tilde{\mathbf{V}}_2 \\ \cdots \\ \frac{\partial \mathbf{B}_N}{\partial h}\tilde{\mathbf{V}}_{N-1} - \frac{\partial \mathbf{C}_N}{\partial h}\tilde{\mathbf{V}}_N \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1\frac{\partial \tilde{\mathbf{V}}_N}{\partial h} - \mathbf{C}_1\frac{\partial \tilde{\mathbf{V}}_1}{\partial h} \\ \mathbf{B}_2\frac{\partial \tilde{\mathbf{V}}_1}{\partial h} - \mathbf{C}_2\frac{\partial \tilde{\mathbf{V}}_2}{\partial h} \\ \cdots \\ \mathbf{B}_N\frac{\partial \tilde{\mathbf{V}}_{N-1}}{\partial h} - \mathbf{C}_N\frac{\partial \tilde{\mathbf{V}}_N}{\partial h} \end{bmatrix},$$

$$\frac{\partial^2 \mathbf{W}}{\partial h \partial \omega} = \begin{bmatrix} \frac{\partial \mathbf{B}_1}{\partial h}\frac{\partial \tilde{\mathbf{V}}_N}{\partial \omega} - \frac{\partial \mathbf{C}_1}{\partial h}\frac{\partial \tilde{\mathbf{V}}_1}{\partial \omega} \\ \frac{\partial \mathbf{B}_2}{\partial h}\frac{\partial \tilde{\mathbf{V}}_1}{\partial \omega} - \frac{\partial \mathbf{C}_2}{\partial h}\frac{\partial \tilde{\mathbf{V}}_2}{\partial \omega} \\ \cdots \\ \frac{\partial \mathbf{B}_N}{\partial h}\frac{\partial \tilde{\mathbf{V}}_{N-1}}{\partial \omega} - \frac{\partial \mathbf{C}_N}{\partial h}\frac{\partial \tilde{\mathbf{V}}_N}{\partial \omega} \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1\frac{\partial^2 \tilde{\mathbf{V}}_N}{\partial h \partial \omega} - \mathbf{C}_1\frac{\partial^2 \tilde{\mathbf{V}}_1}{\partial h \partial \omega} \\ \mathbf{B}_2\frac{\partial^2 \tilde{\mathbf{V}}_1}{\partial h \partial \omega} - \mathbf{C}_2\frac{\partial^2 \tilde{\mathbf{V}}_2}{\partial h \partial \omega} \\ \cdots \\ \mathbf{B}_N\frac{\partial^2 \tilde{\mathbf{V}}_{N-1}}{\partial h \partial \omega} - \mathbf{C}_N\frac{\partial^2 \tilde{\mathbf{V}}_N}{\partial h \partial \omega} \end{bmatrix},$$

are provided by the solutions of (4.25) to (4.27). The complete vectors of nodal voltages and their derivatives are the summation of the vectors corresponding to each phase. The particular components of these vectors, corresponding to the output node of interest, are substituted into (4.24) to obtain the group delay sensitivity.

4.4 The Algorithm

To illustrate the intermediate steps, and to organize the method for the purpose of computer programming, we provide here the algorithm of the method. The reader is referred to Appendix-B for the detailed structures of the matrices. A program, called GRPSN, was written in MATLAB based on this algorithm. It accepts all types of periodically switched linear networks. The program was tested on several switched networks. One example is given at the end of this section.

Part 1 : Pre-Processing

We apply the one-Graph modified nodal analysis to formulate the circuit equations. The switches are modeled as resistors with a small resistance when they are closed and infinite resistance when open. The topology of the circuit therefore does not change during the different phases, and the matrices $\mathbf{B}_k$ become equal to $\mathbf{C}_k$. The number of phases is denoted by N.

1-1- Prepare $(\mathbf{G}_1, \mathbf{C}_1), (\mathbf{G}_2, \mathbf{C}_2), \ldots, (\mathbf{G}_N, \mathbf{C}_N)$, where the matrices $\mathbf{G}_k$ and $\mathbf{C}_k$ construct the system matrix $\mathbf{R}_k$ during phase k, i.e. $\mathbf{R}_k = \mathbf{G}_k + s\ \mathbf{C}_k$.

1-2- Prepare $\left(\frac{\partial \mathbf{G}_1}{\partial h}, \frac{\partial \mathbf{C}_1}{\partial h}\right), \left(\frac{\partial \mathbf{G}_2}{\partial h}, \frac{\partial \mathbf{C}_2}{\partial h}\right), \ldots, \left(\frac{\partial \mathbf{G}_N}{\partial h}, \frac{\partial \mathbf{C}_N}{\partial h}\right)$

1-3- Prepare $(\mathbf{P}_1, \mathbf{M}_1), (\mathbf{P}_2, \mathbf{M}_2), \ldots, (\mathbf{P}_N, \mathbf{M}_N)$

1-4- Prepare $\left(\frac{\partial \mathbf{P}_1}{\partial h}, \frac{\partial \mathbf{M}_1}{\partial h}\right), \left(\frac{\partial \mathbf{P}_2}{\partial h}, \frac{\partial \mathbf{M}_2}{\partial h}\right), \ldots, \left(\frac{\partial \mathbf{P}_N}{\partial h}, \frac{\partial \mathbf{M}_N}{\partial h}\right)$

1-5- Prepare $\left(\frac{\partial \mathbf{P}_1}{\partial \omega}\right), \left(\frac{\partial \mathbf{P}_2}{\partial \omega}\right), \ldots, \left(\frac{\partial \mathbf{P}_N}{\partial \omega}\right)$

1-6- Prepare $\left(\frac{\partial^2 \mathbf{P}_1}{\partial h \partial \omega}\right), \left(\frac{\partial^2 \mathbf{P}_2}{\partial h \partial \omega}\right), \ldots, \left(\frac{\partial^2 \mathbf{P}_N}{\partial h \partial \omega}\right)$

Part 2 : Solution of the Discrete-Time Equations

2-1- Construct the following matrices related to the Discrete-Time set of equations:
$\widehat{\mathbf{R}}$, $\widehat{\mathbf{W}}$, $\frac{\partial \widehat{\mathbf{R}}}{\partial \omega}$, $\frac{\partial \widehat{\mathbf{R}}}{\partial h}$, $\frac{\partial^2 \widehat{\mathbf{R}}}{\partial h \partial \omega}$, $\frac{\partial \widehat{\mathbf{W}}}{\partial \omega}$, $\frac{\partial \widehat{\mathbf{W}}}{\partial h}$, $\frac{\partial^2 \widehat{\mathbf{W}}}{\partial h \partial \omega}$

2-2- Compute

$$\widehat{\mathbf{V}}_P = \widehat{\mathbf{R}}^{-1}\, \widehat{\mathbf{W}} \tag{4.31}$$

2-3- Using $\widehat{\mathbf{V}}_P$ from (4.31), compute

$$\frac{\partial \widehat{\mathbf{V}}_P}{\partial \omega} = -\widehat{\mathbf{R}}^{-1} \frac{\partial \widehat{\mathbf{R}}}{\partial \omega} \widehat{\mathbf{V}}_P + \widehat{\mathbf{R}}^{-1} \frac{\partial \widehat{\mathbf{W}}}{\partial \omega} \tag{4.32}$$

2-4- Using $\widehat{\mathbf{V}}_P$ from (4.31), compute

$$\frac{\partial \widehat{\mathbf{V}}_P}{\partial h} = -\widehat{\mathbf{R}}^{-1} \frac{\partial \widehat{\mathbf{R}}}{\partial h} \widehat{\mathbf{V}}_P + \widehat{\mathbf{R}}^{-1} \frac{\partial \widehat{\mathbf{W}}}{\partial h} \tag{4.33}$$

2-5- Using $\widehat{\mathbf{V}}_P$, $\frac{\partial \widehat{\mathbf{V}}_P}{\partial \omega}$, and $\frac{\partial \widehat{\mathbf{V}}_P}{\partial h}$ obtained in the previous steps, compute

$$\frac{\partial^2 \widehat{\mathbf{V}}_P}{\partial h \partial \omega} = \widehat{\mathbf{R}}^{-1} \left[-\frac{\partial \widehat{\mathbf{R}}}{\partial h} \frac{\partial \widehat{\mathbf{V}}_P}{\partial \omega} - \frac{\partial^2 \widehat{\mathbf{R}}}{\partial h \partial \omega} \widehat{\mathbf{V}}_P - \frac{\partial \widehat{\mathbf{R}}}{\partial \omega} \frac{\partial \widehat{\mathbf{V}}_P}{\partial h} + \frac{\partial^2 \widehat{\mathbf{W}}}{\partial h \partial \omega} \right] \tag{4.34}$$

Part 3 : Solution of the Continuous-Time Equations

3-1- Construct the following matrices related to the Continuous-Time set of equations. Use the results of (4.31) to (4.34) to construct these matrices:
$\mathbf{R}$, $\mathbf{W}$, $\frac{\partial \mathbf{R}}{\partial h}$, $\frac{\partial \mathbf{R}}{\partial \omega}$, $\frac{\partial^2 \mathbf{R}}{\partial h \partial \omega}$, $\frac{\partial \mathbf{W}}{\partial h}$, $\frac{\partial \mathbf{W}}{\partial \omega}$, $\frac{\partial^2 \mathbf{W}}{\partial h \partial \omega}$

3-2- Compute

$$\mathbf{V_P} = \mathbf{R}^{-1}\ \mathbf{W} \tag{4.35}$$

3-3- Using $\mathbf{V_P}$ from (4.35), compute

$$\frac{\partial \mathbf{V_P}}{\partial \omega} = -\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial \omega}\mathbf{V_P} + \mathbf{R}^{-1}\frac{\partial \mathbf{W}}{\partial \omega} \tag{4.36}$$

3-4- Using $\mathbf{V_P}$ from (4.35), compute

$$\frac{\partial \mathbf{V_P}}{\partial h} = -\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial h}\mathbf{V_P} + \mathbf{R}^{-1}\frac{\partial \mathbf{W}}{\partial h} \tag{4.37}$$

3-5- Using $\mathbf{V_P}$, $\frac{\partial \mathbf{V_P}}{\partial \omega}$, and $\frac{\partial \mathbf{V_P}}{\partial h}$ obtained in the previous equations, compute

$$\frac{\partial^2 \mathbf{V_P}}{\partial h \partial \omega} = \mathbf{R}^{-1}\left[-\frac{\partial \mathbf{R}}{\partial h}\frac{\partial \mathbf{V_P}}{\partial \omega} - \frac{\partial^2 \mathbf{R}}{\partial h \partial \omega}\mathbf{V_P} - \frac{\partial \mathbf{R}}{\partial \omega}\frac{\partial \mathbf{V_P}}{\partial h} + \frac{\partial^2 \mathbf{W}}{\partial h \partial \omega}\right] \tag{4.38}$$

Part 4 : Computing the Group Delay, and Group Delay Sensitivity

The complete vector of nodal voltages is the summation of the vectors corresponding to each phase:

$$\begin{aligned}
\mathbf{V} &= \sum_{k=1}^{N} \mathbf{V}_k \\
\frac{\partial \mathbf{V}}{\partial h} &= \sum_{k=1}^{N} \frac{\partial \mathbf{V}_k}{\partial h} \\
\frac{\partial \mathbf{V}}{\partial \omega} &= \sum_{k=1}^{N} \frac{\partial \mathbf{V}_k}{\partial \omega} \\
\frac{\partial^2 \mathbf{V}}{\partial h \partial \omega} &= \sum_{k=1}^{N} \frac{\partial^2 \mathbf{V}_k}{\partial h \partial \omega}
\end{aligned}$$

Assume that the output is related to the vector $\mathbf{V}$ by the selector vector $\mathbf{d}$, i.e. $\phi = \mathbf{d}^t\ \mathbf{V}$). The group delay is calculated by (4.14), and the group delay sensitivity by (4.24).

4.4.1 Example

The standard two-phase switched-capacitor band-pass filter [39], shown in Fig. 4.5, is simulated by GRPSN. The switches are modeled as conductances with large values when they are closed ($10^6 S$) and zero values when open. The element values are chosen to provide a center frequency of 1kHz, and quality factor of $Q = 30$. The switching frequency is 20kHz, with equal phases. A 100Ω resistor is connected to the output as the load. The filter's group delay and its sensitivity with respect to $C_4 = 1nF$ were computed as shown in Fig. 4.6, and Fig.4.7. The group delay is not flat inside the pass band. Its normalized sensitivity is around -1, suggesting that increasing C_4 reduces the group delay, and makes it more flat. Considering

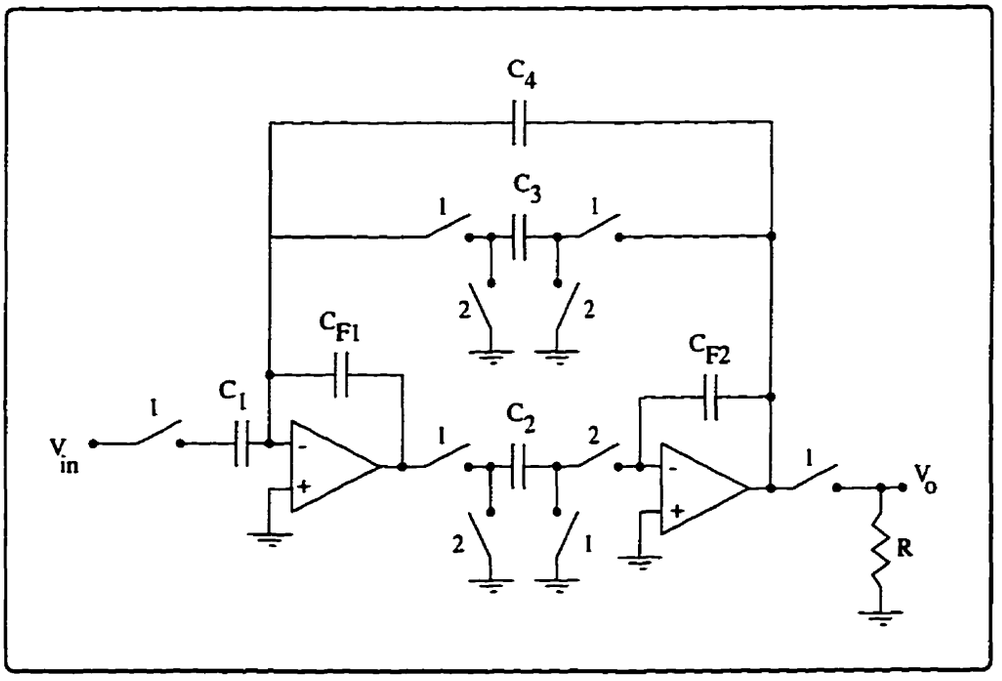


Figure 4.5: A standard SC band-pass filter with $f_0 = 1kHz$, ($C_1 = 10nF, C_2 = C_3 = 9.781nF, C_4 = 1nF, C_{F1} = C_{F2} = 31.25nF, R = 100\Omega$).

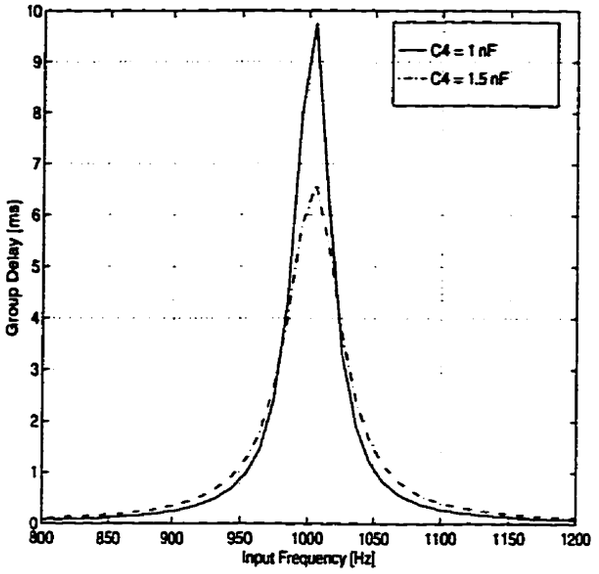


Figure 4.6: Group delay of the output in Fig. 4.5.

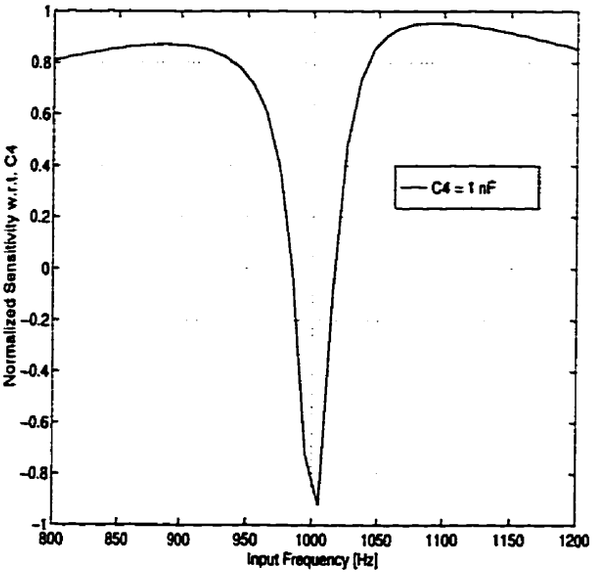


Figure 4.7: Group delay sensitivity of the output in Fig. 4.5 with respect to C_4.

$C_4 = 1.5nF$, we computed the group delay and plotted it in Fig. 4.6. The modified circuit has flatter group delay. This example indicates that by coupling GRPSN with an optimizer, an automatic tool is obtained for the design of distortionless filters.

In the next example, we consider some of the nonideal effects of the elements in Fig. 4.5. The switches are replaced by the resistors with the value of $1k\Omega$ when closed, and $1M\Omega$ when open. The opamp is nonideal with the open-loop gain of

$$A(s) = \frac{A_0}{\left(1 + \frac{s}{\omega_b}\right)}$$

where $A_0 = 1000\ \frac{V}{V}$, and $\omega_b = 100\ Hz$. The circuit group delay and its sensitivity with respect to C_4 were computed as shown in Fig. 4.8, and Fig.4.9.

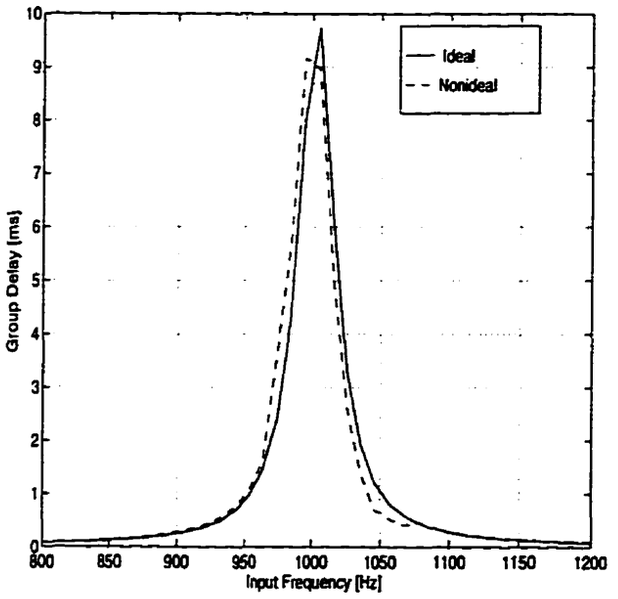


Figure 4.8: Group delay of the output in Fig. 4.5 with ideal opamp (infinite gain-bandwidth) and switches ($R_{ON} = 0, R_{OFF} = \infty$) in compare to nonideal opamp (gain-bandwidth=10^5) and switches ($R_{ON} = 1k\Omega, R_{OFF} = 1M\Omega$).

This simulation demonstrates GRPSN's ability in simulating switched-capacitor networks where the resistances of the switches, or frequency-dependence of the amplifiers can not be neglected. GRPSN also simulates the switched-current circuits as long as the MOS transistors are modeled by linear components such as dependent sources, capacitors, and resistors.

4.5 Computation of P and $\frac{\partial \mathbf{P}}{\partial \omega}$ with Sinusoidal Inputs

The numerical computations of **P**, **M**, and their derivatives with respect to h are explained in Appendix-A. In this section, we compute the derivatives of **P** and

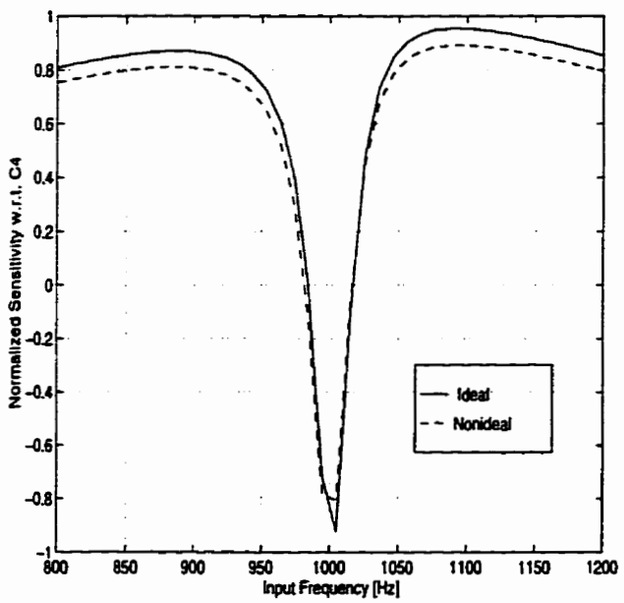


Figure 4.9: Group delay sensitivity of the output in Fig. 4.5 with ideal opamp (infinite gain-bandwidth) and switches ($R_{ON} = 0, R_{OFF} = \infty$) in compare to nonideal opamp (gain-bandwidth=10^5) and switches ($R_{ON} = 1k\Omega, R_{OFF} = 1M\Omega$).

M with respect to the input frequency ω. Assume that the input is a complex exponential, $w(t) = e^{j\omega t}$. The derivative of **M** with respect to ω is zero because **M** relates to the zero-input response of the network and does not depend on the input frequency.

$$\frac{\partial \mathbf{M}}{\partial \omega} = \mathbf{0} \tag{4.39}$$

To compute $\frac{\partial \mathbf{P}}{\partial \omega}$ we note that $W(s) = \frac{1}{s-j\omega}$.

$$\frac{\partial \mathbf{P}}{\partial \omega} = \frac{\partial}{\partial \omega}\mathcal{L}^{-1}\left(\mathbf{R}^{-1}\frac{1}{s-j\omega}\right)\Bigg|_{t=T} = \mathcal{L}^{-1}\left(\mathbf{R}^{-1}\frac{j}{(s-j\omega)^2}\right)\Bigg|_{t=T} \tag{4.40}$$

Similarly, it can be verified that

$$\frac{\partial^2 \mathbf{P}}{\partial h \partial \omega} = \mathcal{L}^{-1}\left(-\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial h}\mathbf{R}^{-1}\frac{j}{(s-j\omega)^2}\right)\Bigg|_{t=T} \tag{4.41}$$

$$\frac{\partial^2 \mathbf{M}}{\partial h \partial \omega} = \mathbf{0} \tag{4.42}$$

The Laplace inversions incorporated in (4.40) and (4.41) can be simultaneously performed using the stepping algorithm and numerical Laplace inversion method explained in Appendix-A.

