

Analog Simulation: Circuit level (including “RF” methods and noise) and Behavioral level

Jaijeet Roychowdhury and Alan Mantooth

1 Introduction

Circuit simulation has always been a crucial component of analog system design and is becoming even more so today. Ever-shrinking DSM technologies are resulting in both analog and digital designs becoming less ideal. Indeed, the separation between digital and analog design is becoming blurred; from the digital standpoint, analog effects, often undesired, are becoming pervasive. Integrated RF and communication system design, involving both analog and digital components on the same substrate, now constitutes an important part of the semiconductor industry’s growth, while traditionally digital circuits (such as microprocessors) are now critically limited by analog effects such as delays, the need to synchronize internal busses, and so on. In short, advances in analog, RF, digital and mixed-signal design over the last few years, combined with the effects of shrinking technologies, have spurred a renaissance in simulation. Old simulation problems have assumed renewed significance and new simulation challenges — in some cases already addressed by novel and elegant algorithmic solutions — have arisen.

Every circuit designer is familiar with the program SPICE, the original circuit simulation program released into the public domain in the mid 1970s by the University of California at Berkeley. The basic algorithms and capabilities within SPICE engendered an entire industry of circuit simulation tools, with noted commercial offerings such as HSPICE, PSPICE, and SPECTRE enjoying widespread adoption. Several of these programs have evolved from originally providing roughly the same capabilities as Berkeley SPICE into tools with significantly more advanced simulation capabilities. Furthermore, the aforementioned resurgence of interest and research in advanced circuit simulation algorithms promise further powerful capabilities which will enable designers to make higher-performance circuits and systems faster, better, and with fewer mistakes.

In this chapter, we provide a quick tour of modern circuit simulation. The style we adopt attempts to combine mathematical rigour with physical intuition, in order to provide a simple, clean exposition that links the various facets of the subject logically. We start with showing how one can write circuit equations as nonlinear differential equations and comment on why it is a good idea to do so. Next, we delve into “device models” — *i.e.*, the equations that describe current-voltage and charge-voltage relationships of the semiconductor devices that pervade most modern circuits. We then show how the basic circuit analyses can be simply derived as specific ways of solving the circuit’s differential equations. We touch on an important feature typical of circuit equations — sparsity — that makes these analyses computationally feasible for large circuits, hence practically useful.

Next, motivated by recent interest in RF circuit simulation, we review more advanced analysis techniques for the circuit’s differential equations, starting with the computation of periodic steady states. We explain two alternative methods in the frequency and time domains — Harmonic Balance (HB) and shooting — for computing periodic steady states that have complementary numerical properties. We show how the equations involved in these analyses lose sparsity, hence can be much more difficult to solve computationally than those for the basic analyses above. We then explain how methods based on preconditioned iterative solution of matrix equations can be used to alleviate the computational problem to a large extent.

A recurring theme in circuit simulation is the simultaneous presence of fast and slow rates of time variation, which presents challenges for basic analyses such as transient simulation. We show how the use of multiple artificial time scales can be used to separate fast/slow behaviour in a fundamental manner at the circuit equation level, leading to a special circuit equation form called Multitime Partial Differential Equations (MPDEs). We then outline how MPDEs can be solved numerically using different techniques to solve fast/slow problems efficiently. We touch on special MPDE forms for oscillators and also outline how MPDEs can be used as a link to enable automatic macromodelling of time-varying systems.

Finally, we touch on the important issue of (statistical) noise analysis of circuits. We first outline the fundamental concepts of basic stationary noise analysis and explain the kinds of circuits it applies to. Then we show how stationary noise analysis is insufficient for noise analysis of RF, switching, and other nonlinear circuits. We explain the fundamental concepts of cyclostationary noise and outline the use of these concepts for the noise analysis of RF circuitry, including computational aspects. Finally, we touch on the important problem of noise analysis of oscillators – in particular, calculating phase noise and jitter in oscillators.