Normally, when the frequency of the input signal changes, a new integration must be performed to recompute vector **P** and its derivatives. However, for *sinusoidal* inputs, once the vectors $\mathbf{P}$, $\frac{\partial \mathbf{P}}{\partial h}$, $\frac{\partial \mathbf{P}}{\partial \omega}$, and $\frac{\partial^2 \mathbf{P}}{\partial h \partial \omega}$ are calculated at one input frequency, they can be computed at the other frequencies without any further integration. To explain this, we start from the theory developed in [46] and extend it to our application. In [46] an efficient method for the time domain solution of linear circuits to sinusoidal inputs is given. It is proved that if the zero-state response of the linear circuit to a sinusoidal input with the frequency of ω_1 is known as $\mathbf{x}(t)$,

then the zero-state response of the circuit at another frequency ω_i can be computed as

$$\mathbf{y}(t) = (\mathbf{G} + j\omega_i\mathbf{C})^{-1}\left[(\mathbf{G} + j\omega_1\mathbf{C})\,\mathbf{x}(t) + \mathbf{d}e^{j\omega_i t} - \mathbf{d}e^{j\omega_1 t}\right] \tag{4.43}$$

where $\mathbf{d}$ is a constant vector related to the input node, and ω_i, $(i = 2, 3, \ldots, m)$ are the different input frequencies at which the response is needed. We extend this idea to efficiently compute the vector $\mathbf{P}$ and its derivatives at different input frequencies. Consider the following notation

$$\mathbf{R}_i = \mathbf{G} + j\omega_i\mathbf{C} \qquad i = 1, 2, \ldots, m$$

and rewrite (4.43) as

$$\mathbf{y}(t) = \mathbf{R}_i^{-1}\left[\mathbf{R}_1\mathbf{x}(t) + \mathbf{d}e^{j\omega_i T} - \mathbf{d}e^{j\omega_1 T}\right] \qquad i = 2, 3, \ldots, m \tag{4.44}$$

Since the zero-state response of the circuit at $t = T$ is equivalent to the vector $\mathbf{P}$, we write

$$\mathbf{x}(T) = \mathbf{P} \tag{4.45}$$

$$\mathbf{y}(T) = \mathbf{P}_i \tag{4.46}$$

Consider $t = T$, and substitute (4.45) and (4.46) in (4.44)

$$\mathbf{P}_i = \mathbf{R}_i^{-1}\left[\mathbf{R}_1\mathbf{P}_1 + \mathbf{d}e^{j\omega_i T} - \mathbf{d}e^{j\omega_1 T}\right] \qquad i = 2, 3, \ldots, m \tag{4.47}$$

Eq.(4.47) provides an explicit relation between $\mathbf{P}_1$ and $\mathbf{P}_i$, $(i = 2, \ldots, m)$. It means that if we prepare $\mathbf{P}_1$ by integration, there is no need for more integration to compute $\mathbf{P}_2$, $\mathbf{P}_3$, ..., $\mathbf{P}_M$. Each $\mathbf{P}_i$ can be calculated by the solution of (4.47).

Taking the derivative of (4.47) with respect to the parameter h, and the input frequency ω_i gives

$$\frac{\partial \mathbf{P}_i}{\partial h} = \mathbf{R}_i^{-1}\left[\frac{\partial \mathbf{R}_1}{\partial h}\mathbf{P}_1 + \mathbf{R}_1\frac{\partial \mathbf{P}_1}{\partial h} - \frac{\partial \mathbf{R}_i}{\partial h}\mathbf{P}_i\right] \qquad i = 2, 3, \ldots, m \tag{4.48}$$

$$\frac{\partial \mathbf{P}_i}{\partial \omega_i} = j\,\mathbf{R}_i^{-1}\left[-\mathbf{C}_i\mathbf{P}_i + \mathbf{d}\,T\,e^{j\omega_i T}\right] \qquad i = 2,3,\ldots,m \quad (4.49)$$

$$\frac{\partial^2 \mathbf{P}_i}{\partial h \partial \omega_i} = -\mathbf{R}_i^{-1}\left[j\,\frac{\partial \mathbf{C}_i}{\partial h}\mathbf{P}_i + \frac{\partial \mathbf{R}_i}{\partial h}\frac{\partial \mathbf{P}_i}{\partial \omega} + j\,\mathbf{C}_i\frac{\partial \mathbf{P}_i}{\partial h}\right] \quad i = 2,3,\ldots,m \quad (4.50)$$

Again, based on (4.48) to (4.50), if we know the sensitivity vectors at the input frequency ω_1, they can be computed at other frequencies ω_i by the solution of some linear systems, and without any further integration.

Chapter 5

Switched-Current Circuits

Switched-Current (SI) [47, 48] is a relatively new analog sampled-data technique that promises to overcome the problems associated with Switched-Capacitor (SC) circuits. SI circuits use MOSFET gate capacitance as the storage element to provide analog memory capability. They do not require linear floating capacitors, and can be integrated into a standard digital CMOS process. This enables digital IC manufacturers to implement both digital and analog circuits on the same chip with the existing low-cost CMOS processes. The fact that SI circuits can be designed exclusively with MOS transistors makes the chip area 30% less than that of similar SC implementation [49].

SI circuits can operate with low power supply because of the small voltage swings associated with the low-impedance nodes. Another key performance feature of SI circuits is their inherent wide bandwidth capability. Since additional capacitors and high impedance nodes do not exist in an SI circuit, its bandwidth can approach the MOSFET transition frequency, f_T. An SI bandpass Delta-Sigma modulator which operates at 10 MHz with a clock frequency of 40 MHz was reported in [50].

A brief review of SI circuits is given in section 1. While studying the switched-current technique, I proposed some nonfiltering applications such as a current-controlled oscillator, a modulator, a rectifier, and a Delta-Sigma modulator. Section 2 presents the building blocks of these applications [51]. The circuit level implementation of these building blocks were performed with the help of the current-mode circuits proposed in [52–57]. These circuits are explained in Appendix-C.

5.1 A Review of Switched-Current Circuits

This section presents the concepts of SI circuits. First, current-mirror and current track-and-hold are explained as the basic building blocks of SI circuits. Next, a brief survey of SI filtering applications is given.

5.1.1 Current Mirror

The current mirror circuit in Fig.5.1 consists of two transistors M_1 and M_2 with the aspect ratios of $\frac{W_1}{L_1}$ and $\frac{W_2}{L_2}$, respectively. L and W are the transistor effective length and width. Define

$$K = \frac{W_2/L_2}{W_1/L_1}. \tag{5.1}$$

There are two biasing current sources in Fig.5.1, I and KI, and the transistor M_1 is diode connected. M_1 and M_2 are biased in the saturation region ($V_{GS} > V_T$ and $V_{DS} \geq (V_{GS} - V_T)$). Neglecting channel-length modulation effects, the following relationship is considered between the drain current i_{DS}, and the gate-source voltage v_{GS}:

$$i_{DS} = \frac{K'}{2}\frac{W}{L}(v_{GS} - V_T)^2, \tag{5.2}$$

where K' is the device transconductance. If we apply two small-signal currents i_1

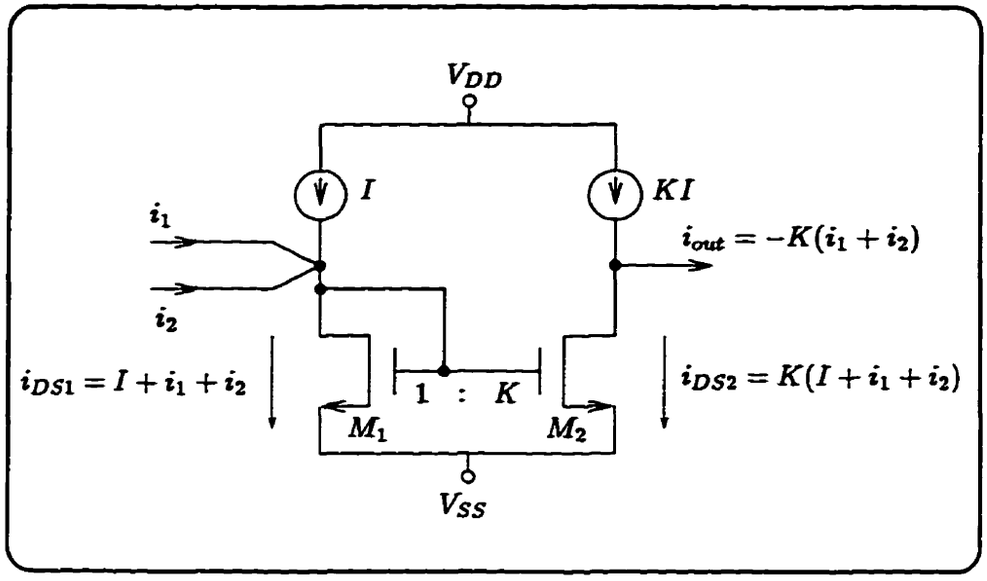


Figure 5.1: A simple current mirror.

and i_2 at the input, the current flowing into the drain of M_1 is the sum of the bias and signal currents :

$$i_{DS1} = I + i_1 + i_2. \tag{5.3}$$

This current generates a corresponding voltage on the gate-source capacitance of M_1. Since the gates and sources of M_1 and M_2 are connected to each other, $v_{GS1} = v_{GS2}$, and because of the different aspect ratios of the transistors, a current equal to $K\, i_{DS1}$ will flow into the drain of M_2 :

$$i_{DS2} = K i_{DS1} = K(I + i_1 + i_2). \tag{5.4}$$

Applying KCL at the drain of M_2 :

$$i_{out} = KI - i_{DS2} = -K(i_1 + i_2). \tag{5.5}$$

The output of the current mirror, i_{out}, is an inverted sum of the input currents scaled by a factor of K. Thus, the current mirror performs the operations of signal

inversion, scaling and summation. In this basic building block, the summation of input signals is accomplished without requiring any additional circuitry, but generation of more than one output current requires adding branches identical to the M_2 branch. These two characteristics are the dual of the voltage-mode system.

5.1.2 Current Track-and-Hold Circuit

The current track-and-hold (T/H) or memory circuit [47] is constructed by placing a switch M_S between the gates of the mirror transistors M_1 and M_2 (Fig. 5.2). When ON, the switch shorts the gates of the two mirror transistors. In this mode, the circuit functions similarly to the current mirror, and the output tracks the input signal (track mode). When the switch is turned off, the gates of M_1 and M_2 are disconnected. The gate voltage of M_1, corresponding to the value of the

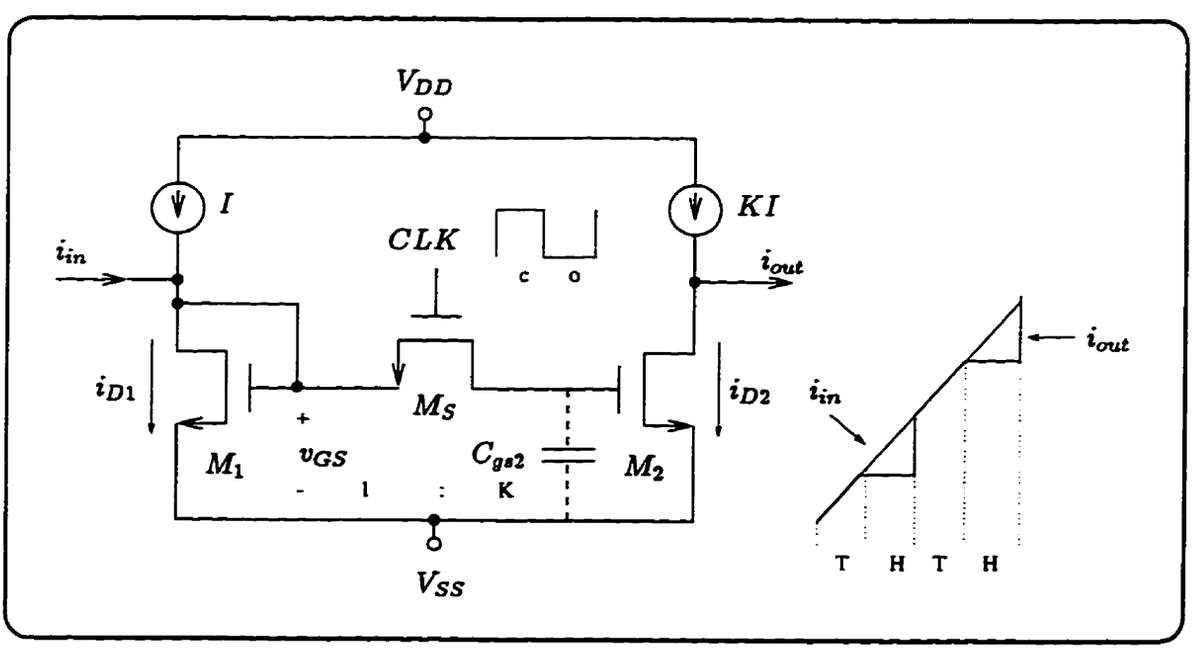


Figure 5.2: Current track-and-hold circuit.

input current at the instant the switch is opened, is sampled onto C_{gs2}. While the switch is open, v_{GS2} remains constant; consequently, the output current is held at a constant value corresponding to the input current at the instant when the switch was opened (hold mode). The output is expressed by the following discrete-time equation

$$i_{out}(n) = -K\, i_{in}(nT - \frac{T}{2}), \tag{5.6}$$

where T is the clock period, and we assumed that the clock has a 50% duty cycle. Taking the z transform of (5.6), the expression for the output current is:

$$I_{out}(z) = -Kz^{-1/2}\ \ I_{in}(z). \tag{5.7}$$

The dynamic current mirror or current copier [58,59] is another current memory circuit (Fig. 5.3). This circuit is controlled by two-phase non-overlapping clocks. When ϕ_1 is active and ϕ_2 is inactive, M_1 is diode-connected and v_{GS1} tracks the total input current $I_1 + i_{in}$. To configure the circuit as a hold amplifier, ϕ_2 is active and ϕ_1 is inactive. The voltage corresponding to the input current, just before S_3 is opened, is held on C_{gs1} and, with S_2 closed, the held signal current is sensed at the output. In a dynamic current mirror, there are no errors due to transistor mismatches (a difficulty associated with the current T/H), but only one copy of the output current is produced. In addition, while the current T/H performs mirroring in space, the dynamic current mirror performs the mirroring in time.

5.1.3 Switched-Current Filters

The most common application of switched-current circuits, like switched-capacitor ones, is frequency domain filtering. Some design principles, building blocks, and

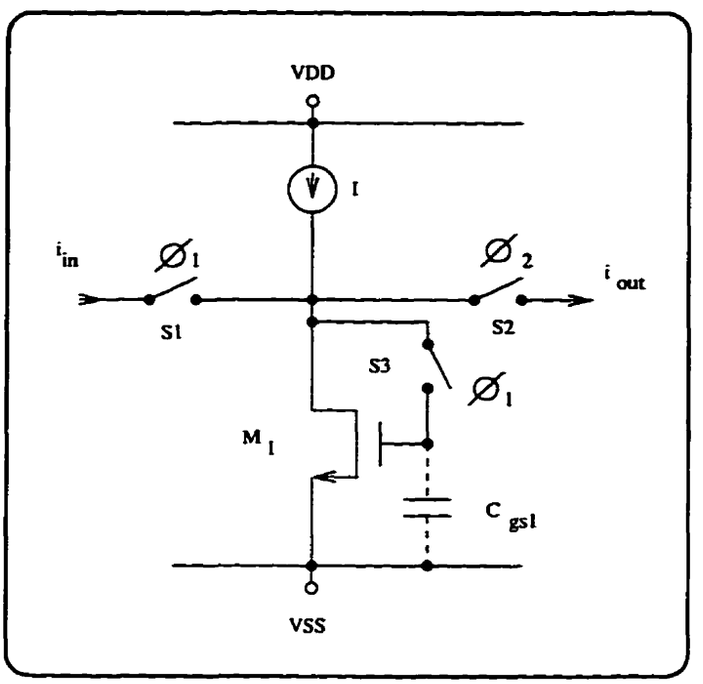


Figure 5.3: Dynamic current mirror (current copier).

actual circuits of such filters have been proposed [60–66]. Since the SI integrator is shown to be directly analogous to the SC integrator, all the synthesis techniques developed for the design of SC filters can be applied to synthesize SI filters. In addition, it has been shown that the signal flow graph (SFG) for a multiple-input SC filter is equivalent to the transpose of the SFG of a multiple-output SI filter [64]. In other words, they are inter-reciprocal. As a consequence of their inter-reciprocity, they will also posses identical component sensitivities. This suggests that the transformation of low sensitivity SC filters will lead to low sensitivity SI filters.

The track-and-hold circuit (Fig 5.2) performs four essential operations required for signal processing: signal inversion, summation, scaling, and time delay. Using these operations, we implement the SI integrator as one of the basic building blocks of SI filters. One configuration of the SI integrator is composed of two cascaded

current T/H circuits (Fig. 5.4). The switches are controlled by two-phase non-overlapping clocks. The output of the second T/H is connected to the input of the first. Breaking the feedback loop at the output, and assuming A1=A2=1, the expression for the output current at the drain of M_4 is :

$$i_f(z) = (i_f(z) + i_1(z))z^{-1}. \tag{5.8}$$

Rearranging this expression yields :

$$i_f(z) = i_1 \frac{z^{-1}}{1 - z^{-1}} \tag{5.9}$$

The integrator output is amplified by scaling the aspect ratio of M_5 to M_4:

$$i_{out}(z) = K\ i_f(z) = K\frac{i_1 z^{-1}}{1 - z^{-1}}, \tag{5.10}$$

where $K = \frac{(W/L)_5}{(W/L)_4}$ is the integrator scale factor. The $(1 - z^{-1})$ term in the denominator represents discrete-time integration. In fact, the expression for $i_f(z)$ is

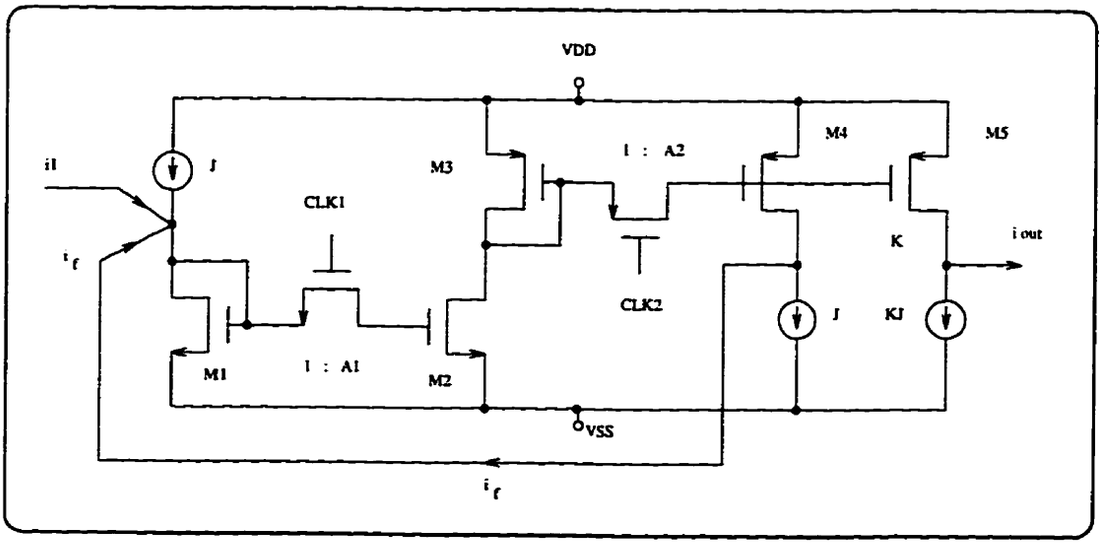


Figure 5.4: An SI integrator.

a Forward Euler transformation of a non-inverting integrator from the s-domain to the z-domain ($s \longrightarrow \frac{1}{T}\frac{1-z^{-1}}{z^{-1}}$).

The SI integrator current output is directly analogous to the SC integrator voltage output (Fig. 5.5), where

$$V_{out}(z) = \frac{C_U}{C_I}\frac{v_1 z^{-1}}{(1-z^{-1})}. \tag{5.11}$$

In the SC integrator, the capacitor ratio $\frac{C_U}{C_I}$ determines the integrator scale factor, while in the SI integrator, the integrator scale factor is determined by the transistor aspect ratio, K. There are other configurations of SI integrators corresponding to the Backward Euler ($s \longrightarrow \frac{1}{T}(1-z^{-1})$) and the Bilinear ($s \longrightarrow \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}$) mapping from the s-domain to the z-domain.

Table 5.1 shows some fabricated SI filters. The numbers in the third column refer to the papers that report the filter performances, and f_{cut} denotes the filter cutoff frequency.

The interested reader is referred to [67–74] for more investigation on the practical considerations in SI circuit design, such as device mismatch, finite output

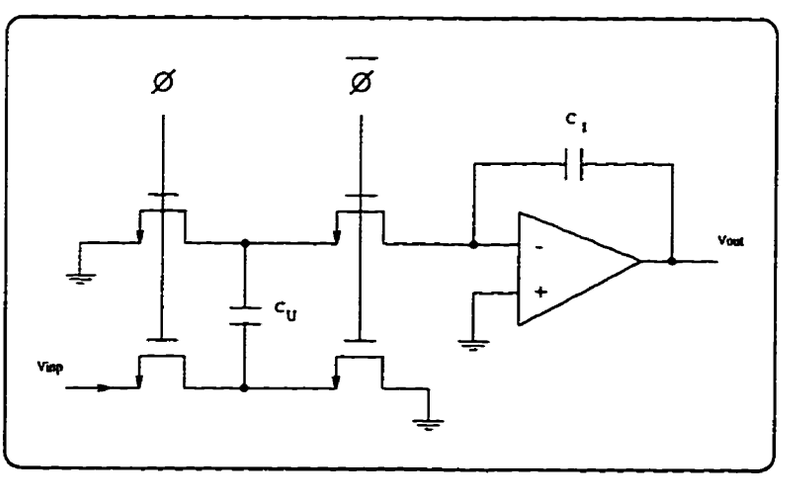


Figure 5.5: An SC integrator, the dual of an SI integrator.

impedance, bandwidth, and noise.

Filter	**Comments**	**Ref.**
Biquad filter	LPF , BPF , HPF	[61]
3^{rd} order Elliptic filter	$f_{cut} = 3\text{kHz}$, $f_{clk} = 128\text{kHz}$	[60,75]
5^{th} order Chebyshev filter	$f_{cut} = 5\text{kHz}$, $f_{clk} = 128\text{kHz}$	[60,75]
6^{th} order Chebyshev filter	$f_{cut} = 1\text{MHz}$, $f_{clk} = 10\text{MHz}$	[48]
FIR filter	$f_{cut} = 1\text{MHz}$, $f_{clk} = 10\text{MHz}$	[76]
Digitally programmable	Variable Gain,f_C, and Q	[77]

Table 5.1: Reported SI filters.

5.2 Non-Filtering Applications of Switched-Current Circuits

Although filtering applications have received most of the attention among SI circuits, there are other analog signal processing tasks that can be performed with the same fabrication technology and circuit elements as those used in switched-current filters. In this section, we present some non-filtering applications of SI circuits [51]. They include a current-controlled oscillator, a modulator and a full-wave rectifier. We also propose a switched-current oversampling Delta-Sigma A/D converter. All proposed circuits have been implemented using the circuits in Appendix-C, and simulated at device level using SPICE.

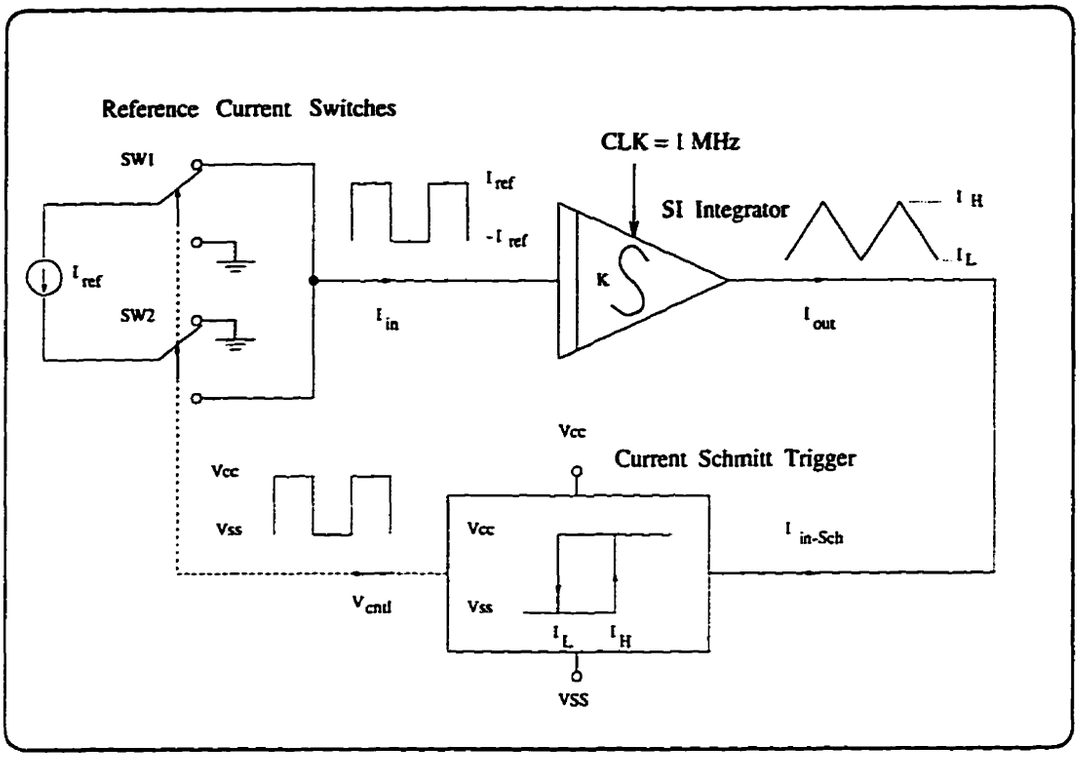


Figure 5.6: Block diagram of an SI current-controlled oscillator.

5.2.1 Current-Controlled Oscillator (CCO)

To provide various frequencies in a circuit, we may use digital scalers to divide the frequency of a master clock signal. This method results in good frequency stability since the master clock is usually crystal controlled. However, the frequencies are restricted to subharmonics of the master-clock frequency, and they can not be readily changed by a control current (or voltage), as required in some applications like phase-locked loops (PLL).

The block diagram of an oscillator which does not require a master clock and can be controlled by a reference current is shown in Fig.5.6. Assume switches SW1 and SW2 are in the position shown on the figure. They apply a constant current,

I_{ref}, to the SI integrator. The integrator creates a ramp output current :

$$I_{out} = \frac{K}{T_s} I_{ref} \quad t \tag{5.12}$$

where K is the integrator amplification factor, T_s is the switching period ($T_s = \frac{1}{f_s}$), and t is time. If this ramp signal goes through a current Schmitt trigger (with threshold I_H and I_L), the output of the trigger will change when the input ramp current approaches I_H. Since the reference current switches are controlled by the output of the Schmitt trigger, their states will change, producing another constant current, $-I_{ref}$, at the input of the integrator. Following the circuit operation continuously, we get a triangular current at the output of the integrator. The oscillation frequency is:

$$f_o = \frac{K}{T_s} \frac{1}{2(I_H - I_L)} I_{ref} \tag{5.13}$$

The oscillation frequency depends linearly on I_{ref}, making the circuit a current-controlled oscillator. In addition, the oscillation frequency is determined by the hysteresis of the Schmitt trigger and by the parameters of the SI integrator, like the switching frequency and the transistors aspect ratio. These facts can be utilized to implement a programmable oscillator. In addition, the oscillator provides three types of signals: a triangular current wave at the output of the SI integrator, a square current wave at the input of the integrator, and a square voltage wave at the output of the Schmitt trigger.

To implement the current-controlled oscillator, we need a reference current switch, an integrator, and a current Schmitt trigger. Appendix-C describes these circuits.

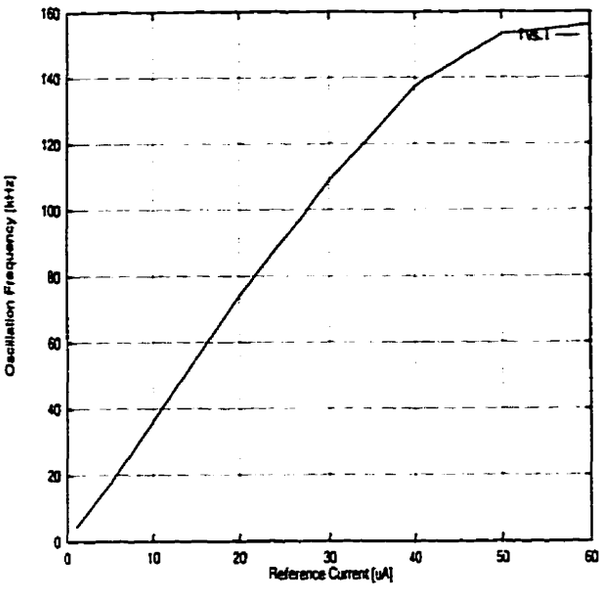


Figure 5.7: Oscillation frequency vs. reference current.

Simulation Results

The proposed SI current-controlled oscillator was simulated with $1.2\mu m$ technology parameters. The hysteresis width for the Schmitt trigger was chosen to be $50\mu A$ to prevent saturation of the integrator. I_{ref} can be changed between $5\mu A$ to $30\mu A$, since a current less than $5\mu A$ can not be switched accurately, and a current of more than $30\mu A$ may saturate the integrator. The switching frequency for the integrator was set to 1 MHz, and its amplification factor was considered as $\frac{(W/L)_{N7}}{W/L)_{N5}} = \frac{(W/L)_{N10}}{W/L)_{N8}} = 0.4$.

Fig.5.7 shows the plot of the I_{ref} versus oscillation frequency. The plot shows that for $5\mu A \leq I_{ref} \leq 30\mu A$ the oscillation frequency is linearly proportional to I_{ref}. For currents more than $30\mu A$, the integrator is saturated causing nonlinearity in the curve. For currents less than $5\mu A$, the reference current switch can not be turned on completely.

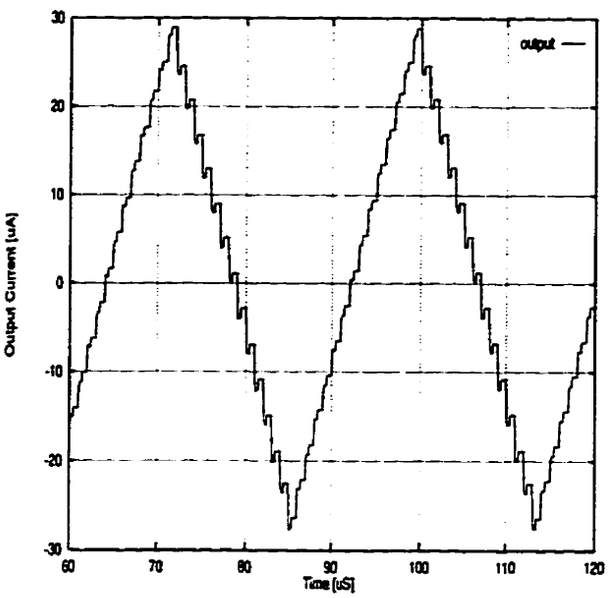


Figure 5.8: Oscillator output for $I_{ref} = 10\mu A$.

Fig.5.8 shows the output waveforms corresponding to $I_{ref} = 10\mu A$ and $f_{clock} = 1MHz$. The oscillation frequency is about 37 kHz, which corresponds to the value obtained by (5.13) :

$$f_o = \frac{K}{T_s}\frac{1}{2(I_H - I_L)}I_{ref} = \frac{0.4}{10^{-6}}\frac{1}{2 \times 50 \times 10^{-6}} \times (10 \times 10^{-6}) = 40kHz$$

5.2.2 Modulator

A modulator is a nonlinear circuit which produces replicas of the spectrum of the input signal, shifted along the frequency axis. To shift the spectrum $M(\omega)$ of the input signal $m(t)$ by the amount of ω_{ca}, the input signal is multiplied by the carrier signal $x(t) = cos(\omega_{ca}t)$

$$y(t) = m(t)\ cos(\omega_{ca}t).$$

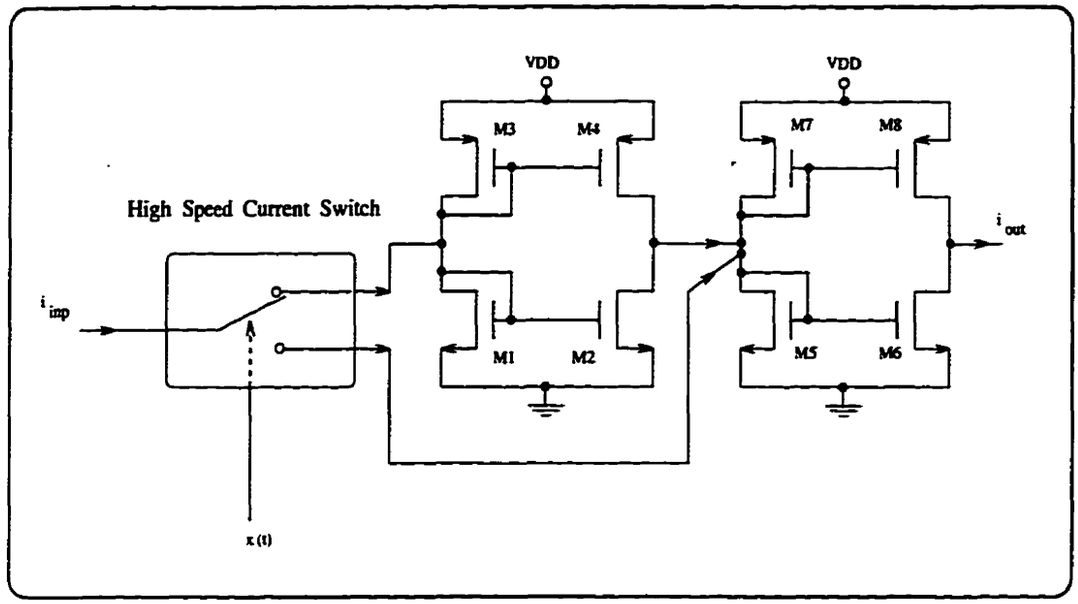


Figure 5.9: Switched-current modulator circuit.

This time domain multiplication shifts the frequency spectrum by $\pm\omega_{ca}$:

$$Y(\omega) = \frac{1}{2}M(\omega + \omega_{ca}) + \frac{1}{2}M(\omega - \omega_{ca}) \tag{5.14}$$

Generally, the carrier signal can be any periodic signal. The square wave is a periodic carrier signal which is readily generated by SI circuitry. Its Fourier series expansion is

$$x(t) = \sum_{n=-\infty}^{+\infty} a_{2n+1} e^{j(2n+1)\omega_{ca}t}$$

where the coefficients a_{2n+1} decrease as 1/(2n+1). Due to the square wave carrier, only odd-indexed side bands are created at $\omega_{ca}, 3\omega_{ca}, 5\omega_{ca}, \ldots$ The modulated signal can be extracted by filtering the spectrum around ω_{ca}.

A method to perform modulation with a square wave carrier is to switch the polarity of the input signal periodically. Fig.5.9 shows the proposed circuit. The

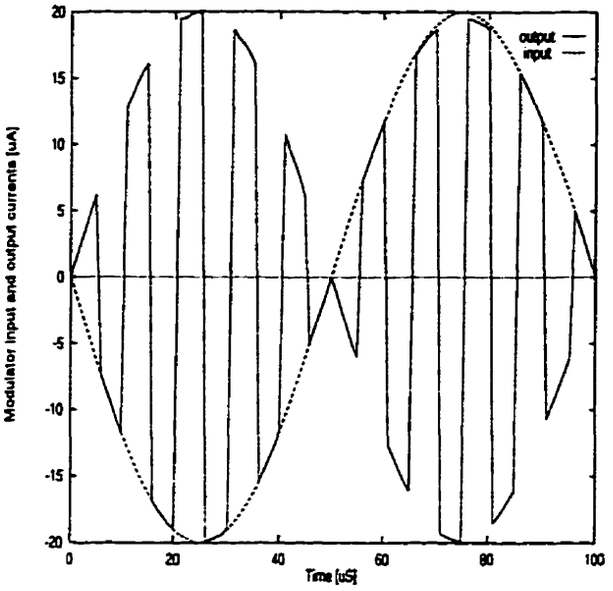


Figure 5.10: Modulator output .

first current mirror (M_1, M_2, M_3, M_4) inverts the input signal and the second current mirror (M_5, M_6, M_7, M_8) performs summation. In each half period of the square wave, the input signal or its inverse, is added in the second current mirror, and constructs the final modulated signal. To switch the input signal to two different paths we use the high-speed current switch explained in Appendix-C. To avoid aliasing, the input spectrum should be band limited and the modulating frequency must be at least twice the maximum signal frequency. A fully balanced current mirror [53,54] can be used instead of a simple current mirror to reduce the nonideal characteristics and get more precise amplification. Fig.5.10 shows the modulator output waveform obtained from SPICE simulation.

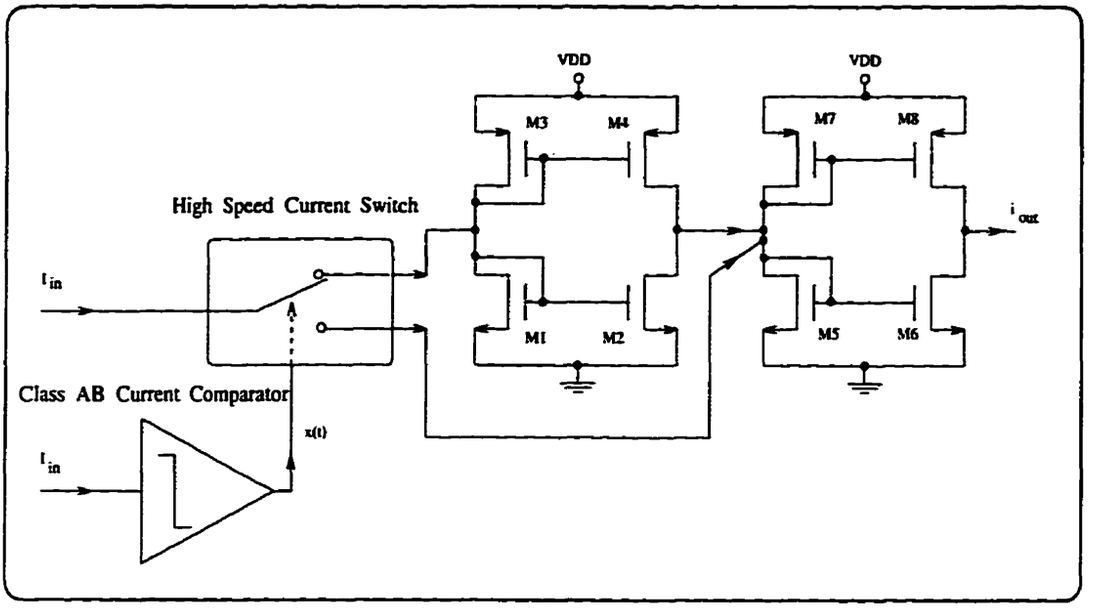


Figure 5.11: Full-wave current-mode rectifier.

5.2.3 Full-Wave Rectifier

A full-wave current rectifier converts an input signal $I_{in}(t)$ to its absolute value $\mid I_{in}(t) \mid$. By adding a current comparator to the modulator explained in the previous section, a SI full-wave rectifier is implemented. When the input signal is positive, the output of the comparator is "high" letting the input signal go directly to the output, and if the input is negative, the comparator output is "low" letting the input become inverted. Thus, $I_{out}(t)$ is proportional to $\mid I_{in}(t) \mid$.

Fig.5.11 shows the proposed circuit of an SI full-wave rectifier. It is composed of two subcircuits: a current modulator, and a current comparator. Appendix-C explains class-B and class-AB current comparators that can be used to implement the proposed rectifier. Fig.5.12 shows the simulation results. The DC offset and AC gain error are two types of errors associated with the output signal. Using fully balanced current mirrors reduces these errors.

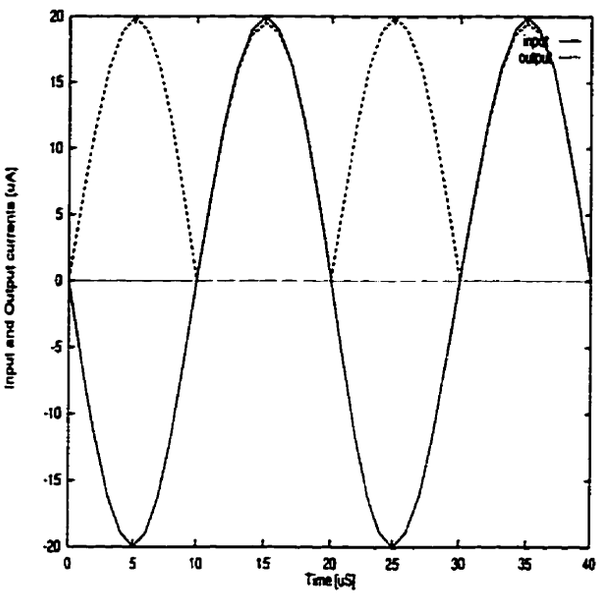


Figure 5.12: Input and output currents of full-wave rectifier.

5.2.4 Oversampled Delta-Sigma Modulator

The block diagram of the proposed second-order Delta-Sigma modulator designed with the SI technique is shown in Figure 5.13. The input is a current signal, and the output is a voltage signal that can be applied to a digital filter. The modulator consists of two integrators, a comparator with a latch, and two D/A converters.

Fully-differential configurations are used for all blocks. The differential topology offers increased dynamic range, increased rejection of noises coming from the power supply and the digital circuit on the chip, and the first-order cancellation of clock feed through effects, resulting in higher accuracy. Summation is simply performed by connecting the outputs of the DACs to the inputs of the integrators.

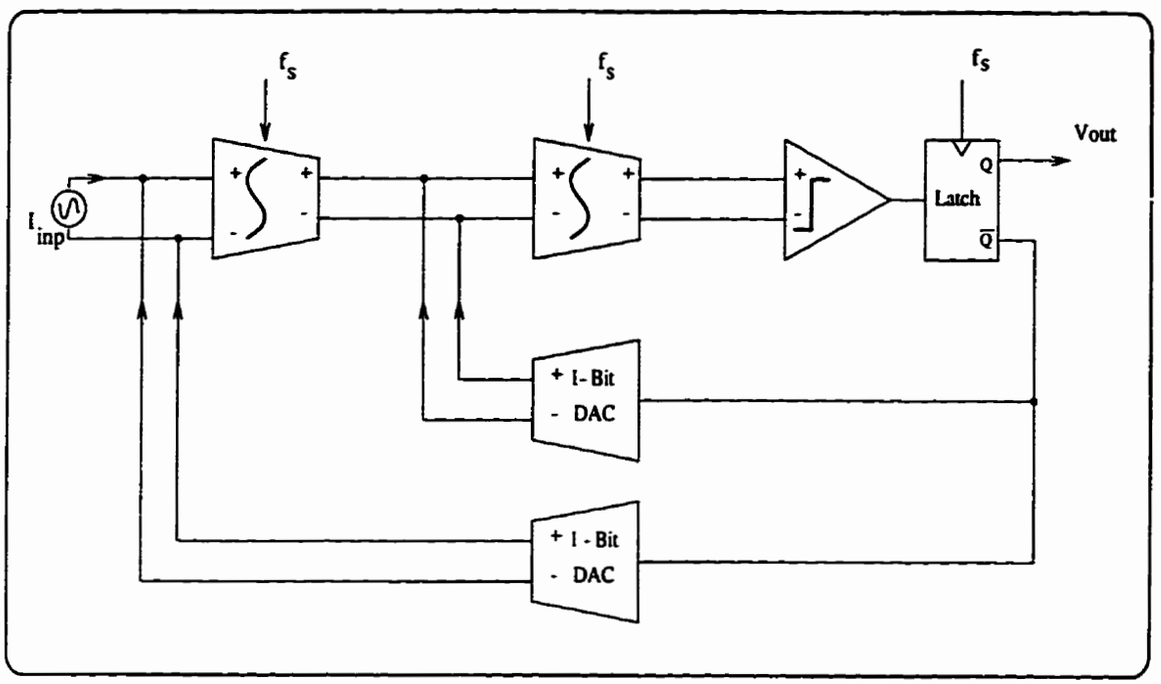


Figure 5.13: Block diagram of the switched-current Delta-Sigma modulator.

One-Bit Quantizer Design

A current comparator can be used as a one-bit quantizer. Neither sensitivity nor offset considerations are important in the design of the comparator in a second-order DSM [78]. Therefore, a simple regenerative current comparator without preamplification or offset cancellation, such as described in Appendix-C, satisfies the comparator requirements. A D-type flipflop, activated on the rising edge of the clock, is used to latch the output of the comparator and provide a glitch-free output.

Digital-to-Analog Converter Design

Two D/A converters are required to convert the one-bit digital output of the modulator back into a current signal to be applied as inputs to the SI integrators. A

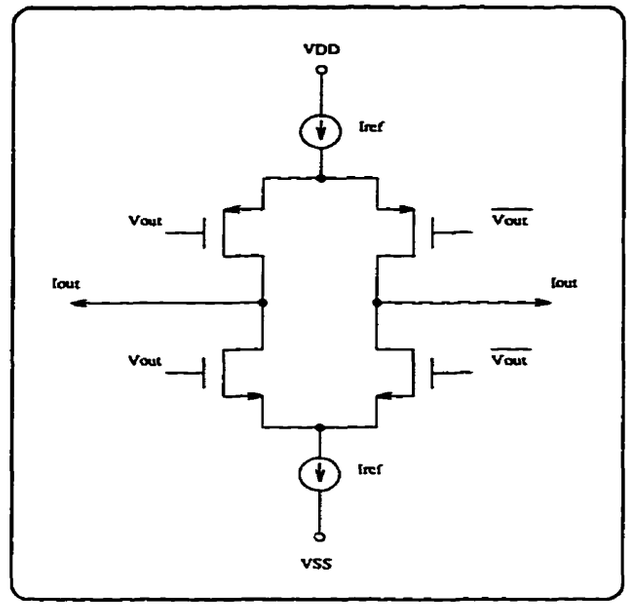


Figure 5.14: A differential switch as 1-bit D/A converter .

fully differential high-speed switch, such as that in Appendix-C, can be used to switch the reference current, I_{ref}, into the integrators. To get two directions of reference current, two differential pair switches are connected together as shown in Fig.5.14. The modulator output, $V_{out}(t)$, determines the switch positions through control circuitry.