2 Differential-Algebraic Equations for Circuits via Modified Nodal Analysis

The fundamental premise behind virtually all types of computer simulation is to first abstract a physical system's behaviour using appropriate mathematical representations and then to use algorithms to solve these representations by computer. For most kinds of circuits, the appropriate mathematical abstraction turns out to be *systems of differential equations*, or more precisely, systems of nonlinear, differential-algebraic equations (DAEs). In the circuit context, the following special DAE form can be shown to be appropriate:

$$\dot{q}(x) + f(x) = b(t). \quad (1)$$

In (1), all quantities (except t , the time variable) are vectors of size n , which is (roughly) the size of the circuit represented by the DAE. The nonlinear vector functions $q(\cdot)$ and $f(\cdot)$ represent the charge/flux and resistive parts of the circuit, respectively, while $b(t)$ is a forcing term representing external inputs from independent current and voltage sources. This DAE form is obtained by writing out the fundamental charge and potential conservation equations of the circuit — Kirchoff's Current Laws (KCL) and Kirchoff's Voltage Law (KVL) — together with the Branch Constitutive Relationships (BCR) of the individual elements. We refer the reader to *e.g.*, [1, 48, 49], for a systematic exposition of how these equations can be obtained for a circuit topology. It is not difficult, however, to understand the basic procedure of defining the quantities in (1) and to apply it to any circuit of interest, “by hand” or via a computer procedure. Consider the simple circuit shown in Figure 1, consisting of a voltage source driving a parallel-RC load through a nonlinear diode. The electrical quantities of interest in this circuit are the two node voltages $v_1(t)$ and $v_2(t)$, as well as the current through the voltage source, $i(t)$.

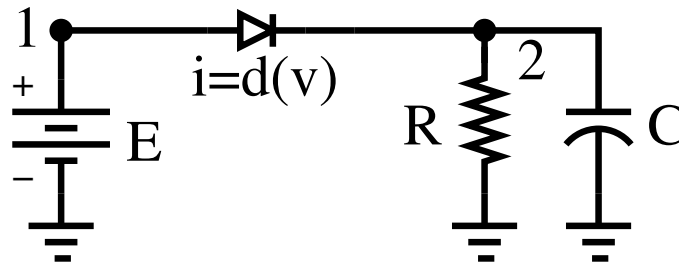


Figure 1: Simple diode-resistor-capacitor circuit.

The resistor's action on the circuit is defined by its current-voltage relationship (Ohm's Law), $i_R(t) = \frac{v_2(t)}{R}$; similarly, the capacitor's current-voltage relationship is given by $i_C(t) = \frac{d}{dt}(Cv_2(t))$. The current-voltage relationship of the diode element,

which is nonlinear, is given by (see Section 3.2)

$$i = d(v_1 - v_2) = I_s \left(e^{\frac{v_1 - v_2}{V_T}} - 1 \right), \quad (2)$$

where I_s and V_T are constant numbers. Finally, the voltage source is captured simply by its setting the voltage of node 1 to its value: $v_1(t) = E$. These four relationships, one for each of the four circuit elements, are the BCRs relevant to this circuit.

By summing currents at each node of interest, the circuit equations are obtained. At node 1, we obtain

$$i(t) + d(v_1 - v_2) = 0, \quad (3)$$

while at node 2, we have

$$\frac{d}{dt}(Cv_2) + \frac{v_2}{R} - d(v_1 - v_2) = 0. \quad (4)$$

Finally, we also have (for the voltage source)

$$v_1 = E. \quad (5)$$

These are known as the Modified Nodal Equations (MNA) of the circuit.