Simulation Results

The overall circuit of the Delta-Sigma modulator is shown in Fig.5.15. The SI integrators, current comparator, and D/A converters shown in this figure are explained in Appendix-C. The circuit was simulated using HSPICE considering $1.2\mu m$ technology parameters. The sampling frequency was 6.4 MHz and the oversampling ratio was 128. This resulted in a baseband signal limit of 25 kHz. The reference current was $I_{ref} = 30\mu A$. We considered a sinusoidal input with an amplitude

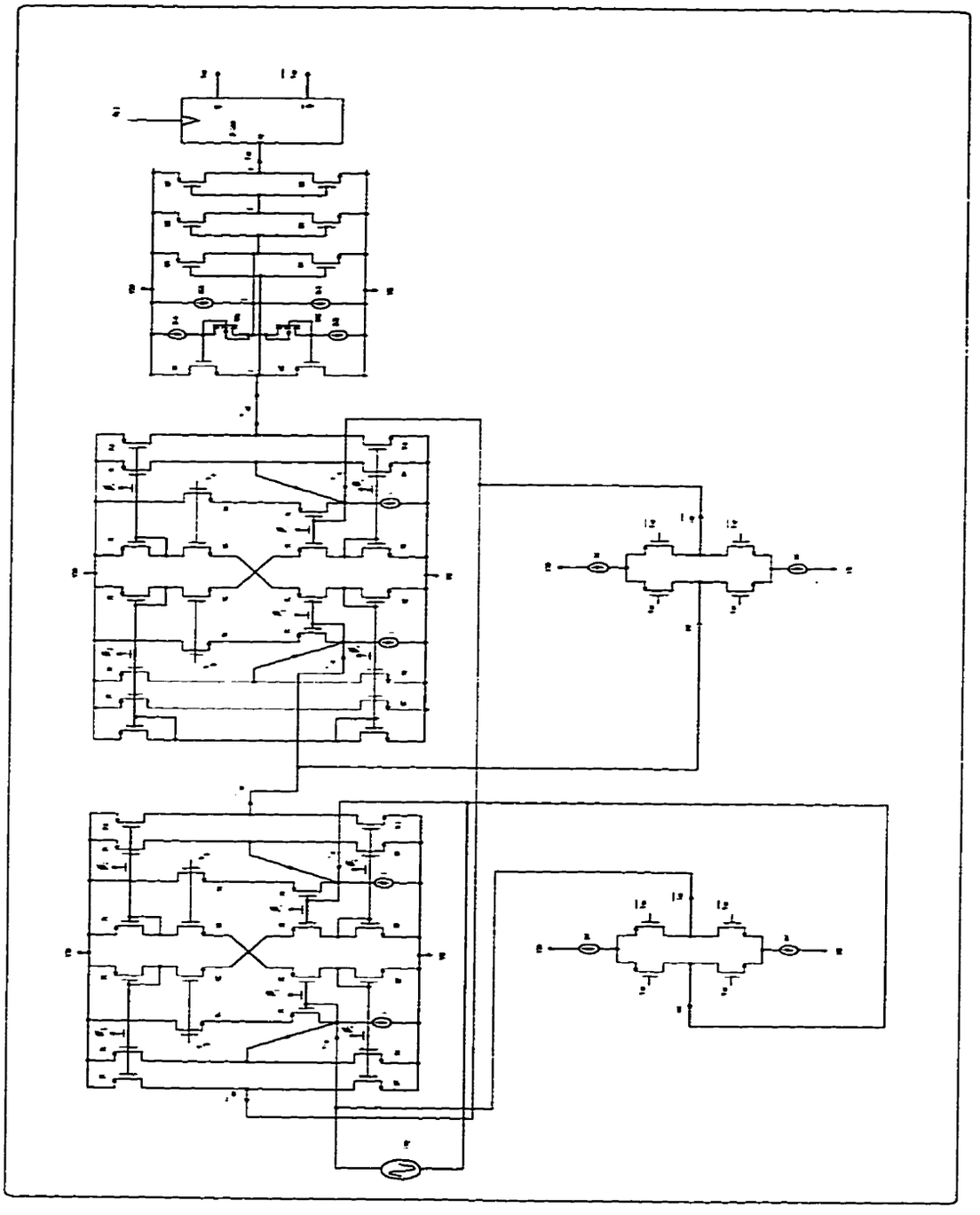


Figure 5.15: The proposed circuit of switched-current Delta-Sigma modulator.

of $20\mu A$ and a frequency of 25 kHz. The circuit was simulated for 10 periods of the input signal and the first two periods were discarded to remove the modulator transients. HSPICE took about 90 minutes CPU execution time on a SPARC-10 workstation to simulate the circuit.

With 8 periods of input signal, consisting of 2048 equally spaced samples, we determine the spectrum of the output by taking a 2048-point FFT. Fig.5.16 shows the spectrum of the output signal. The signal-to-noise ratio is about 60 dB, equivalent to 9.5-bits of linear resolution. Table 5.2 shows a summary of the measured results.

This work shows that the design of a Delta-Sigma modulator using the SI technique results in a fairly simple circuit which does not require linear capacitors, and is controlled by only two phases of the clock.

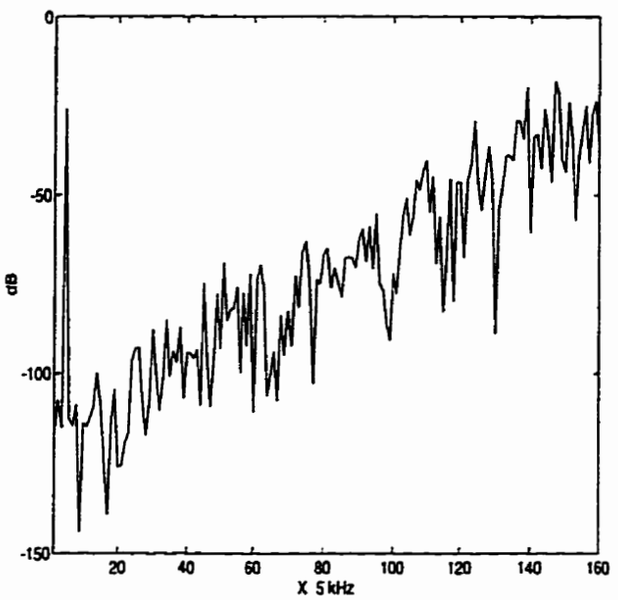


Figure 5.16: Spectrum of the output of switched-current DSM.

Clock Frequency	6.4 MHz
Signal Bandwidth	25 kHz
S(N+D)R	60 dB
Resolution	9.5 b
Power Supply	± 2.5 V
Power Dissipation	10 mW

Table 5.2: Simulated switched-current DSM performances.

Chapter 6

Harmonic Distortion in Switched-Current Circuits

Harmonic distortion in SI circuits is more severe than in SC networks. Clock feed through and mismatch in the transistor threshold voltage are two major sources of distortion in SI circuits. These sources are introduced in section 1, and their contributions to distortion are examined. In section 2, we impose both upper and lower bounds on the total harmonic distortion of a SI memory cell.

6.1 Sources of Harmonic Distortion

We study the nonlinear behavior of SI circuits on the current memory cell of Fig. 6.1 as one of the basic building blocks. The first transistor (M_1) takes the square-root of the input signal and generates the corresponding voltage on the gate-source. The second transistor (M_2) generates a current proportional to the square of this voltage. Therefore, the total characteristic of the circuit from input to output

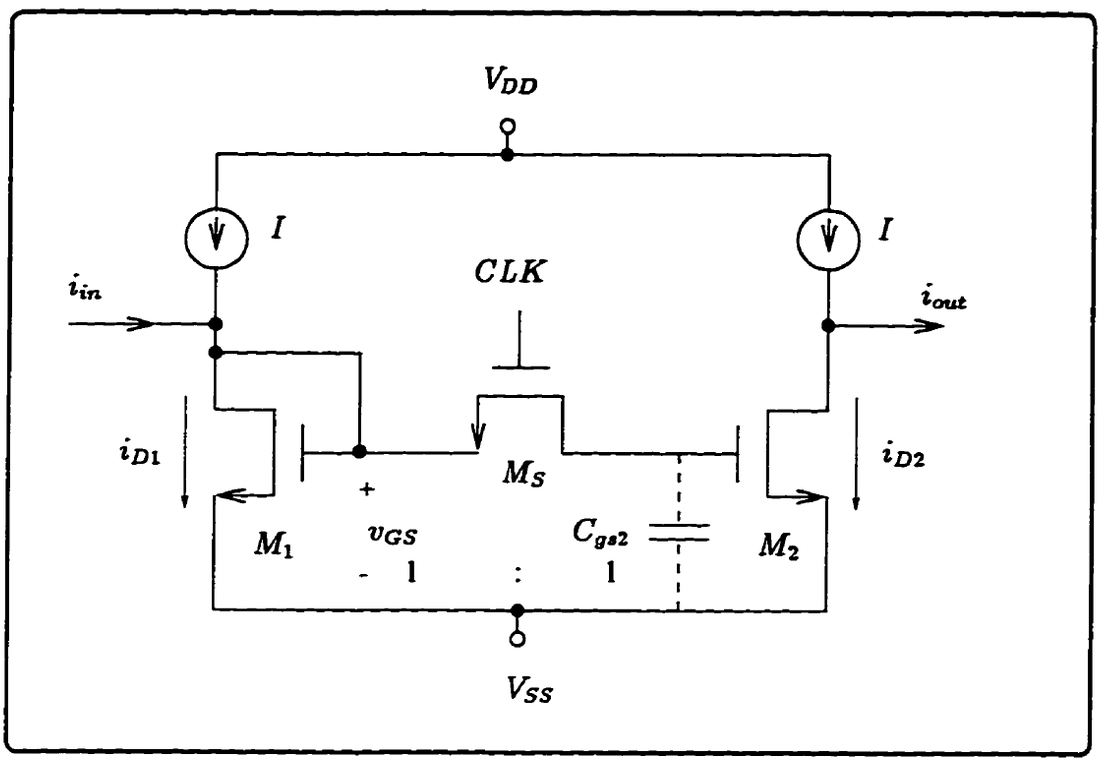


Figure 6.1: Current track-and-hold circuit.

becomes linear. This is valid under the following conditions:

1. The transistors are matched (same V_T),
2. The switch is ideal and does not generate clock feed through,
3. The signal and the switching frequencies are much lower than the circuit bandwidth.

In this section, we examine the effect of these conditions on the distortion of SI circuits.

6.1.1 V_T Mismatch

The accuracy of current-mode sampled-data systems relies on an accurate matching of the mirror transistors. Assuming M_1 and M_2 (Fig. 6.1) operate in the saturation region, the drain-to-source current is :

$$i_{DS} = \frac{K'}{2}\frac{W}{L}(v_{GS} - V_T)^2(1 + \lambda v_{DS}). \tag{6.1}$$

Mismatches in the transistor threshold voltage V_T, device aspect ratio $\frac{W}{L}$, transconductance parameter K', and the channel-length modulation λ cause errors in the output current. Only the threshold voltage mismatch produces harmonic distortion, because the other parameters are linearly related to the drain-source current, i.e. may generate DC offset and/or AC gain error.

To evaluate the contribution of V_T mismatch to the distortion, we assume M_1 and M_2 are identical except for the threshold voltage. Also we assume that the DC bias currents I are identical. The drain current in M_1 is

$$i_{D1} = i_{in} + I = \frac{\beta}{2}(v_{GS1} - V_{T1})^2 \tag{6.2}$$

where $\beta = K'\frac{W}{L}$. In the sampling mode, the gates of M_1 and M_2 are connected together, $v_{GS1} = v_{GS2}$, and

$$i_{D2} = \frac{\beta}{2}(v_{GS1} - V_{T2})^2 \tag{6.3}$$

Substituting (6.2) in (6.3) for v_{GS1} gives

$$i_{D2} = I + i_{in} + \frac{\beta}{2}\Delta V_T^2 + \beta\Delta V_T\sqrt{\frac{2}{\beta}(i_{in} + I)}$$

where $\Delta V_T = V_{T1} - V_{T2}$ is the device mismatch. The output current is

$$\begin{aligned} i_{out} &= i_{D2} - I \\ &= i_{in} + \frac{\beta}{2}\Delta V_T^2 + \sqrt{2\beta I}\Delta V_T\sqrt{1 + \frac{i_{in}}{I}} \end{aligned} \tag{6.4}$$

The last term in (6.4) can be expanded in a Taylor series

$$i_{out} = i_{in} + \frac{\beta}{2}\Delta V_T^2 + \sqrt{2\beta I}\Delta V_T \left(1 + \frac{1}{2}(\frac{i_{in}}{I}) - \frac{1}{8}(\frac{i_{in}}{I})^2 + \frac{1}{16}(\frac{i_{in}}{I})^3 - \cdots\right) \quad (6.5)$$

Ideally $i_{out} = i_{in}$, but the actual current in (6.5) has a DC offset I_{xtra}, and an AC error i_{xtra} as

$$\begin{aligned} I_{xtra} &= \frac{\beta}{2}\Delta V_T^2 + \sqrt{2\beta I}\Delta V_T \\ i_{xtra} &= \sqrt{\frac{\beta I}{2}}\Delta V_T \left(\frac{i_{in}}{I}\right) \end{aligned}$$

The remaining terms in (6.5), which have the exponent of 2,3, and more, generate the harmonic distortion. Considering a sinusoidal input $i_{in} = \hat{i}sin(\omega t)$, and assuming $\frac{\hat{i}}{I} \ll 1$, we get

$$\begin{aligned} THD \approx HD_2 &= \frac{1}{8}\sqrt{2\beta I}\Delta V_T \left(\frac{\hat{i}}{I}\right) \\ &= \frac{\sqrt{2\beta}}{8}\Delta V_T \left(\frac{\hat{i}}{\sqrt{I}}\right) \end{aligned} \quad (6.6)$$

Eq.(6.6) suggests the following solutions to reduce THD :

- reducing the signal peak $\hat{i}$,
- increasing the bias current I ,
- reducing β by reducing the $\frac{W}{L}$ ratio.

6.1.2 Clock Feed Through

Clock feed through (CFT) is due to to the non-ideal characteristics of the switch transistor [79-85]. Considering the structure of a MOS transistor (Fig. 6.2), two types of parasitic capacitances can be recognized: overlap capacitance, and channel

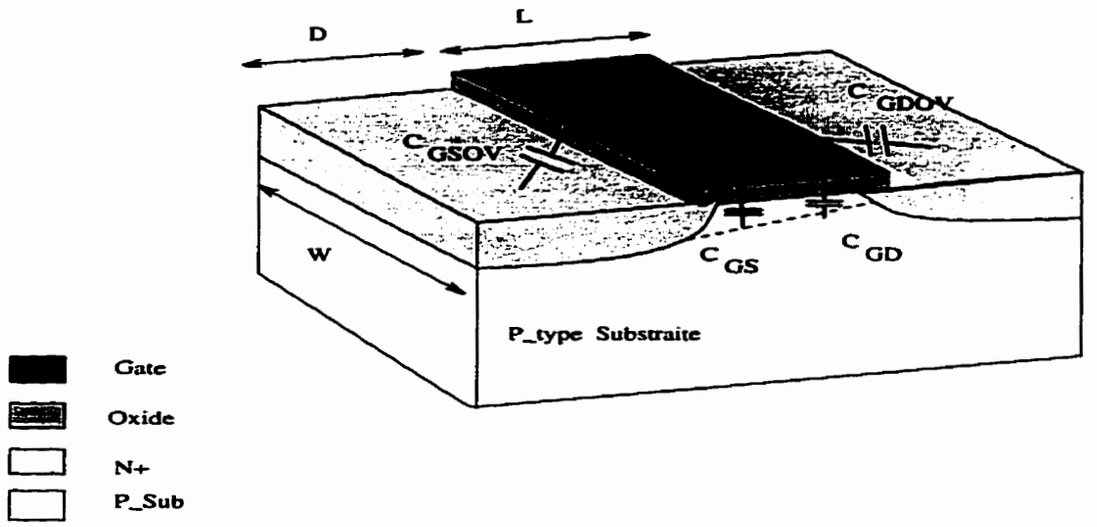


Figure 6.2: The structure of a MOS transistor with its parasitic capacitances.

capacitance. Due to over diffusion (lateral diffusion), an overlap is made between the gate and source (gate and drain). With the presence of the oxide layer between the gate and the over-diffused areas, overlap capacitance is constructed between gate and source (gate and drain) which is indicated by C_{GSOV} and C_{GDOV} in Fig. 6.2. When the voltage at the gate of the switch transistor changes rapidly, a portion of it transfers to the data-holding node through this overlap capacitance.

Channel capacitance is another type of parasitic which is constructed between the gate and the channel (depletion area). These capacitances are indicated by C_{GS} and C_{GD} in Fig. 6.2. The values of these capacitances are not constant across the channel, but normally an average value is considered. When the switch is turned off, the channel charge flows out of the drain, source, and substrate and a portion of it gets dumped to the gate capacitance of the memorizing transistor.

The total gate capacitance of the switch is [80]

$$C_{SW} = 2C_{OV} + C_H = 2C_{OV} + WLC_{OX}, \tag{6.7}$$

where C_{OV} and C_H are the overlap and the channel capacitances, respectively. Following the circuit proposed in [84, 86] the model of the transistor in the ON and OFF state can be shown as Fig. 6.3 (a),(b). When the transistor is ON, both channel and overlap capacitances exist. A voltage dependent conductance ($g = \beta(V_G - V_T)$) also appears between the source and drain. When the transistor is OFF, just the overlap capacitance exists and there is almost zero conductance between the source and gate.

The injected charge produces an error in the held drain current, corresponding to $(V_{gs} - V_T)^2$. Clock feed through effects in SI circuits are similar to those in SC circuits, which have been studied extensively in [79, 85]. The amount of injected charge is a function of the switch turn-off rate, the aspect ratio of M_s to M_2, the source-to-load capacitance ratio, the switch-to-source resistances, and the voltage at the hold node. According to [72], the CFT voltage can be expressed as :

$$V_{cft} \approx \frac{(V_H - V_L)W_{SW}[LD + \eta(L_{SW}/2)]}{W_2 L_2} \tag{6.8}$$

where LD is the lateral diffusion length, W_{SW} is the switch width, W_2 and L_2 are

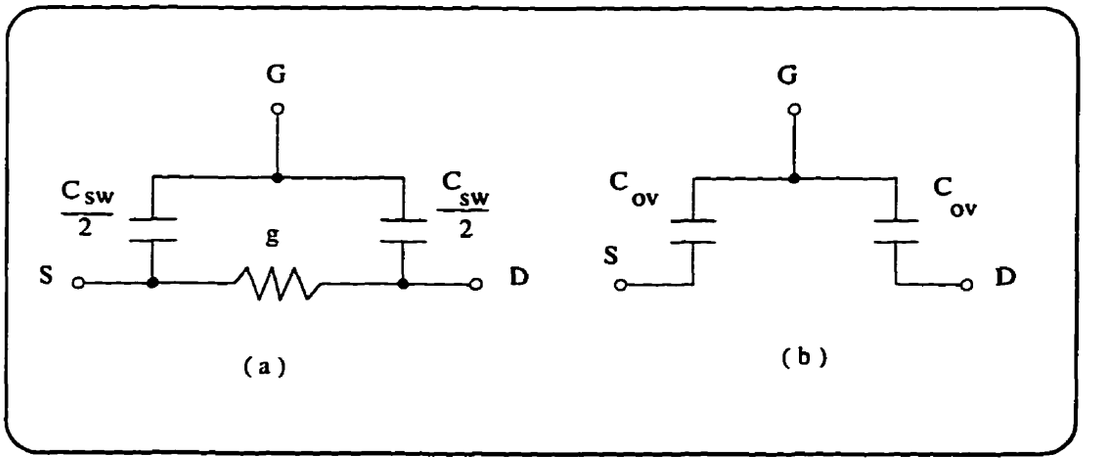


Figure 6.3: Model for switch transistor in (a) ON state, (b) OFF state

the memory transistor dimensions, and η denotes the fraction of the clock voltage swing during which the switch is ON.

The V_{cft} causes the same distortion as the threshold voltage mismatch. To calculate the CFT current in Fig.6.1, we express the total current in the drain of M_2 as:

$$i_{total} = I + i_{ac} + i_{cft}. \tag{6.9}$$

Here I is the DC bias current, i_{ac} the AC signal, and i_{cft} the CFT current. We can also write:

$$\begin{aligned} i_{total} &= \frac{K'}{2}\frac{W_2}{L_2}(V_{gs} + V_{cft} - V_T)^2 \\ &= \frac{K'}{2}\frac{W_2}{L_2}V_{cft}^2 + \frac{K'}{2}\frac{W_2}{L_2}(V_{gs} - V_T)^2 + K'\frac{W_2}{L_2}(V_{gs} - V_T)V_{cft} \end{aligned} \tag{6.10}$$

Since the second term in (6.10) is equal to $I + i_{ac}$, the remaining terms are due to the CFT voltage:

$$i_{cft} = \frac{k'}{2}\frac{W_2}{L_2}V_{cft}^2 + K'\frac{W_2}{L_2}(V_{gs} - V_T)V_{cft}. \tag{6.11}$$

Considering

$$V_{gs} - V_T = \sqrt{\frac{2(I + i_{ac})}{K'(W_2/L_2)}},$$

we write

$$i_{cft} = \frac{K'}{2}\frac{W_2}{L_2}V_{cft}^2 + V_{cft}\sqrt{2K'\frac{W_2}{L_2}(I + i_{ac})}. \tag{6.12}$$

It is clear from (6.12) that the current i_{cft} is not only a square function of V_{cft}, which generates harmonic distortion, but also varies with bias and signal currents, which results in a signal-dependent CFT. This equation can be separated into a

DC offset term and a polynomial which exhibits harmonic distortion

$$i_{cft,DC} = \frac{K'W_2}{2}V_{cft}^2 + V_{cft}[2IK'(W_2/L_2)]^{1/2}$$

$$i_{cft,AC} = V_{cft}[2IK'(W_2/L_2)]^{1/2}[\frac{(i_{ac}/I)}{2} - \frac{(i_{ac}/I)^2}{8} + \frac{(i_{ac}/I)^3}{16} - \ldots]$$

The first term in $i_{cft,AC}$ represents the AC gain error. The other terms, , with exponent of 2,3, and more, generate the harmonic distortion. Considering a sinusoidal input $i_{in} = \hat{i}sin(\omega t)$, and assuming $\frac{\hat{i}}{I} \ll 1$, we get the same equation as (6.6) for the harmonic distortion.

$$THD \approx HD_2 = \frac{1}{8}\sqrt{2IK'(W_2/L_2)}V_{cft}\left(\frac{\hat{i}}{I}\right)$$

$$= \frac{\sqrt{2K'(W_2/L_2)}}{8}V_{cft}\left(\frac{\hat{i}}{\sqrt{I}}\right)$$

6.1.3 Operating Frequency

Denote the 3dB frequency bandwidth of the current memory cell in Fig.6.1 by f_{3dB}. If the input signal frequency is comparable to f_{3dB}, the nonlinear settling-time behavior of the circuit generates distortion as explained in [87]. Also at this range of signal frequency, the switching operation transfers the high frequency components back into the circuit bandwidth which directly contributes to distortion.

6.2 Predicting Harmonic Distortion in the SI Memory Cell

The nonlinearity of SI circuits has been studied in [87] and variations in the settling behavior of the current memory cell were shown to be the major source of distortion

in SI circuits. In addition, an approximate formula for the upper bound on the total harmonic distortion (THD) was derived. In this section, we present a general expression for the THD of non-linear circuits. Using this expression, we impose both upper and lower bounds on the THD of a switched-current memory cell.

We follow similar derivations as in [87], but provide the following improvements:

- a general equality expression for the THD of non-linear circuits is introduced instead of an inequality,
- a lower bound on the THD is derived from the general expression,
- a slightly tighter upper bound (compared to that presented in [87]) is imposed on the THD.

The method is applied to some examples. The bounds predicted by our method are in agreement with SPICE simulation results.

6.2.1 THD Measurement

Denote the input signal of a SI circuit by i_{IN} and the output by i_{OUT}. The THD of the circuit can be measured by applying a pure sinusoidal signal to the input and measuring the power associated with the fundamental tone and harmonics at the output. Since the input is periodic, the output is also periodic and can be expressed in terms of its Fourier series

$$i_{OUT}(t) = \sum_{k=1}^{\infty} i_{OUT_k}(t)$$

where $i_{OUT_k}(t)$ $(k = 1, 2, \cdots, \infty)$ are harmonically related

$$i_{OUT_k}(t) = a_k cos(k\omega_0 t + \phi_k)$$

The DC component is disregarded as it does not introduce THD. We separate $i_{OUT_k}(t)$ into two components: a linear component $i_{OUT_1}(t)$ which is the fundamental component, and a deviation from the fundamental component Δi_{OUT}

$$\Delta i_{OUT} = i_{OUT} - i_{OUT_1} = \sum_{k=2}^{\infty} i_{OUT_k}(t). \tag{6.13}$$

The THD is defined as the square-root of the sum of the powers of the harmonics divided by the power of the fundamental component

$$THD = \left[\frac{\sum_{k=2}^{\infty} \| i_{OUT_k} \|^2}{\| i_{OUT_1} \|^2} \right]^{1/2} \tag{6.14}$$

where $\| x \|^2$ denotes the power associated with the signal $x(t)$

$$\| x \|^2 = \frac{1}{T} \int_0^T x^2(t) dt$$

Recalling Parseval's theorem, the power of a periodic signal is equal to the sum of the powers in it's harmonics

$$\| \Delta i_{OUT} \|^2 = \| i_{OUT} - i_{OUT_1} \|^2 = \sum_{k=2}^{\infty} \| i_{OUT_k} \|^2 . \tag{6.15}$$

Substitute (6.15) in (6.14)

$$THD = \left[\frac{\| \Delta i_{OUT} \|^2}{\| i_{OUT1} \|^2} \right]^{1/2} = \left[\frac{\| i_{OUT} - i_{OUT_1} \|^2}{\| i_{OUT_1} \|^2} \right]^{1/2} . \tag{6.16}$$

Given this definition, the THD can be computed if we know $i_{OUT} - i_{OUT_1}$ and i_{OUT_1}. Although these terms can not be usually measured, we can still impose the bounds on THD. In [87] the authors gave only an upper bound. We impose both upper and lower bounds on THD.

6.2.2 The Upper Bound on the THD

An upper bound on the THD of the current memory cell (Fig. 6.4) was given in [87]. Considering the small signal model of the cell during the memorizing phase ϕ_1

(Fig. 6.5), we define the time constant τ

$$\tau = \frac{C_{gs} + C_d}{g_m}$$

where C_{gs} denotes gate-to-source capacitance, C_d the capacitance connected to the drain node, and g_m the transconductance of the transistor. The output resistance r_o is ignored because $r_o > \frac{1}{g_m}$. The difference equation that relates the input and output current signals of the memory cell was shown to be

$$i_{OUT}(n) = -(1-\gamma(n))[i_{IN}(n-\frac{1}{2}) + i_{OUT}(n-1)] + i_{OUT}(n-1) \qquad (6.17)$$

where

$$\gamma(n) = e^{-T/(2\tau)} = e^{\frac{-T}{2(c_{gs}+c_d)}g_m}$$

represents the settling error that occurs in the nth sample, and can be written as a deviation $\Delta\gamma(n)$ from a linear component γ_{lin}

$$\gamma(n) = \gamma_{lin} + \Delta\gamma(n). \qquad (6.18)$$

From (6.13), the output signal can be written as

$$i_{OUT}(n) = i_{OUT_1}(n) + \Delta i_{OUT}(n). \qquad (6.19)$$

Substituting (6.18) and (6.19) back into (6.17)

$$\begin{aligned} i_{OUT_1}(n) \; + \; \Delta i_{OUT}(n) &= -(1-\gamma_{lin}-\Delta\gamma(n))[i_{IN}(n-1/2) \\ &+ \; i_{OUT_1}(n-1) + \Delta i_{OUT}(n-1)] + [i_{OUT_1}(n-1) + \Delta i_{OUT}(n-1)]. \end{aligned} \qquad (6.20)$$

We decompose this equation into a time-invariant linear, and a time-varying non-linear part. The linear time-invariant equation is

$$i_{OUT_1}(n) = -(1-\gamma_{lin})[i_{IN}(n-\frac{1}{2}) + i_{OUT_1}(n-1)] + i_{OUT_1}(n-1), \qquad (6.21)$$

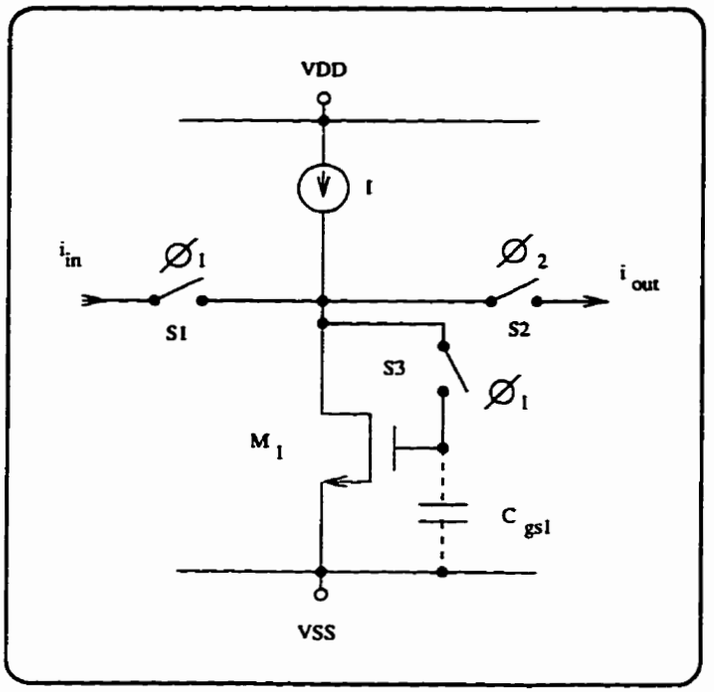


Figure 6.4: Current memory cell (current copier).

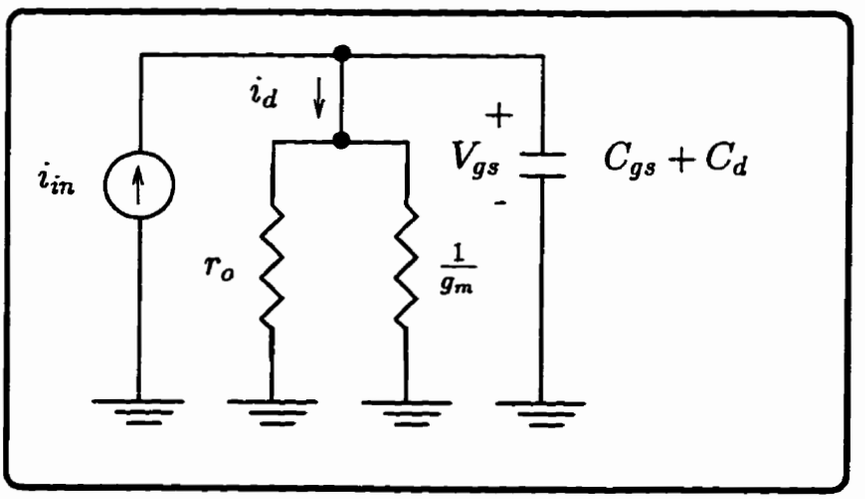


Figure 6.5: Small-signal model of the current memory cell for the clock phase ϕ_1.

and the nonlinear time-varying equation is

$$\Delta i_{OUT}(n) = (\Delta\gamma(n) + \gamma_{lin})\Delta i_{OUT}(n-1) + \Delta\gamma(n)[i_{OUT_1}(n-1) + i_{IN}(n-\frac{1}{2})]. \tag{6.22}$$

The term $\gamma(n)$ is bounded between γ_{min} and γ_{max} allowing us to observe the worst-case conditions imposed on the THD. Based on (6.16), the THD is proportional to the ratio of the power associated with Δi_{OUT} over the power associated with i_{OUT}. To find an upper bound on THD, we therefore maximize $\Delta i_{OUT}(n)$ and minimize $i_{OUT_1}(n)$. According to (6.21), $i_{OUT_1}(n)$ is smallest when γ_{lin} is largest (note that γ is less than one). This will occur when $\gamma_{lin} = \gamma_{max}$. The resulting difference equation then becomes

$$i_{OUT_1}(n) = -(1-\gamma_{max})[i_{IN}(n-\frac{1}{2}) + i_{OUT_1}(n-1)] + i_{OUT_1}(n-1).$$

Taking the z-transform of this equation and re-arranging it, we get

$$\frac{I_{OUT_1}(z)}{I_{IN}(z)} = -\frac{1-\gamma_{max}}{1-\gamma_{max}z^{-1}}\, z^{-1/2}. \tag{6.23}$$

Also, the largest $\Delta i_{OUT}(n)$ is determined from (6.22) as

$$\Delta i_{OUT}(n) = \gamma_{max}\Delta i_{OUT}(n-1) + (\Delta\gamma)_{max}[i_{OUT_1}(n-1) + i_{IN}(n-\frac{1}{2})],$$

where $(\Delta\gamma)_{max}$ is the maximum change in $\gamma(n)$ when the input signal sweeps from the minimum peak to maximum. Since the above equation is now linear and time-invariant, we can take the z-transform. Taking the z-transform, and substituting $I_{OUT_1}(z)$ from (6.23), we get

$$\frac{\Delta I_{OUT}(z)}{I_{IN}(z)} = (\Delta\gamma)_{max}\frac{(1-z^{-1})}{(1-\gamma_{max}z^{-1})^2}\, z^{-1/2}. \tag{6.24}$$

Assuming an input sine wave with the amplitude A and frequency ω_0, the power associated with i_{OUT_1} and Δi_{OUT} are calculated from the following equations:

$$\| i_{OUT_1} \|^2 = \frac{A^2}{2} \left| \frac{I_{OUT_1}(z)}{I_{IN}(z)} |_{z=e^{j\omega_0 T}} \right|^2$$

$$\| \Delta i_{OUT} \|^2 = \frac{A^2}{2} \left| \frac{\Delta I_{OUT}(z)}{I_{IN}(z)} |_{z=e^{j\omega_0 T}} \right|^2$$

where $\frac{I_{OUT_1}(z)}{I_{IN}(z)}$ and $\frac{\Delta I_{OUT}(z)}{I_{IN}(z)}$ are the transfer functions obtained from (6.23) and (6.24), and T is the switching period. Substituting the expression for $\| i_{OUT_1} \|^2$ and $\| \Delta i_{OUT} \|^2$ into (6.16), and cancelling common terms, we finally obtain

$$THD(\omega_0) \leq \left| \frac{(\Delta\gamma)_{max}}{(1-\gamma_{max})} \frac{1-e^{-j\omega_0 T}}{(1-\gamma_{max}e^{-j\omega_0 T})} \right| \tag{6.25}$$

where ω_0 is the input signal frequency. It is evident from (6.25) that the THD bound changes with the frequency of the input signal. In fact, as γ_{max} and γ_{min} are normally quite small, the THD increases with increasing input frequency because of the term $(1-e^{j\omega_0 T})$ in the numerator of (6.25).

To calculate the upper bound in (6.25), we need an estimate on $(\Delta\gamma)_{max}$. One estimate was suggested in [87] as

$$(\Delta\gamma)_{max} = \gamma_{max} - \gamma_{min},$$

which uses only the information about the extreme case of $\gamma(n)$. To find a better estimate of $(\Delta\gamma)_{max}$, we try to use all information available for $\gamma(n)$: when the circuit input signal is zero, $\gamma(n)$ is equal to γ_{nom}, and when the input signal sweeps from minimum value to maximum, $\gamma(n)$ changes from γ_{max} to γ_{min}, respectively. A better estimate of $(\Delta\gamma)_{min}$ is therefore

$$(\Delta\gamma)_{max} = Max \quad (\gamma_{nom} - \gamma_{min} \ , \ \gamma_{max} - \gamma_{nom}) \, . \tag{6.26}$$

6.2.3 The Lower Bound on the THD

We now try to find the smallest $\Delta i_{OUT}(n)$ and the largest $i_{OUT_1}(n)$ to obtain a lower bound on THD given by (6.16). Based on the difference equation (6.21), $i_{OUT_1}(n)$ is a maximum when γ_{lin} is a minimum (considering that γ is less than one). This happens when $\gamma_{lin} = \gamma_{min}$. The resulting difference equation is

$$i_{OUT_1}(n) = -(1-\gamma_{min})[i_{IN}(n-\frac{1}{2}) + i_{OUT_1}(n-1)] + i_{OUT_1}(n-1).$$

Taking the z-transform and re-arranging the equation, we get

$$\frac{I_{out1}(z)}{I_{in}(z)} = -\frac{(1-\gamma_{min})}{1-\gamma_{min}z^{-1}}\ z^{-1/2}. \tag{6.27}$$

To minimize $\Delta i_{OUT}(n)$, (6.22) is written as

$$\Delta i_{OUT}(n) = \gamma_{min}\Delta i_{OUT}(n-1) + (\Delta\gamma)_{min}[i_{OUT_1}(n-1) + i_{IN}(n-\frac{1}{2})], \tag{6.28}$$

in which $(\Delta\gamma)_{min}$ is the minimum change in $\gamma(n)$ when the input signal sweeps from peak to peak. Taking the z-transform of (6.28), and substituting (6.27) for $I_{out_1}(z)$, we get

$$\frac{\Delta I_{out}(z)}{I_{in}(z)} = (\Delta\gamma)_{min}\frac{(1-z^{-1})}{(1-\gamma_{min}z^{-1})^2}\ z^{-1/2}.$$

Following the same steps given in the calculation of the upper bound, we establish a lower bound on the THD as

$$THD(\omega_0) \geq \left|\frac{(\Delta\gamma)_{min}}{(1-\gamma_{min})}\frac{1-e^{-j\omega_0 T}}{(1-\gamma_{min}e^{-j\omega_0 T})}\right|, \tag{6.29}$$

and consider the following estimate for $(\Delta\gamma)_{min}$

$$(\Delta\gamma)_{min} = Min\ \ (\gamma_{nom}-\gamma_{min}\ ,\ \gamma_{max}-\gamma_{nom}). \tag{6.30}$$

6.2.4 Comparing the THD Bounds With Simulation Results

We choose two examples given in [87], and impose our new bounds on them .

Example-1 :

The current memory cell of Fig. 6.4 with a bias current of $20\mu A$ ($\gamma_{nom} = 0.0233$) is excited by a $16kHz$ tone having a $10\mu A$ amplitude. The drain current in the memory transistor varies between $10\mu A$ and $30\mu A$ producing a $\gamma_{min} = 0.01$ and a $\gamma_{max} = 0.071$. According to our derivations, (6.25) and (6.29), the bounds on the THD would be $0.26\% \leq THD \leq 1.082\%$. When compared to the result obtained from HSPICE analysis, i.e. 0.26%, we see that it is within the bounds predicted by (6.25) and (6.29).

Since the THD depends on the input frequency, the frequency of the input tone was changed from $1kHz$ to $64kHz$, and the THD bounds for different input frequencies are computed and plotted in Fig. 6.6. This figure also shows simulation results and the computed upper bound presented in [87]. As seen, the THD obtained from the HSPICE simulation is always within the bounds predicted by (6.25) and (6.29). The new upper bound is also improved.

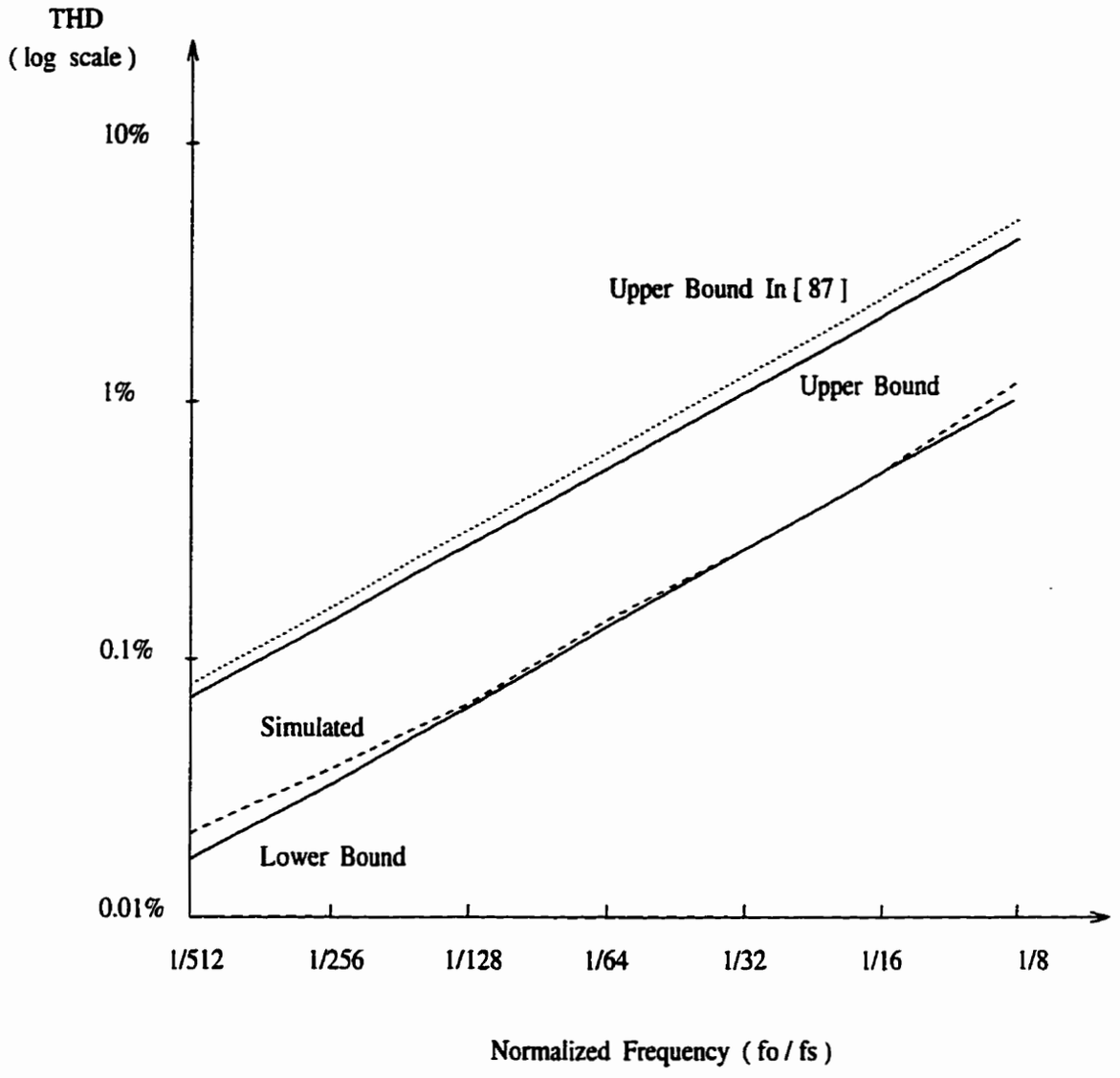


Figure 6.6: Comparing the THD computed by HSPICE with results predicted by our method.

Example-2 :

For the first generation SI memory cell in Fig. 6.1 the following parameters were

given in [87]:

$$J = 200\mu A \ , \ C_{gs2} = 1000pF$$

$$\gamma_{nom} = 0.0201 \ , \ \gamma_{min} = 0.006 \ , \ \gamma_{max} = 0.0649$$

The circuit was constructed using discrete MOSFET components (RCA 4007,RCA 4066), and clocked at the frequency of $64kHz$. A $1000pF$ capacitor was placed across the gate-source terminal of transistor T_2. A $60\mu A$ sinusoidal current signal, with a frequency varying between $125Hz$ and $15kHz$, was applied as input. The measured THD and the upper bound predicted in [87] are shown in Fig. 6.7. Also shown on this figure are the bounds predicted by our method, (6.25) and (6.29). Again, the validity of the two bounds computed by the method and the improvement on the upper bound are evident. At frequencies below $500Hz$ the switch charge injection dominates the distortion caused by settling error variation.

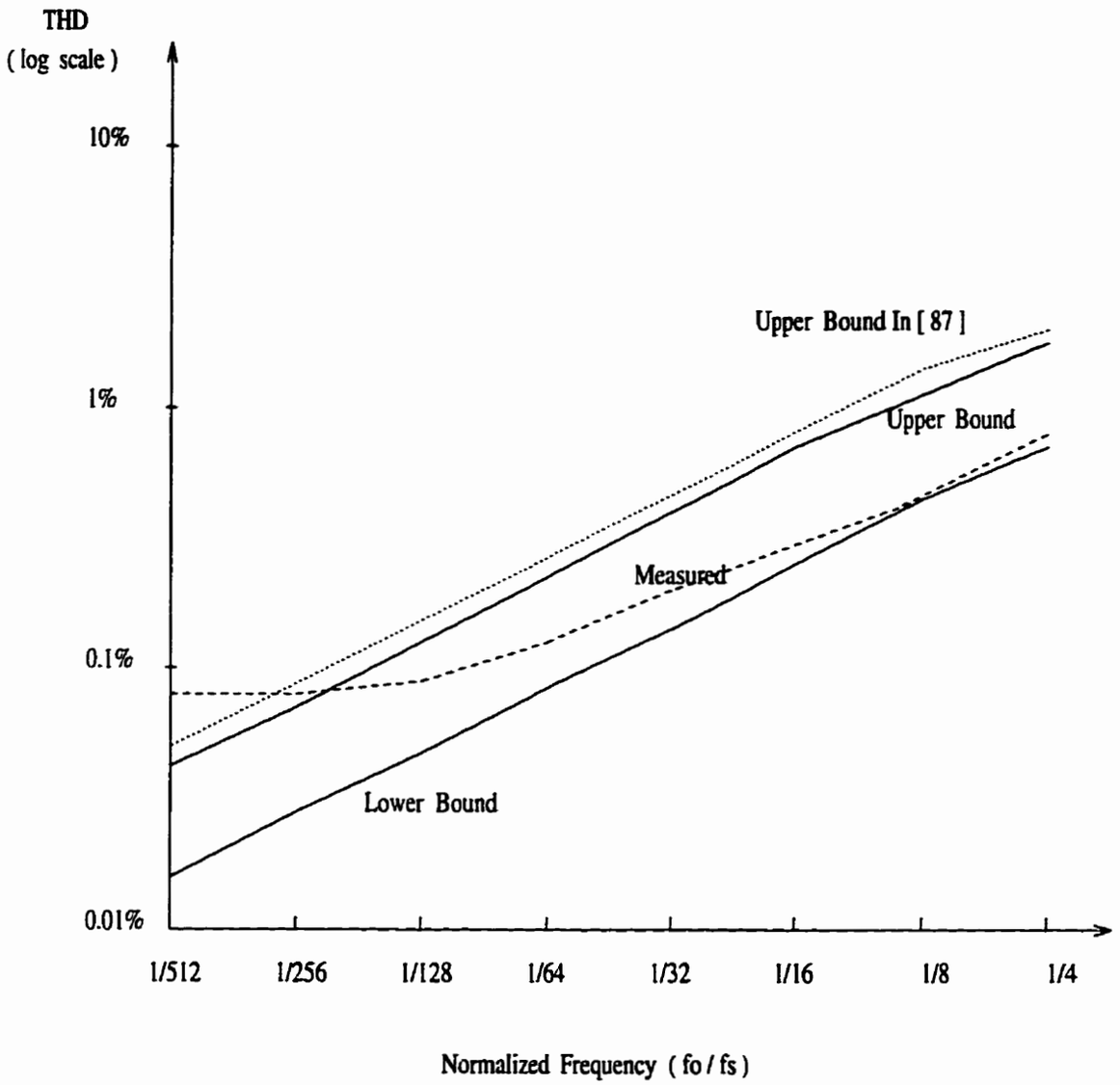


Figure 6.7: Comparing the measured THD of the circuit in Fig.6.1 with results predicted by our method.