By writing (3)–(5) in vector form, the form of (1) can be obtained. First, the voltage/current unknowns are written in vector form as

$$x = \begin{bmatrix} v_1(t) \\ v_2(t) \\ i(t) \end{bmatrix}. \quad (6)$$

Define the vector function $q(\cdot)$ in (1) to be

$$q(x) \equiv \begin{bmatrix} 0 \\ C \\ 0 \end{bmatrix} x = \begin{bmatrix} 0 \\ Cv_2(t) \\ 0 \end{bmatrix}, \quad (7)$$

$f(\cdot)$ to be

$$f(x) \equiv \begin{bmatrix} i(t) + d(v_1 - v_2) \\ \frac{v_2}{R} - d(v_1 - v_2) \\ v_1 \end{bmatrix}, \quad (8)$$

and the vector $b(t)$ to be

$$b(t) = \begin{bmatrix} 0 \\ 0 \\ E(t) \end{bmatrix}. \quad (9)$$

With these definitions, (1) simply represents the circuit equations (3)–(5), but in vector DAE form. The utility of writing circuit equations in this canonical form is that a variety of mathematical and computational techniques may be brought to bear on (1), without having to worry about the details of $f(\cdot)$, $q(\cdot)$ and $b(t)$, so long as they can be realistically assumed to have some simple properties like continuity, smoothness, *etc.*. Indeed, the DAE form (1) is not restricted to circuits; it subsumes virtually all physical systems, including those from mechanical, optical, chemical, *etc.*, applications.

It will be readily understood, of course, that DAEs for typical industrial circuits are usually much larger and more complex than the one for the simple example of Figure 1. Much of the complexity of these equations stems from the complicated equations describing the *semiconductor devices*, such as MOS and bipolar transistors, that populate circuits in large numbers. These equations, called *device models*, are described further in Section 3.

3 Device Models

Many books have been written that describe the details of various device models available in circuit simulators [51, 52, 53, 54, 55]. In these few pages it is not possible to delve into the derivation of device physics that represent any specific device models. Rather, the focus of this section will be on the role models play and constraints placed on models by simulators. This section first describes models in the context of traditional circuit simulation and then touches on two key device models, the diode and MOSFET, which are illustrative of issues relevant to this chapter. In each case, some remarks are made with respect to versions of these models suitable for power electronic applications as well.

3.1 Role of models in circuit simulation

In the context of this chapter, models are the representation of the properties of devices or a group of interconnected devices by means of mathematical equations, circuit representations (i.e., macromodels), or tables. Care has been taken to avoid defining models strictly as electrical devices. In the era of hardware description languages (HDLs), circuit simulation and modeling have expanded beyond the boundaries of passive and active electrical devices [56, 57, 58, 59]. However, for the purpose of this section our focus will remain on electrical device models.

Semiconductor device models such as the MOSFET involve many complicated equations. Often these equations are piecewise defined according to the different regions of operation observed in the device. Timing studies performed on circuit simulations indicate that the majority of computational effort in network analysis is spent on evaluation of these complicated model equations. This is not hard to imagine when "compact" model implementations of advanced MOSFET technologies are over 15,000 lines of C code in SPICE3 [60]. While much of this code involves data structures in the simulator, several thousand lines of code are devoted to describing model behavior alone. As such, models and modeling techniques are as important to addressing simulation efficiency and accuracy as the solution algorithms employed. In fact, the fidelity of simulation results will be dictated by the weakest link among the two, so both are vital to the simulation process.

Figure 2 below depicts a very high level view of the simulation process which specifically pulls out the role models play in the solution of the system. This flow diagram begins with the processing of the input netlist to the simulator and proceeds to the formulation of the system of equations. The system formulation step involves both the interconnectivity of the specific models in the circuit as well as the matrix stamps of the models. The interconnectivity defines what models contribute current to which nodes and thus leads to the matrix statement of KCL. The matrix stamps of the models, which will be described further in Section 4, indicate the specific contributions a model makes to various rows and columns of the matrix. Once the system formulation is complete, the process proceeds to the numerical solution based on the user-specified analysis. In any event, one thing that is common to all analyses (e.g., DC, AC, transient) and solution methods (e.g., Newton-Raphson) is the need to evaluate the models' constitutive relationships at each iteration, subject to any bypass techniques that may be employed in a given algorithm. This is the model evaluation time referred to above. The independent variables indicated in Figure 2 are those variables being solved for by the simulator and are thus given to the models as known values for an iteration. Then, the model is evaluated and provides the dependent variable values back to the simulator for the purpose of determining if the simulator's guess for the independent variable values was the correct solution to the system.