Chapter 7

Conclusions

This thesis presents analysis of analog sampled data systems. Each chapter looked at one of the different aspects of these systems, starting with chapter 2 on time domain sensitivity of linear circuits using sampled data simulation. An obvious application of the theory developed in chapter 2 is in sensitivity analysis of Delta-Sigma modulators, which was presented in chapter 3. Another aspect of sampled data systems, the group delay and group delay sensitivity of switched linear networks, was derived in chapter 4. The switched-current circuit was introduced in chapter 5 as a new analog sampled data technique, and some nonfiltering applications of this technique were proposed. Finally, sources of distortion in switched-current were examined in chapter 6, and both upper and lower bounds were imposed on total harmonic distortion of current memory cell.

A detail list of my contribution in each area is given in section 1. Section 2 presents the proposal for continuation of the work and future research directions.

7.1 Contributions

My contributions to each area are as follows. I tried to list them in order of their significance.

1. A new method for computation of the time domain sensitivity of linear circuits was proposed. The method is accurate because no approximation is made, and efficient because some parts of the computations are performed only once, in a pre-processing step before simulation starts (Chapter 2).

2. A fast and accurate method for the calculation of group delay, and the group delay sensitivity of periodically switched linear networks was proposed. The method handles all types of linear elements, and therefore can be applied to nonideal switched-capacitor and switched-current circuits (Chapter 4).

3. We formulated the sensitivity networks for Delta-Sigma modulators, and computed the sensitivity of the output magnitude with respect to any circuit elements. The proposed method can be applied to all types/configurations of modulators as long as all elements, except the comparator, are linear. The sensitivity of the output of the clocked/unclocked comparator with respect to its input was derived (Chapter 3).

4. We proposed some non-filtering applications of switched-current circuits. They include a current-controlled oscillator, a modulator, and a full-wave rectifier. We also proposed a switched-current oversampling Delta-Sigma A/D converter (Chapter 5).

5. Sampled data simulation of linear circuits was extended to the case of inconsistent initial conditions (Chapter 2).

6. We analyzed the harmonic distortion of switched-current circuits, investigated the sources of distortion, and imposed both upper and lower bounds on the total harmonic distortion of a switched-current memory cell (Chapter 6).

7. When the input signal is sinusoidal, an efficient method was proposed for the computation of the vector $\mathbf{P}$, its derivatives with respect to frequency $\frac{\partial \mathbf{P}}{\partial \omega}$, and with respect to element values $\frac{\partial \mathbf{P}}{\partial h}$. It was shown that if these vectors are evaluated at one frequency ω_1, there is no need for further integrations to compute them at other frequencies ω_2, ω_3, etc. They can be obtained by a few matrix manipulations (Chapter 4).

8. Considering both sensitivity and adjoint networks, we explained that the adjoint method in frequency domain is equivalent to that in time domain (Chapter 2).

9. As a side work, we wrote two programs in MATLAB based on the theories developed in this thesis: SDSEN for transient analysis and the time domain sensitivity of linear circuits, and GRPSN for the computation of group delay and the group delay sensitivity of switched linear networks. Some other pieces of programs, MPgen and MPsgen, were written based on the numerical Laplace inversion, to provide the pre-processing matrices ($\mathbf{M}$, $\mathbf{P}$) and their derivatives (Chapters 2, and 3).

7.2 Future Research

During the course of this research, the following points have been detected. They are suggested here for future investigations.

1. We wrote a program for computation of group delay and the group delay sensitivity of switched linear networks, and explained how it can be used to obtain a flatter group delay of switched networks. As the next step, this program can be coupled with an optimizer to provide an integrated environment for the optimized design of switched capacitor and switched current filters.

2. Using an adjoint network for the time domain sensitivity gives the sensitivity of one output with respect to all elements at one instant of time. The following subjects can be investigated as the applications of the adjoint network :

   (a) the time domain sensitivity analysis of switched network at the end of each phase when the circuit does not reach the steady state,

   (b) transient analysis of the rise/fall time in digital circuits and transmission lines.

3. We imposed both lower and upper bounds on the harmonic distortion of the switched-current memory cell. To do this, we suggested an estimation of the settling behavior. The next step is investigating a better estimate of the settling behavior in the switched-current memory cell to tighten up the bounds we imposed on the total harmonic distortion.

4. To calculate the sensitivity of the Delta-Sigma modulator, we considered some approximations in the sensitivity of the comparator with respect to its input. The computation of the sensitivity of the Delta-Sigma modulator can be improved by considering more conditions, such as comparator hysteresis, on the signal applied to the input of the comparator.

5. We calculated the sensitivity of the magnitude of the output of Delta-Sigma modulator. The next step is computing the sensitivity of the signal-to-noise ratio

(SNR) with respect to an element, because the SNR curve conveys more information about the modulator.

6. The sampled data simulation has been shown to be a fast and accurate method of simulation for dual time systems. One application, the Delta-Sigma modulator, was presented in chapter 3. The next step could be the investigation of sampled data simulation in other dual time systems, such as phased-locked loops (PLL), and frequency modulators.

Publications Resulting From This Research

1. B. Raahemi, A. Opal, and J. Vlach, "Non-Filtering Applications of Switched-Current Circuits", *IEEE 37th Midwest Symposium on Circuits and Systems*, Vol.1, pp 169-172, August 1994.

2. B. Raahemi and A. Opal, "Time Domain Sensitivity of Linear Circuits Using Sampled Data Simulation", *Submitted to IEEE Trans. on Circuits and Systems, I- Fundamental theory and applications*, August 1996.

3. B. Raahemi and A. Opal, "Sensitivity Analysis of Delta-Sigma Modulators", *Submitted to Canadian Conference on Electrical and Computer Engineering*, August 1996.

4. B. Raahemi and A. Opal, " Group Delay and Group Delay Sensitivity of Periodically Switched Linear Networks", *Submitted to IEEE Trans. on Circuits and Systems, I- Fundamental theory and applications*, August 1996.

5. B. Raahemi and A. Opal, " Harmonic Distortion in Switched-Current Memory Cell", *Submitted to Canadian Conference on Electrical and Computer Engineering*, August 1996.

Appendix A

Computation of P, M, and Their Derivatives

Using the modified nodal formulation, let the system matrix be defined as

$$\mathbf{R}(s) = \mathbf{G} + s\mathbf{C}$$

where $\mathbf{G}$ is the conductance matrix ($m \times m$), and $\mathbf{C}$ the capacitance matrix ($m \times m$). The system equations are

$$\mathbf{R}(s)\mathbf{X}(s) = \mathbf{W}(s) + \mathbf{I} \tag{A.1}$$

where $\mathbf{W}(s)$ is the Laplace transform of the sources and $\mathbf{I}$ the vector of impulse sources $C_i V_{i0}$ or $-L_i I_{i0}$, due to initial conditions. The solution of (A.1) at the time $t = h$ is calculated by [12–15]

$$\mathbf{x}(h) = -\frac{1}{h}\sum_{i=1}^{M'} K_i \mathbf{X}(\frac{z_i}{h}) - \frac{1}{h}\sum_{i=1}^{M'} \overline{K_i}\mathbf{X}(\frac{\overline{z_i}}{h})$$

where z_i (poles) and K_i (residues) are given in Table (A.1), M' is the number of rows in the table, and the bar denotes the complex conjugate. To go to the next

z_i	$2 * K_i$
11.83009373916819 +j 1.593753005885813	16286.62368050479 -j 139074.7115516051
11.22085377939519 +j 4.792964167565670	-28178.11171305163 +j 74357.58237274176
9.933383722175002 +j 8.033106334266296	14629.74025233142 -j 19181.80818501836
7.781146264464616 +j 11.36889164904993	-2870.418161032078 +j 1674.109484084304
4.234522494797000 +j 14.95704378128156	132.1659412474876 +j 17.47674798877164

Table A.1: Poles z_i and residues K_i used in numerical Laplace transform inversion (M=10 , N=8).

point in time, we reset the problem so that in the next evaluation the previous result is considered as the initial point for the new step. To do this, the initial condition **I** is obtained from **x** by means of

$$\mathbf{I} = \mathbf{C}\mathbf{x}$$

We apply the above method, called *stepping algorithm*, to compute the **P** and $\mathbf{P_s}$ vectors defined by

$$\mathbf{P} = \mathcal{L}^{-1}\left(\mathbf{R}^{-1}\mathbf{W}\right)\Big|_{t=T}$$

$$\mathbf{P_s} = \mathcal{L}^{-1}\left(-\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial l}\mathbf{R}^{-1}\mathbf{W}\right)\Big|_{t=T}$$

The algorithm is

1. Prepare the vector of initial conditions $\mathbf{I}(t_0 = 0)$ for the original network, and $\mathbf{I}_s(t_0 = 0)$ for the sensitivity network. These are null vectors for initially relaxed circuits. Select the step size $h = T/N$, where N is the number of steps.

It is an arbitrary number that determines the accuracy of the computation, and is explained later. Set $k = 1$.

2. Substitute $\frac{z_i}{h}$ for each s in the system of equation

$$\mathbf{R}(s)\mathbf{X}(s) = \mathbf{W}(s) + \mathbf{I}((k-1)h),$$

and solve it to get

$$\mathbf{X}_i(\frac{z_i}{h}) = \mathbf{R}^{-1}(\frac{z_i}{h})\left[\mathbf{W}(\frac{z_i}{h}) + \mathbf{I}((k-1)h)\right].$$

Create the right hand side of the sensitivity network (sensitivity with respect to element l)

$$\mathbf{Q}(\frac{z_i}{h}) = -\frac{\partial \mathbf{R}}{\partial l}(\frac{z_i}{h})\mathbf{X}_i(\frac{z_i}{h}) + \frac{\partial \mathbf{W}}{\partial l}(\frac{z_i}{h})$$

and solve

$$\mathbf{R}((\frac{z_i}{h}))\mathbf{Y}_i(\frac{z_i}{h}) = \mathbf{Q}(\frac{z_i}{h}) + \mathbf{I}_s((k-1)h)$$

to obtain $\mathbf{Y}_i(\frac{z_i}{h})$.

3. Multiply $\mathbf{X}_i(\frac{z_i}{h})$ by K_i and add the products. Multiply $\mathbf{Y}_i(\frac{z_i}{h})$ by K_i and add the products.

4. Repeat step 2 and step 3 for all z_i's and K_i's in Table (A.1) to get

$$\mathbf{x} = \sum_{i=1}^{M'} K_i \mathbf{X}_i(\frac{z_i}{h})$$

$$\mathbf{y} = \sum_{i=1}^{M'} K_i \mathbf{Y}_i(\frac{z_i}{h})$$

5. Consider the conjugates of z_i and K_i, repeat the steps 2,3, and 4, and divide the results by $(-h)$ to obtain

$$\mathbf{x}(t=kh) = -\frac{1}{h}\sum_{i=1}^{M'} K_i \mathbf{X}_i(\frac{z_i}{h}) - \frac{1}{h}\sum_{i=1}^{M'} \overline{K_i}\mathbf{X}_i(\frac{\overline{z_i}}{h})$$

$$\mathbf{y}(t=kh) = -\frac{1}{h}\sum_{i=1}^{M'} K_i \mathbf{Y}_i(\frac{z_i}{h}) - \frac{1}{h}\sum_{i=1}^{M'} \overline{K_i}\mathbf{Y}_i(\frac{\overline{z_i}}{h})$$

6. Prepare the new initial vector $\mathbf{I}(kh) = \mathbf{C}\ \mathbf{x}(kh), \mathbf{I}_s(kh) = \mathbf{C}\ \mathbf{y}(kh)$. Set $k = k+1$, repeat steps 2 to 6 till k reaches N.

7. Vector $\mathbf{x}$ now contains the time domain solution of the original network at $t = Nh = T$, which is equivalent to vector P required for sampled data simulation. Vector $\mathbf{y}$ contains the time domain solution of the sensitivity network at $t = Nh = T$, which is considered as $\mathbf{P_s}$ in the time domain sensitivity analysis.

The number of steps, N, determines the step size $h = \frac{T}{N}$, and so the truncation error. The above algorithm is repeated once with the number of steps equal to N, and then equal to 2N. The difference of the two results gives the truncation error. If the error is more than the desired value, the number of steps is multiplied by 2, the step size is divided by 2, and the computation is repeated with the new step size.

The matrices $\mathbf{M}$ and $\mathbf{M_s}$ are calculated with the same algorithm. Since the source vector $\mathbf{W}$ does not appear in the computation of $\mathbf{M}$ and $\mathbf{M_s}$, an identity matrix is considered as the input. With this artificial input, all of the computation steps are the same as what we explained for $\mathbf{P}$ and $\mathbf{P_s}$.

Appendix B

The Algorithm for Calculation of Group Delay and Its Sensitivity

Part 1 : Pre-Processing

We apply one-Graph modified nodal analysis to formulate the circuit equations. The switches are modeled as resistors with a small resistance when they are closed and infinite resistance when open. The topology of the circuit therefore does not change during the different phases, and the matrices $\mathbf{B}_k$ become equal to $\mathbf{C}_k$. The number of phases is denoted by N.

The the input frequency is ω_0. All of the frequency variables in the following steps are $\omega = \omega_0$. If the input frequency is changed, all of the following steps (except 1-1 and 1-2) have to be repeated at the new frequency. An efficient method for computation of matrices in steps (1-3) to (1-6) is given in chapter 4. N is the total number of phases, and T is the clock period.

1-1- Prepare $(\mathbf{G}_1, \mathbf{C}_1), (\mathbf{G}_2, \mathbf{C}_2), \ldots, (\mathbf{G}_N, \mathbf{C}_N)$ where the matrices $\mathbf{G}_k$ and $\mathbf{C}_k$ construct the system matrix $\mathbf{R}_k$ during phase k, i.e $\mathbf{R}_k = \mathbf{G}_k + s\ \mathbf{C}_k$.

1-2- Prepare $\left(\frac{d\mathbf{G}_1}{dh}, \frac{d\mathbf{C}_1}{dh}\right), \left(\frac{d\mathbf{G}_2}{dh}, \frac{d\mathbf{C}_2}{dh}\right), \ldots, \left(\frac{d\mathbf{G}_N}{dh}, \frac{d\mathbf{C}_N}{dh}\right)$

1-3- Prepare $(\mathbf{P}_1, \mathbf{M}_1), (\mathbf{P}_2, \mathbf{M}_2), \ldots, (\mathbf{P}_N, \mathbf{M}_N)$

1-4- Prepare $\left(\frac{d\mathbf{P}_1}{dh}, \frac{d\mathbf{M}_1}{dh}\right), \left(\frac{d\mathbf{P}_2}{dh}, \frac{d\mathbf{M}_2}{dh}\right), \ldots, \left(\frac{d\mathbf{P}_N}{dh}, \frac{d\mathbf{M}_N}{dh}\right)$

1-5- Prepare $\left(\frac{d\mathbf{P}_1}{d\omega}\right), \left(\frac{d\mathbf{P}_2}{d\omega}\right), \ldots, \left(\frac{d\mathbf{P}_N}{d\omega}\right)$

1-6- Prepare $\left(\frac{d^2\mathbf{P}_1}{dhd\omega}\right), \left(\frac{d^2\mathbf{P}_2}{dhd\omega}\right), \ldots, \left(\frac{d^2\mathbf{P}_N}{dhd\omega}\right)$

Part 2 : Solution of the Discrete-Time Equations

2-1- Construct the following matrices related to the Discrete-Time set of equations:

$$\widehat{\mathbf{R}} = \begin{bmatrix} \mathbf{1}e^{j\omega\sigma_1} & \mathbf{0} & \mathbf{0} & \cdots & -\mathbf{M}_1 \\ -\mathbf{M}_2 e^{j\omega\sigma_1} & \mathbf{1}e^{j\omega\sigma_2} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & -\mathbf{M}_3 e^{j\omega\sigma_2} & \mathbf{1}e^{j\omega\sigma_3} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{M}_N e^{j\omega\sigma_{N-1}} & \mathbf{1}e^{j\omega T} \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \\ \cdots \\ \mathbf{P}_N \end{bmatrix}$$

$$\widetilde{\mathbf{D}} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1}e^{j\omega\sigma_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1}e^{j\omega\sigma_2} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1}e^{j\omega\sigma_{N-1}} \end{bmatrix}$$

$$\widehat{\mathbf{W}} = \frac{1}{T}\,\widetilde{\mathbf{D}}\,\mathbf{P}$$

$$\frac{d\widehat{\mathbf{R}}}{d\omega} = \begin{bmatrix} \mathbf{1}j\sigma_1 e^{j\omega\sigma_1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ -\mathbf{M}_2 j\sigma_1 e^{j\omega\sigma_1} & \mathbf{1}j\sigma_2 e^{j\omega\sigma_2} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & -\mathbf{M}_3 j\sigma_2 e^{j\omega\sigma_2} & \mathbf{1}j\sigma_3 e^{j\omega\sigma_3} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{M}_N j\sigma_{N-1} e^{j\omega\sigma_{N-1}} & \mathbf{1}jTe^{j\omega T} \end{bmatrix}$$

$$\frac{d\widehat{\mathbf{R}}}{dh} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & -\frac{d\mathbf{M}_1}{dh} \\ -\frac{d\mathbf{M}_2}{dh} e^{j\omega\sigma_1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & -\frac{d\mathbf{M}_3}{dh} e^{j\omega\sigma_2} & \mathbf{0} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\frac{d\mathbf{M}_N}{dh} e^{j\omega\sigma_{N-1}} & \mathbf{0} \end{bmatrix}$$

$$\frac{d^2\widehat{\mathbf{R}}}{dhd\omega} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ -\frac{d\mathbf{M}_2}{dh} j\sigma_1 e^{j\omega\sigma_1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & -\frac{d\mathbf{M}_3}{dh} j\sigma_2 e^{j\omega\sigma_2} & \mathbf{0} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\frac{d\mathbf{M}_N}{dh} j\sigma_{N-1} e^{j\omega\sigma_{N-1}} & \mathbf{0} \end{bmatrix}$$

$$\frac{d\widehat{\mathbf{W}}}{d\omega} = \frac{1}{T} \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1} j\sigma_1 e^{j\omega\sigma_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} j\sigma_2 e^{j\omega\sigma_2} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1} j\sigma_{N-1} e^{j\omega\sigma_{N-1}} \end{bmatrix} \mathbf{P} + \frac{1}{T}\widetilde{\mathbf{D}} \begin{bmatrix} \frac{d\mathbf{P}_1}{d\omega} \\ \frac{d\mathbf{P}_2}{d\omega} \\ \frac{d\mathbf{P}_3}{d\omega} \\ \cdots \\ \frac{d\mathbf{P}_N}{d\omega} \end{bmatrix}$$

$$\frac{d\widehat{\mathbf{W}}}{dh} = \frac{1}{T}\,\widetilde{\mathbf{D}} \begin{bmatrix} \frac{d\mathbf{P}_1}{dh} \\ \frac{d\mathbf{P}_2}{dh} \\ \frac{d\mathbf{P}_3}{dh} \\ \cdots \\ \frac{d\mathbf{P}_N}{dh} \end{bmatrix}$$

$$\frac{d^2\widehat{\mathbf{W}}}{dhd\omega} = \frac{1}{T} \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1} j\sigma_1 e^{j\omega\sigma_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} j\sigma_2 e^{j\omega\sigma_2} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1} j\sigma_{N-1} e^{j\omega\sigma_{N-1}} \end{bmatrix} \begin{bmatrix} \frac{d\mathbf{P}_1}{dh} \\ \frac{d\mathbf{P}_2}{dh} \\ \frac{d\mathbf{P}_3}{dh} \\ \cdots \\ \frac{d\mathbf{P}_N}{dh} \end{bmatrix} + \frac{1}{T}\,\widetilde{\mathbf{D}} \begin{bmatrix} \frac{d^2\mathbf{P}_1}{dhd\omega} \\ \frac{d^2\mathbf{P}_2}{dhd\omega} \\ \frac{d^2\mathbf{P}_3}{dhd\omega} \\ \cdots \\ \frac{d^2\mathbf{P}_N}{dhd\omega} \end{bmatrix}$$

2-2- Compute

$$\widehat{\mathbf{V}}_P = \widehat{\mathbf{R}}^{-1}\,\widehat{\mathbf{W}} \qquad \text{(B.1)}$$

2-3- Using $\widehat{\mathbf{V}}_P$ from (B.1), compute

$$\frac{d\widehat{\mathbf{V}}_P}{d\omega} = -\widehat{\mathbf{R}}^{-1}\frac{d\widehat{\mathbf{R}}}{d\omega}\widehat{\mathbf{V}}_P + \widehat{\mathbf{R}}^{-1}\frac{d\widehat{\mathbf{W}}}{d\omega} \tag{B.2}$$

2-4- Using $\widehat{\mathbf{V}}_P$ from (B.1), compute

$$\frac{d\widehat{\mathbf{V}}_P}{dh} = -\widehat{\mathbf{R}}^{-1}\frac{d\widehat{\mathbf{R}}}{dh}\widehat{\mathbf{V}}_P + \widehat{\mathbf{R}}^{-1}\frac{d\widehat{\mathbf{W}}}{dh} \tag{B.3}$$

2-5- Using $\widehat{\mathbf{V}}_P$, $\frac{d\widehat{\mathbf{V}}_P}{d\omega}$, and $\frac{d\widehat{\mathbf{V}}_P}{dh}$ obtained in the previous equations, compute

$$\frac{d^2\widehat{\mathbf{V}}_P}{dhd\omega} = \widehat{\mathbf{R}}^{-1}\left[-\frac{d\widehat{\mathbf{R}}}{dh}\frac{d\widehat{\mathbf{V}}_P}{d\omega} - \frac{d^2\widehat{\mathbf{R}}}{dhd\omega}\widehat{\mathbf{V}}_P - \frac{d\widehat{\mathbf{R}}}{d\omega}\frac{d\widehat{\mathbf{V}}_P}{dh} + \frac{d^2\widehat{\mathbf{W}}}{dhd\omega}\right] \tag{B.4}$$

Part 3 : Solution of the Continuous-Time Equations

3-1- Construct the following matrices related to the Continuous-Time set of equations. Use the results of (B.1) to (B.4) to construct these matrices.

$$\mathbf{R} = \begin{bmatrix} \mathbf{G}_1 + j\omega\mathbf{C}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_2 + j\omega\mathbf{C}_2 & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{G}_N + j\omega\mathbf{C}_N \end{bmatrix}$$

$$\mathbf{W} = \begin{bmatrix} \mathbf{g}_1\theta_1 + \mathbf{C}_1\widetilde{\mathbf{V}}_N - \mathbf{C}_1\widetilde{\mathbf{V}}_1 \\ \mathbf{g}_2\theta_2 + \mathbf{C}_2\widetilde{\mathbf{V}}_1 - \mathbf{C}_2\widetilde{\mathbf{V}}_2 \\ \cdots \\ \mathbf{g}_N\theta_N + \mathbf{C}_N\widetilde{\mathbf{V}}_{N-1} - \mathbf{C}_N\widetilde{\mathbf{V}}_N \end{bmatrix}$$

$$\frac{d\mathbf{R}}{d\omega} = \begin{bmatrix} j\mathbf{C}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & j\mathbf{C}_2 & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \cdots & j\mathbf{C}_N \end{bmatrix}$$

$$\frac{d\mathbf{R}}{dh} = \begin{bmatrix} \frac{d\mathbf{G}_1}{dh} + j\omega\frac{d\mathbf{C}_1}{dh} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \frac{d\mathbf{G}_2}{dh} + j\omega\frac{d\mathbf{C}_2}{dh} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \cdots & \frac{d\mathbf{G}_N}{dh} + j\omega\frac{d\mathbf{C}_N}{dh} \end{bmatrix}$$

$$\frac{d^2\mathbf{R}}{dhd\omega} = \begin{bmatrix} j\frac{d\mathbf{C}_1}{dh} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & j\frac{d\mathbf{C}_2}{dh} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \cdots & j\frac{d\mathbf{C}_N}{dh} \end{bmatrix}$$

$$\frac{d\mathbf{W}}{d\omega} = \begin{bmatrix} \mathbf{C}_1\frac{d\tilde{\mathbf{V}}_N}{d\omega} - \mathbf{C}_1\frac{d\tilde{\mathbf{V}}_1}{d\omega} \\ \mathbf{C}_2\frac{d\tilde{\mathbf{V}}_1}{d\omega} - \mathbf{C}_2\frac{d\tilde{\mathbf{V}}_2}{d\omega} \\ \cdots \\ \mathbf{C}_N\frac{d\tilde{\mathbf{V}}_{N-1}}{d\omega} - \mathbf{C}_N\frac{d\tilde{\mathbf{V}}_N}{d\omega} \end{bmatrix}$$

$$\frac{d\mathbf{W}}{dh} = \begin{bmatrix} \frac{d\mathbf{C}_1}{dh}\widetilde{\mathbf{V}}_N - \frac{d\mathbf{C}_1}{dh}\widetilde{\mathbf{V}}_1 \\ \frac{d\mathbf{C}_2}{dh}\widetilde{\mathbf{V}}_1 - \frac{d\mathbf{C}_2}{dh}\widetilde{\mathbf{V}}_2 \\ \cdots \\ \frac{d\mathbf{C}_N}{dh}\widetilde{\mathbf{V}}_{N-1} - \frac{d\mathbf{C}_N}{dh}\widetilde{\mathbf{V}}_N \end{bmatrix} + \begin{bmatrix} \mathbf{C}_1\frac{d\tilde{\mathbf{V}}_N}{dh} - \mathbf{C}_1\frac{d\tilde{\mathbf{V}}_1}{dh} \\ \mathbf{C}_2\frac{d\tilde{\mathbf{V}}_1}{dh} - \mathbf{C}_2\frac{d\tilde{\mathbf{V}}_2}{dh} \\ \cdots \\ \mathbf{C}_N\frac{d\tilde{\mathbf{V}}_{N-1}}{dh} - \mathbf{C}_N\frac{d\tilde{\mathbf{V}}_N}{dh} \end{bmatrix}$$

$$\frac{d^2\mathbf{W}}{dhd\omega} = \begin{bmatrix} \frac{d\mathbf{C}_1}{dh}\frac{d\tilde{\mathbf{V}}_N}{d\omega} - \frac{d\mathbf{C}_1}{dh}\frac{d\tilde{\mathbf{V}}_1}{d\omega} \\ \frac{d\mathbf{C}_2}{dh}\frac{d\tilde{\mathbf{V}}_1}{d\omega} - \frac{d\mathbf{C}_2}{dh}\frac{d\tilde{\mathbf{V}}_2}{d\omega} \\ \cdots \\ \frac{d\mathbf{C}_N}{dh}\frac{d\tilde{\mathbf{V}}_{N-1}}{d\omega} - \frac{d\mathbf{C}_N}{dh}\frac{\tilde{\mathbf{V}}_N}{d\omega} \end{bmatrix} + \begin{bmatrix} \mathbf{C}_1\frac{d^2\tilde{\mathbf{V}}_N}{dhd\omega} - \mathbf{C}_1\frac{d^2\tilde{\mathbf{V}}_1}{dhd\omega} \\ \mathbf{C}_2\frac{d^2\tilde{\mathbf{V}}_1}{dhd\omega} - \mathbf{C}_2\frac{d^2\tilde{\mathbf{V}}_2}{dhd\omega} \\ \cdots \\ \mathbf{C}_N\frac{d^2\tilde{\mathbf{V}}_{N-1}}{dhd\omega} - \mathbf{C}_N\frac{d^2\tilde{\mathbf{V}}_N}{dhd\omega} \end{bmatrix}$$

3-2- Compute

$$\mathbf{V_P} = \mathbf{R}^{-1}\,\mathbf{W} \tag{B.5}$$

3-3- Using $\mathbf{V_P}$ from (B.5), compute

$$\frac{d\mathbf{V_P}}{d\omega} = -\mathbf{R}^{-1}\frac{d\mathbf{R}}{d\omega}\mathbf{V_P} + \mathbf{R}^{-1}\frac{d\mathbf{W}}{d\omega} \tag{B.6}$$

3-4- Using $\mathbf{V_P}$ from (B.5), compute

$$\frac{d\mathbf{V_P}}{dh} = -\mathbf{R}^{-1}\frac{d\mathbf{R}}{dh}\mathbf{V_P} + \mathbf{R}^{-1}\frac{d\mathbf{W}}{dh} \tag{B.7}$$

3-5- Using $\mathbf{V_P}$, $\frac{d\mathbf{V_P}}{d\omega}$, and $\frac{d\mathbf{V_P}}{dh}$ obtained in the previous equations, compute

$$\frac{d^2\mathbf{V_P}}{dhd\omega} = \mathbf{R}^{-1}\left[-\frac{d\mathbf{R}}{dh}\frac{d\mathbf{V_P}}{d\omega} - \frac{d^2\mathbf{R}}{dhd\omega}\mathbf{V_P} - \frac{d\mathbf{R}}{d\omega}\frac{d\mathbf{V_P}}{dh} + \frac{d^2\mathbf{W}}{dhd\omega}\right] \tag{B.8}$$

Part 4 : Computing the Group Delay, and Group Delay Sensitivity

The complete vector of nodal voltages is the summation of the vectors corresponding to each phase:

$$\begin{aligned} \mathbf{V} &= \sum_{k=1}^{N}\mathbf{V}_k \\ \frac{d\mathbf{V}}{dh} &= \sum_{k=1}^{N}\frac{d\mathbf{V}_k}{dh} \\ \frac{d\mathbf{V}}{d\omega} &= \sum_{k=1}^{N}\frac{d\mathbf{V}_k}{d\omega} \\ \frac{d^2\mathbf{V}}{dhd\omega} &= \sum_{k=1}^{N}\frac{d^2\mathbf{V}_k}{dhd\omega} \end{aligned}$$

Assume that the output is one of the elements in $\mathbf{V}$, like $\mathbf{V}(k)$ (in general, the output is related to the vector $\mathbf{V}$ by the selector vector $\mathbf{d}$, i.e $\phi = \mathbf{d}^t\ \mathbf{V}$).
The group delay is calculated as

$$\tau(\omega) = -\mathrm{Im}\left[\frac{1}{\mathbf{V}(k)}\frac{d\mathbf{V}(k)}{d\omega}\right]$$

and the group delay sensitivity with respect to element h is calculated by

$$\frac{\tau(\omega)}{dh} = \mathrm{Im}\left[\frac{1}{\mathbf{V}^2(k)}\frac{d\mathbf{V}(k)}{dh}\frac{d\mathbf{V}(k)}{d\omega} - \frac{1}{\mathbf{V}(k)}\frac{d^2\mathbf{V}(k)}{dhd\omega}\right]$$

Appendix C

Selected Switched-Current Circuits

The nonfiltering applications of switched-current circuits proposed in section 2 of chapter 5 were implemented using the circuits selected from the literature. The citations to the references and a brief explanations of the circuits are presented in this appendix.

I - Reference Current Switch

To provide two levels of reference current, I_{ref} and $-I_{ref}$, we consider a fully-differential current switch reported in [52]. Such a switch achieves fast settling times and high switching speeds.

A circuit diagram for differential current switch is shown in Figure C.1. In this circuit, the reference current I_{ref} is switched to either I_{out} or $\overline{I_{out}}$ using an NMOS differential pair consisting of transistors M_1 and M_2. Switching the reference current to I_{out} requires driving the gate of transistor M_1 to a positive potential V_{bias} while

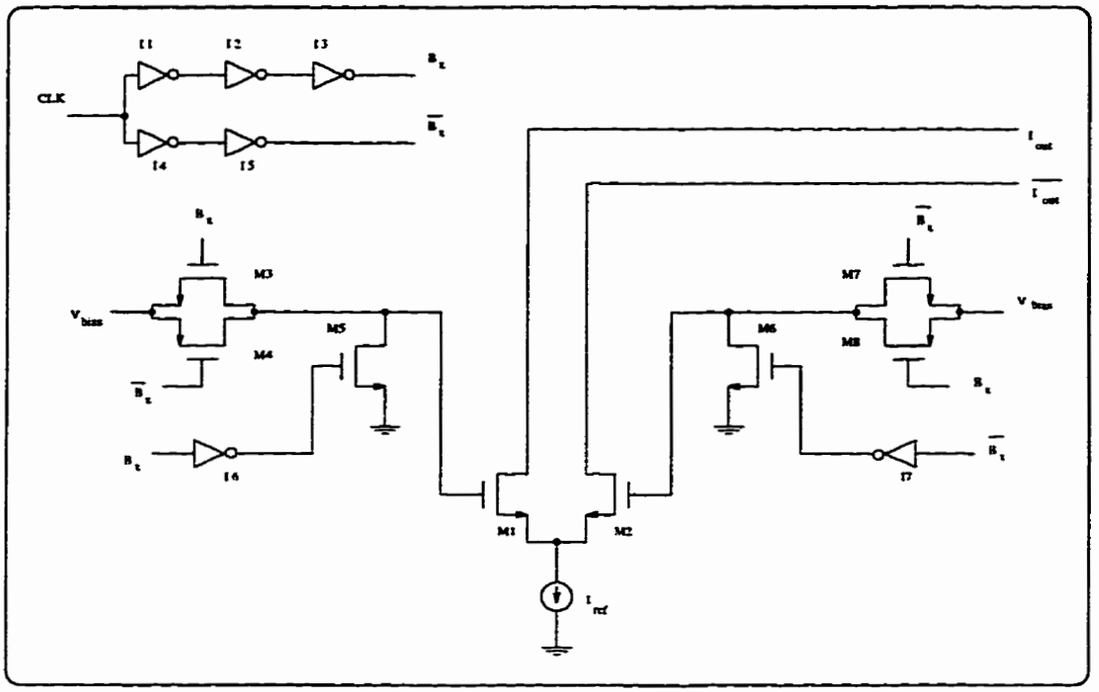


Figure C.1: A fully differential current switch.

pulling the gate of transistor M_2 to ground. Driving the gate potential of M_1 to V_{bias} is done with the transmission gate composed of transistors M_3 and M_4 while the gate of M_2 is pulled to ground using a single N-channel device M_6.

The transmission gate is driven by a pair of complementary signals, B_x and $\overline{B_x}$. These signals are driven from the digital input clock by using two inverter strings I_1, I_2, I_3 and I_4, I_5. By equalizing the delay through each inverter string, the rising and falling transitions of B_x and $\overline{B_x}$ can be made to overlap. Furthermore, if the delay through inverter i equals T_i, then choosing $T_2 = T_5$ and $T_1 + T_3 = T_4$ ensures that the rising and falling transitions of B_x and $\overline{B_x}$ occur simultaneously, even in the face of process variations [52]. To satisfy the above conditions, device sizes are chosen such that each of the inverters I_1 to I_5 drives the same load capacitance. Inverters I_1, I_2, I_3 and I_5 have the same size with $W_p/L_p = 36\mu/3\mu$ and $W_n/L_n =$

$15\mu/3\mu$, while inverter I_4 is chosen to have twice the channel length $W_p/L_p = 36\mu/6\mu, W_n/L_n = 15\mu/6\mu$. The pull down transient at the gate of M_2 is controlled by the device M_6 driven by inverter I_7. Adding inverter I_7 delays the pull down transient at the gate of M_2 until the voltage at the gate of M_1 has risen sufficiently to turn M_1 on. Both inverters I_6 and I_7 have $W_p/L_p = 6\mu/3\mu, W_n/L_n = 15\mu/3\mu$.

II - Fully Balanced SI Integrator

We consider a fully balanced SI integrator proposed in [53,54] because of its first-order cancellation of clock feed through effects, 6-dB increase in dynamic range, improvement in common-mode rejection ratio (CMRR) and power supply rejection ratio (PSRR). To understand the operation of the fully balanced integrator, we first explain the operation of the fully balanced current mirror shown in Figure C.2. When a signal current $+i_{in}/2$ is injected into the low-impedance node associated with the diode-connected device P_1, the current $(I - i_{in}/2)$ that follows through the $N_1 - p_1$ branch is converted to a voltage between the gates of N_1 and P_1. The resulting voltage is applied to the gate of P_2 while the gate of N_3 is connected to V_B. Hence, the current $(I - i_{in}/2)$ is mirrored from $P_1 - N_1$ to $P_2 - N_3$ and reflected to the differential outputs as shown. Assuming matched pairs, the fully balanced current mirror has a small signal current gain of two since the differential-mode output current is $2i_{in}$. One way to increase the current gain is to scale the current mirror aspect ratios. As shown in Figure C.2, if $P_2 - P_3$ and $N_2 - N_3$ are scaled by K_1 relative to $P_1 - P_4$ and $N_1 - N_4$, and the output mirrors are scaled by K_2, the small signal current gain is $2K_1K_2$.

A fully balanced SI integrator is obtained from the fully balanced mirror by adding MOS switches between current-mirror transistor pairs as shown in Figure C.3. During ϕ_1, the input currents are applied to the low impedance nodes

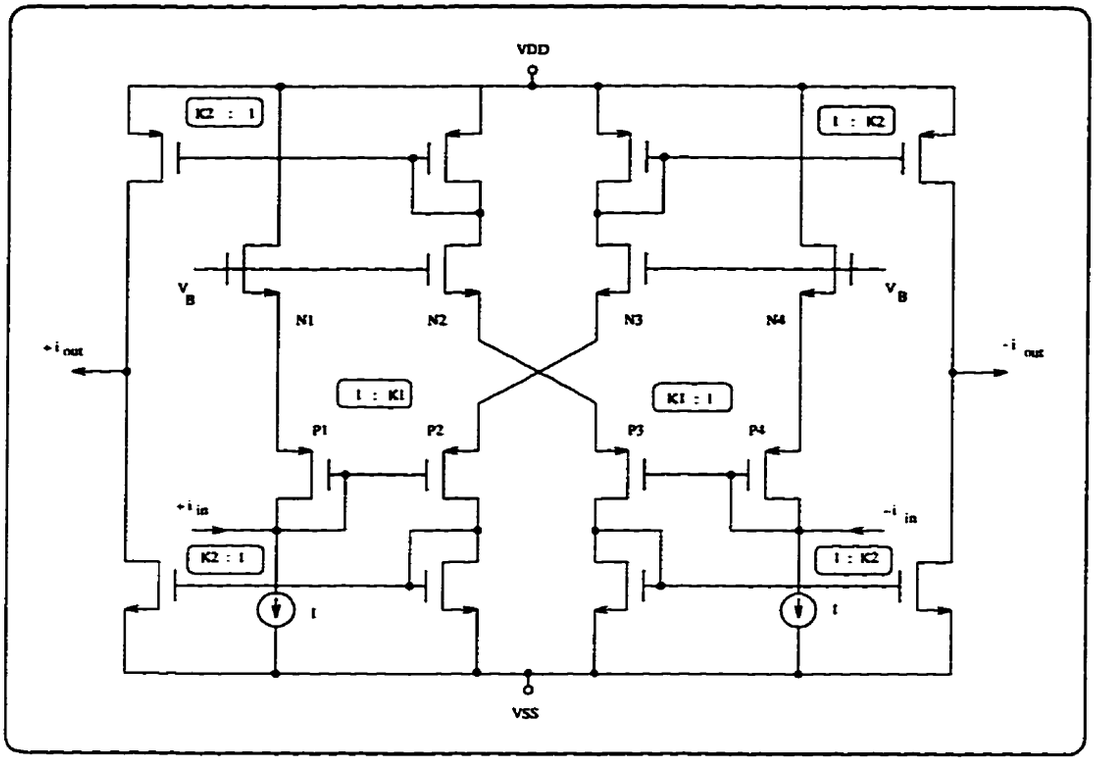


Figure C.2: A fully balanced current mirror with open loop current gain $2K_1K_2$.

associated with P_1 and P_4, and during ϕ_2, the signal currents are sampled by the output T/H stages $N_5 - N_6, N_8 - N_9, P_5 - P_6$, and $P_8 - P_9$.The integration function is obtained by feedback of the balanced outputs $+i_{out}$ and $-i_{out}$ to the balanced inputs. Scaled output branches N_7, P_7 and N_{10}, P_{10} provide the desired integrator gain constant.

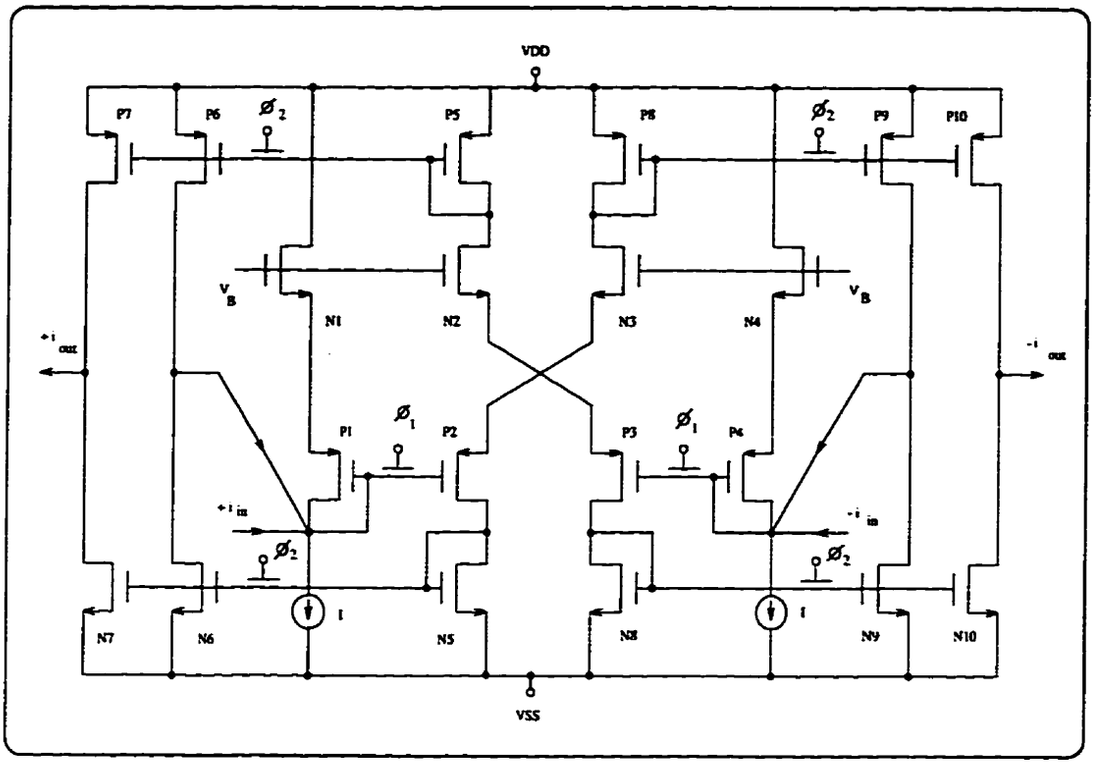


Figure C.3: A fully balanced SI integrator.

III - Current Schmitt Trigger

A two-input CMOS current Schmitt trigger with adjustable hysteresis reported in [55] is shown in Figure C.4. Transistors M_1, M_2, M_{11}, M_{12} and M_5, M_6, M_9, M_{10} are matched pairs which compare the currents $I_{d7} + I_{in1}$ and $I_{d8} + I_{in2}$. The output of this comparison controls the differential switching stage (M_7, M_8). We start with the input I_{in2}, very small in comparison with I_{in1}. V_{d3} is then "high" and V_{d4} is "low". It means M_7 is ON and M_8 is OFF. The current through M_1 is $I_{in1} + I_{hy}$. If I_{in2} increases and exceeds the above value, V_{d3} will snap to "low" and V_{d4} to "high". Therefore, when

$$I_{in2} > (I_{in1} + I_{hy}) \tag{C.1}$$

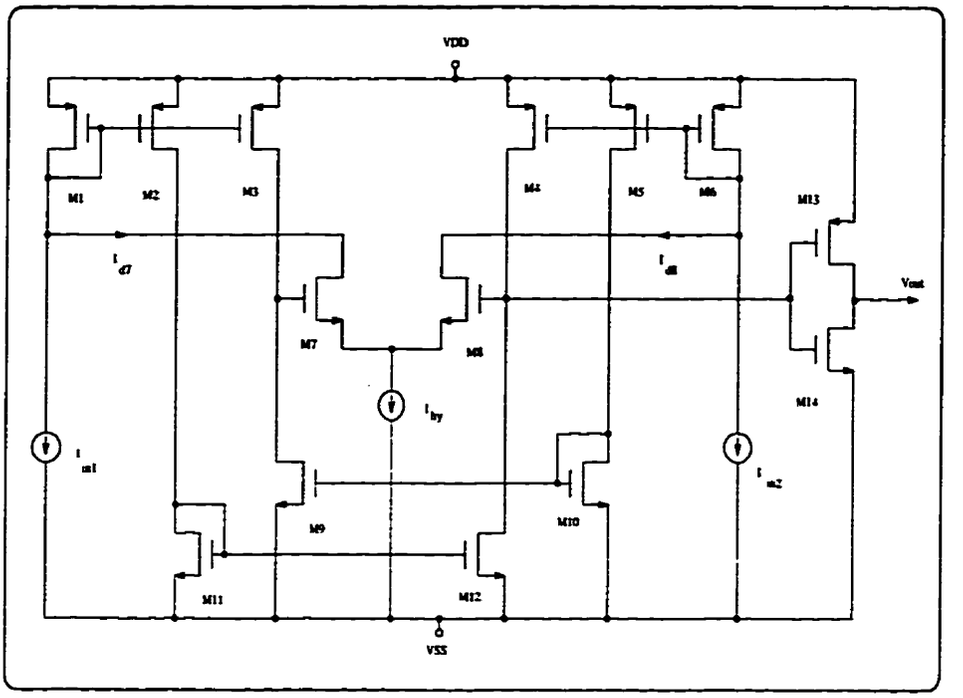


Figure C.4: Two-input current Schmitt trigger with adjustable hysteresis.

the transistor M_7 is OFF and M_2 is ON. The current through M_6 is $I_{in2}+I_{hy}$, if I_{in1} exceeds this value, the circuit comes back to the starting state where V_{d3} is "high" and V_{d4} is "low" (M_7 is ON and M_8 is OFF). Therefore, the following condition will change the state of the Schmitt trigger :

$$I_{in1} > (I_{in2} + I_{hy}) \tag{C.2}$$

From (C.1) and (C.2) we conclude that the following conditions will change the state of the Schmitt trigger:

$$\begin{aligned} I_{in1} - I_{in2} &< -I_{hy} \\ I_{in1} - I_{in2} &> I_{hy} \end{aligned} \tag{C.3}$$

As long as we consider the difference of the two input currents, $I_{in1} - I_{in2}$, equation(C.3) indicates that the hysteresis is equal to I_{hy}, and is independent of pro-

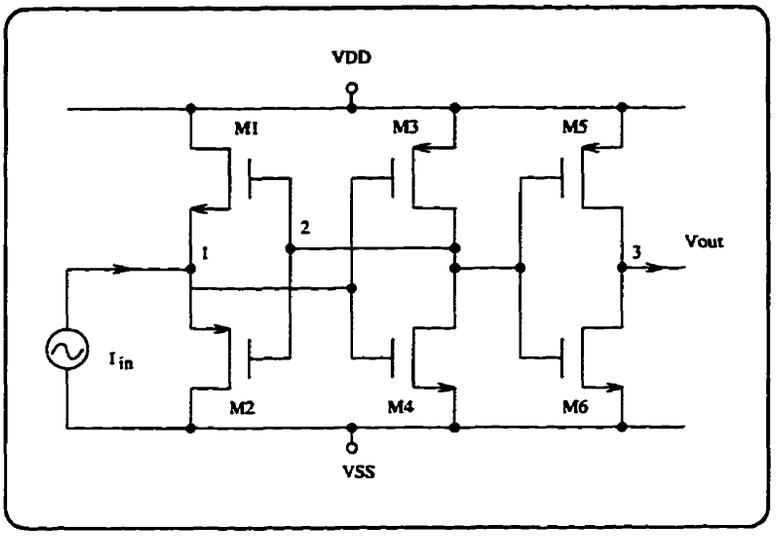


Figure C.5: A class B current comparator.

cess parameters, transistor dimensions and power supplies. The only requirement is matching of all transistors in current mirrors.

IV - Current Comparator

The current comparator reported in [56] is shown in Figure C.5. Transistors M_1, M_4, and M_6 have $W_n/L_n = 3\mu/1.2\mu$, and transistors M_2, M_3, and M_5 have $W_p/L_p = 9\mu/1.2\mu$. M_1 and M_2 form a class B voltage buffer and M_3 to M_6 form two inverting amplifiers, each with gain $-g_m/g_{ds}$. I_{in} is the input current. When I_{in} is positive, V(1) is pulled high. This is amplified by M_3 and M_4, causing V(2) to go low. V_{GS1} and V_{GS2} are negative, turning M_1 off and M_2 on. When I_{in} changes sign, there is insufficient gate drive for the buffer to supply I_{in}, thus V(1) is temporarily a high impedance node. When I_{in} is negative, V(1) is pulled low and V(2) is high, turning M_1 off and M_2 on.

Another alternative is a class AB current comparator proposed in [57]. In the circuit of Figure C.5 the size of dead region is determined by the threshold voltages

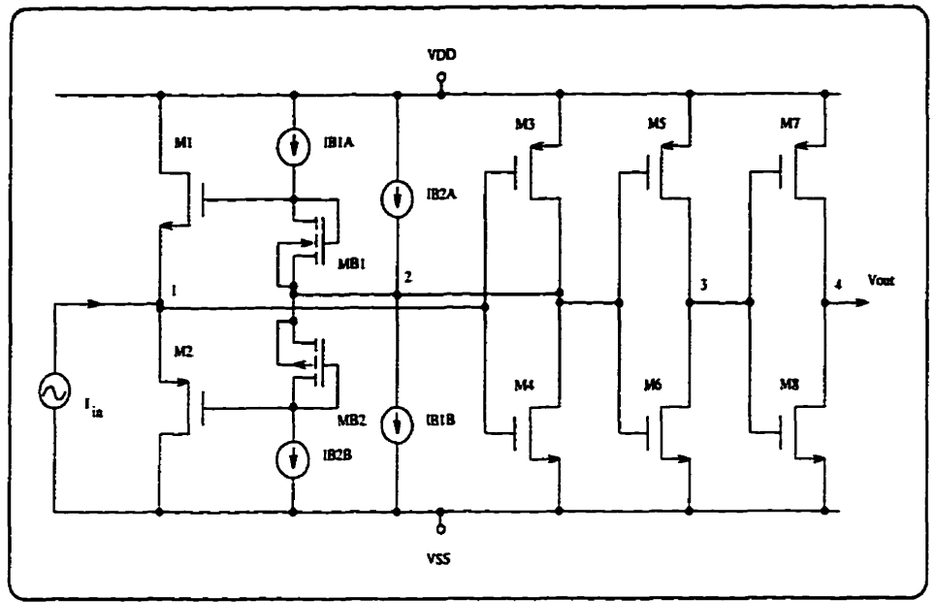


Figure C.6: A class AB current comparator.

of M_1 and M_2 which could be a large value. In Figure C.6, M_1 and M_2 are biased in class AB operation with gate-source voltages of VB1 and VB2, respectively. As the magnitudes of VB1 and VB2 are increased towards the magnitude of V_{T1} and V_{T2}, the dead-band in the transfer characteristic of the buffer is reduced. This results in smaller voltage swings at V(1) and V(2) and hence faster response times. Transistors MB1 and MB2 develop the voltage VB1 and VB2. Because the bulk-source voltage for MB1 and MB2 is less than that for M_1 and M_2, the threshold voltages are also lower due to the body effect. Since K' and V_T are different for NMOS and PMOS, the bias currents are different for MB1 and MB2. IB1A and IB1B are used to bias MB1, and IB2A and IB2B are used to bias MB2. Ideally, IB1A=IB1B and IB2A=IB2B.

Bibliography

[1] A. Opal, "Sampled data simulation of linear and nonlinear circuits," *IEEE Trans. on Computer-Aided Design*, vol. 15, pp. 295–307, March 1996.

[2] W. Nagel, *SPICE2: A computer program to simulate semiconductor circuits.* Memo No. ERL-M520: University of California, May 1975.

[3] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design.* New York: Van Nostrand Reinhold, second ed., 1994.

[4] S.W. Director and R.A. Rohrer, "The generalized adjoint network and network sensitivities," *IEEE Trans. on Circuit Theory*, vol. CT-16, pp. 318–323, August 1969.

[5] S.W. Director and R.A. Rohrer , "Automated network design - the frequency domain case," *IEEE Trans. on Circuit Theory*, vol. CT-16, pp. 330–337, August 1969.

[6] R. D. Davis, "A derivation of the switched-capacitor adjoint network based on a modified tellegen's theorem," *IEEE Trans. on Circuits and Systems*, vol. CAS-29, pp. 215–220, April 1982.

[7] L. Chua and P. Lin, *Computer-Aided Analysis of Electronic Circuits.* Englewood Cliffs, New Jersey: Prentice-Hall, 1975.

[8] C.W. Ho, A.E. Ruehli, and P.A. Brennan, "The modified nodal approach to network analysis," *IEEE Trans. on Circuits and Systems*, vol. CAS-22, pp. 504–509, June 1975.

[9] *HSPICE User's Manual.* 50 Curtner Ave., Suite 16, Campbell, CA., USA 95008: Meta Software, 1991.

[10] C. Moler and C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix," *SIAM Review*, vol. 20, pp. 801–836, October 1978.

[11] A. Opal, "The transition matrix for linear circuits," To be published.

[12] J. Vlach, "Numerical method for transient responses of linear networks with lumped, distributed or mixed parameters," *Journal of the Franklin Institute*, vol. 288, pp. 99–113, August 1969.

[13] K. Singhal and J. Vlach, "Computation of time domain response by numerical inversion of the Laplace transform," *Journal of the Franklin Institute*, vol. 299, pp. 109–126, February 1975.

[14] K. Singhal, J. Vlach, and M. Nakhla, "Absolutely stable, high order methods for time domain solution of networks," *AEU (Electronics and Communication*, vol. 30, pp. 157–166, April 1976.

[15] K. Singhal and J. Vlach, "Method for computing time response of systems described by transfer functions," *Journal of the Franklin Institute*, vol. 311, pp. 123–130, February 1981.

[16] *MATLAB for Sun Workstations, User's Guide.* Cochituate Place, 24 Prime Park Way, Natick, MA 01760: The Math works Inc., January 1990.

[17] G. Dahlquist and A. Bjorck, *Numerical methods.* Englewood Cliffs: Prentice-Hall, 1974.

[18] K.S. Kundert, "Sparce matrix techniques," *Circuit analysis, simulation and design*, vol. 3, pp. 281–324, 1986.

[19] A. Opal and J. Vlach, "Consistent initial conditions of linear switched networks," *IEEE Trans. on Circuits and Systems*, vol. CAS-37, pp. 364–372, March 1990.

[20] A. Opal and J. Vlach, "Consistent initial conditions of nonlinear networks with switches," *IEEE Trans. on Circuits and Systems*, vol. CAS-38, pp. 698–710, July 1991.

[21] B.D.H. Tellegen, "A general network theorem with applications," *Phillips Res. Rep.*, no. 7, pp. 259–269, 1952.

[22] P. Penfield, R. Spence, and S. Duinker, *Tellegen's theorem and electrical networks.* Massachusetts Institute of Technology. Research Monograph no. 58.: Cambridge, Mass., M.I.T. Press, 1970.

[23] M.W. Hauser, "Principles of oversampling A/D conversion," *Journal of Audio Eng. Soc.*, vol. 39, pp. 3–26, January-February 1991.

[24] J.C. Candy and G.C. Temes, Ed., *Oversampling Delta-Sigma Data Converters.* New York, USA: IEEE Press, 1992. The Chapter on Oversampling methods for A/D and D/A conversion, pages 1-29.

[25] B. Leung, "Theory of sigma-delta analog to digital converter," in *Circuits and Systems, IEEE Tutorials (ISCAS'94)* (G. Temes, ed.), ch. 4.1, pp. 195–223, New York, USA: IEEE Press, 1994.

[26] M.W. Hauser, "Technology scaling and performance limitations in Delta-Sigma analog-digital converters," *IEEE International Symposium on Circuits and Systems*, pp. 356–359, May 1990.

[27] P.J. Hurst and R.A. Levinson, "Delta-Sigma A/Ds with reduced sensitivity to opamp noise and gain," *IEEE International Symposium on Circuits and Systems*, pp. 254–257, May 1989.

[28] M.W. Hauser and R.W. Broderson, "Circuit and technology considerations for MOS Delta-Sigma converters," *IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 1310–1315, May 1986.

[29] O. Shoaei and W.M. Snelgrove, "Optimal bandpass continuous time $\sum \Delta$ modulator," *IEEE International Symposium on Circuits and Systems*, vol. London, England, pp. 489–492, May 1994.

[30] S.D. Willingham and K. Martin, *Integrated Video-Frequency Continuous-Time Filters: High performance realizations in BiCMOS.* 101 Philip Drive, Assinippi Park, Norwell, Massachusetts, 02061 USA: Kluwer Academic Publishers, 1995.

[31] G. Bird, *Design of continuous and digital electronic systems.* Maidenhead, Berkshire, England: McGraw-Hill Book Company Ltd., 1980.

[32] B. Christian, *LC-Filters: Design, Testing, and Manufacturing.* New York, USA: John Wiley & Sons, 1983.

[33] H. Blinchikoff and A. Zverev, *Filtering in the time and frequency domains.* New York, USA: John Wiley & Sons, 1976.

[34] J.T. Caves, M.A. Copeland, C.F. Rahim, and S.D. Rosenbaum, "Sampled analog filtering using switched capacitors as resistor equivalents," *IEEE Journal of Solid State Circuits*, vol. SC-12, pp. 592–599, December 1977.

[35] B.J. Hosticka, R.W. Brodersen, and P.R. Gray, "MOS sampled data recursive filters using switched capacitors integrators," *IEEE Journal of Solid State Circuits*, vol. SC-12, pp. 600–608, December 1977.

[36] E. G.S. Moschytz, *MOS switched-capacitor filters: Analysis and Design.* New York, USA: IEEE Press, 1984.

[37] M. Ghausi and K. Laker, *Modern filter design: Active RC and switched-capacitor.* Englewood cliffs, NJ, USA: Prentice-Hall Inc., 1981.

[38] R. Gregorian and G.C. Temes, *Analog MOS Integrated Circuits For Signal Processing.* U.S.A: John Wiley & Sons, 1986.

[39] Rolf Unbehauen and Andrzej Cichocki, *MOS switched-capacitor and continuous-time integrated circuits and systems.* Germany: Springer-Verlag, 1989.

[40] M.L. Liou, Y.L. Kuo, and C.F. Lee, "A tutorial on computer-aided analysis of switched-capacitor circuits," *Proc. IEEE*, vol. 71, pp. 987–1005, August 1983.

[41] Y. Tsividis, "Principles of operation and analysis of switched-capacitor circuits," *Proc. IEEE*, vol. 71, pp. 926–940, August 1983.

[42] J. Vlach, K. Singhal, and M. Vlach, "Computer oriented formulation of equations and analysis of switched capacitor networks," *IEEE Trans. on Circuits and Systems*, vol. CAS-31, pp. 753–765, September 1984.

[43] A. Opal and J. Vlach, "Analysis and sensitivity of periodically switched linear networks," *IEEE Trans. on Circuits and Systems*, vol. 36, pp. 552–532, April 1989.

[44] M.L. Liou, "Exact analysis of linear circuits containing periodically operated switches with applications," *IEEE Trans. on Circuit Theory*, vol. CT-19, pp. 146–154, March 1972.

[45] T. Strom and S. Signell, "Analysis of periodically switched linear networks," *IEEE Trans. on Circuits and Systems*, vol. CAS-24, pp. 531–540, October 1978.

[46] K. Raahemifar and A. Opal, "Efficient transient solution of linear circuits to sinusoidal inputs," *IEEE Trans. on Circuits and Systems, I- Fundamental theory and applications*, vol. 42, pp. 36–40, January 1995.

[47] J.B. Hughes, N.C. Bird, and I.C. Macbeth, "Switched-currents: A new technique for analog sampled-data signal processing," *IEEE International Symposium on Circuits and Systems*, pp. 1584–1587, 1989.

[48] J.B. Hughes, I.C. Macbeth, and D.M. Pattullo, "Second generation switched-current signal processing," *IEEE International Symposium on Circuits and Systems*, pp. 2805–2808, May 1990.

[49] Mohammed Ismail and Terri Fiez, ed., *Analog VLSI signal and information processing*. U.S.A: McGraw-Hill, 1994.

[50] S.V. Patttamatta, P. Manapragada, V. Dala, and R. Schreier, "A switched-current bandpass delta-sigma modulator," *IEEE International Symposium on Circuits and Systems*, pp. 477–480, 1994.

[51] B. Raahemi, A. Opal, J. Vlach, "Non-filtering applications of switched-current circuits," *Midwest Symposium on Circuits and Systems*, vol. 1, pp. 169–172, August 1994.

[52] M. Shoji, "Elimination of process dependent clock skew in CMOS VLSI," *IEEE Journal of Solid-State Circuits*, vol. SC-21, pp. 875–880, 1986.

[53] R.H. Zele, D.J. Allstot, and T.S. Fiez, "Fully balanced CMOS current-mode circuits," *IEEE Journal of Solid-State Circuits*, vol. 28, pp. 569–575, May 1993.

[54] R.H. Zele, D.J. Allstot, and T.S. Fies, "Fully-differential CMOS current-mode circuits," *IEEE Custom Integrated Circuits Conference*, pp. 24.1.1–24.1.4, 1991.

[55] Z. Wang and W. Guggenbuhl , "CMOS current Schmitt trigger with fully adjustable hysteresis," *Electronics Letters*, vol. 25, pp. 397–398, March 1989.

[56] H. Traff, "Novel approach to high speed CMOS current comparator," *Electronics Letters*, vol. 28, pp. 310–312, 30th January 1992.

[57] A.T. Tang and C. Toumazou , "High performance CMOS current comparator," *Electronics Letters*, vol. 30, pp. 5–6, 6th January 1994.

[58] S.J. Daubert, D. Vallancourt, and Y.P. Tsividis, "Current copier cells," *Electronics Letters*, vol. 24, pp. 1560–1562, 8th December 1988.

[59] G. Wegmann and E.A. Vittoz , "Analysis and improvements of accurate dynamic current mirrors," *IEEE Journal of Solid-State Circuits*, vol. 25, pp. 699–706, June 1990.

[60] T. S. Fiez and D.J. Allstot , "CMOS switched-current ladder filters," *IEEE Journal of Solid-State Circuits*, vol. 25, pp. 1360–1367, December 1990.

[61] T.S. Fiez, B. Lee, and D.J. Allstot , "CMOS switched-current biquadratic filters," *IEEE International Symposium on Circuits and Systems*, pp. 2300–2304, May 1990.

[62] J.B. Hughes, I.C. Macbeth, and D.M. Pattullo, "Switched current filters," *IEE Proceedings, Part G*, vol. 137, pp. 156–162, April 1990.

[63] J.B. Hughes, I.C. Macbeth, and D.M. Pattullo, "Switched-current system cells," *IEEE International Symposium on Circuits and Systems*, pp. 303–306, 1990.

[64] G.W. Roberts and A.S. Sedra , "Synthesizing switched-current filters by transposing the SFG of switched-capacitor filter circuits," *IEEE Transactions on Circuits and Systems*, vol. CAS-38, pp. 337–340, March 1991.

[65] C. Toumazou, F.J. Lidgey, and D.G. Haigh, ed., *Analogue IC design : the current-mode approach*. England: Peter Peregrinus Ltd., 1990.

[66] J.B. Hughes and K. Moulding, "Switched-current signal processing for video frequencies and beyond," *IEEE Journal of Solid-State Circuits*, vol. 28, pp. 314–322, March 1993.

[67] S.J. Daubert and D. Vallancourt, "Operation and analysis of current copier circuits," *Electronics Letters*, vol. 137, pp. 109–115, April 1990.

[68] C. Toumazou, J.B. Hughes, and D.M. Pattullo, "Regulated cascode switched-current memory cell," *Electronics Letters*, vol. 26, pp. 303–305, 1st March 1990.

[69] G. Wegmann and E.A. Vittoz, "Very accurate dynamic current mirror," *Electronics Letters*, vol. 25, pp. 644–646, 11th May 1989.

[70] N.C. Battersby and C. Toumazou, "Class AB switched-current memory for analogue sampled-data systems," *Electronics Letters*, vol. 27, pp. 873–875, 9th May 1991.

[71] C.A. Gobet and A. Knob, "Noise analysis of switched capacitor networks," *IEEE Transactions on Circuits and Systems*, vol. CAS-30, pp. 37–43, January 1983.

[72] R.T. Baird, T.S. Fiez, and D.J. Allstot , "Speed and accuracy considerations in switched-current circuits," *IEEE Proceedings of International Symposium on Circuits and Systems*, vol. 3, pp. 1809–1812, June 1991.

[73] G.C. Temes, P. Deval, and V. Valencic , "SC circuits : the state of the art compared to SI techniques," *IEEE Proceedings of International Symposium on Circuits and Systems*, pp. 1231–1234, 1993.

[74] L. Jonsson and B.A. Molin, "Accurate switched-current ladder filters based on active current mirrors," *Electronics Letters*, vol. 29, pp. 1840–1841, 14th October 1993.

[75] T.S. Fiez, G. Liang, and D.J. Allstot, "Switched-current circuit design issues," *IEEE Journal of Solid-State Circuits*, vol. 26, pp. 192–202, March 1991.

[76] G. Liagn and J. Allstot, "FIR filtering using CMOS switched current techniques," *IEEE International Symposium on Circuits and Systems*, pp. 2291–2294, May 1990.

[77] A. Begisi, T.S. Fiez, and D.J. Allstot , "Digitally-programmable switched-current filters," *IEEE International Symposium on Circuits and Systems*, pp. 3178–3181, May 1990.

[78] B.E. Boser and B.A. Wooley, "The design of sigma-delta modulation analog-to-digital converters," *IEEE Journal of Solid-State Circuits*, vol. SC-23, pp. 1298–1308, December 1988.

[79] J. H. Shieh, M. Patil, and B.J. Sheu , "Measurement and analysis of charge injection in MOS analog switches," *IEEE Journal of Solid-State Circuits*, vol. SC-22, pp. 277–281, April 1987.

[80] B.J. Sheu, J. Shieh, and M. Patil, "Modeling charge injection in MOS analog switches," *IEEE Trans. on Circuits and Systems*, vol. CAS-34, pp. 214–216, February 1987.

[81] D. Macq and P. Jespers, "Charge injection in current copier cells," *Electronics Letters*, vol. 29, pp. 780–781, April 1993.

[82] H.K. Yang and E.I. El-Masry, "Clock feedthrough analysis and cancellation in current sample/hold circuits," *IEE Proc. Circuits Devices Syst.*, vol. 141, pp. 510–516, December 1994.

[83] H.C. Yang , T.S. Fiez, and D.J. Allstot, "Current-feedthrough effects and cancellation techniques in switched-current circuits," *IEEE International Symposium on Circuits and Systems*, pp. 3186–3188, May 1990.

[84] C. Eichenberger, *Charge injection in MOS integrated sample-and-hold and switched-capacitor circuits.* Germany: Hartung-Gorre, 1989.

[85] C. Eichenberger and W. Guggenbuhl , "On charge injection in analog MOS switches and dummy switch compensation techniques," *IEEE Transactions on Circuits and Systems*, vol. 37, pp. 256–264, February 1990.

[86] E. Vittoz, "Dynamic analog techniques," in *Design of MOS VLSI circuits for telecommunications* (Tsividis and Antognetti, eds.), Englewood Cliffs, New Jersey: Printice-Hall, 1985.

[87] J. Crawley and G. W. Roberts, "Predicting harmonic distortion in switched-current memory circuits," *IEEE Trans. on Circuits and Systems*, vol. 41, pp. 73–86, February 1994.