3.2 Diode model

For the model of a basic $p-n$ junction diode, the constitutive relations are derived in [61] and also shown in [51, 62]. Figure 3 shows the basic large-signal model of the diode. The series resistance is a straightforward application of Ohm's law. The current

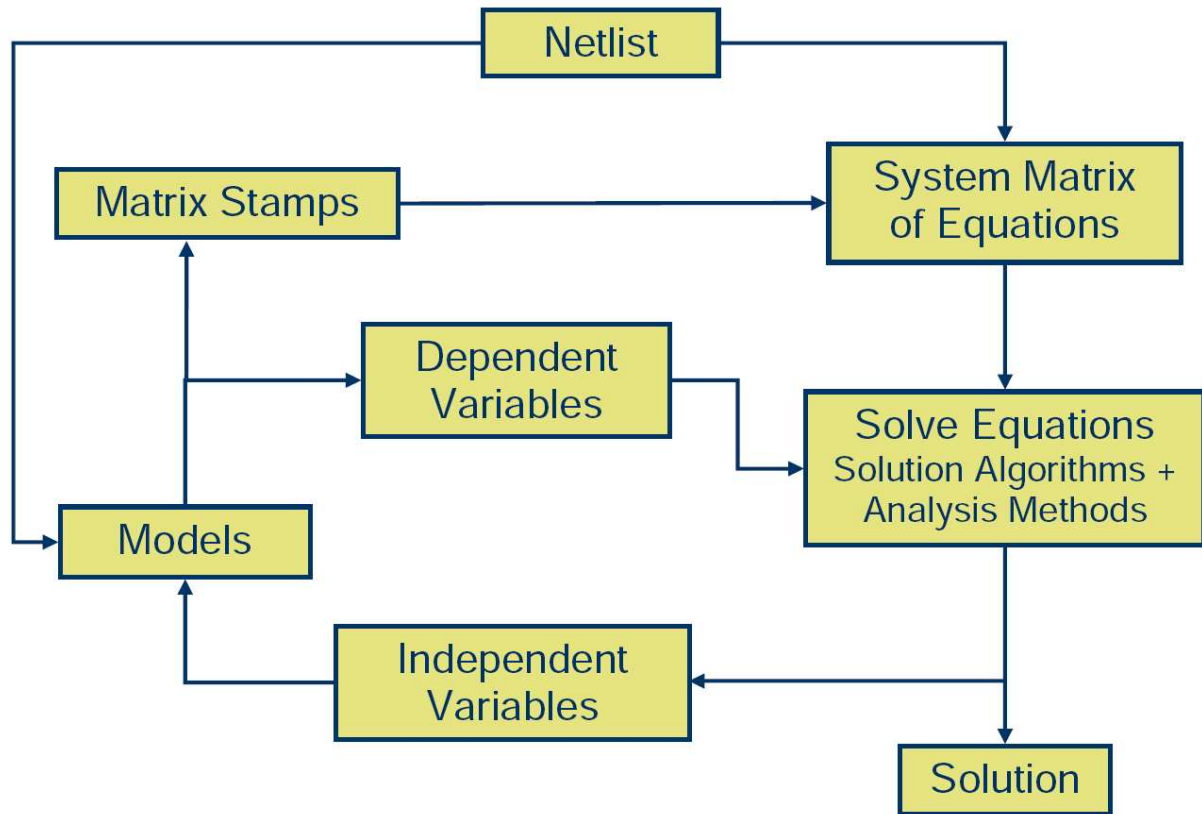


Figure 2: Depiction of the role of models in the traditional circuit simulation process.

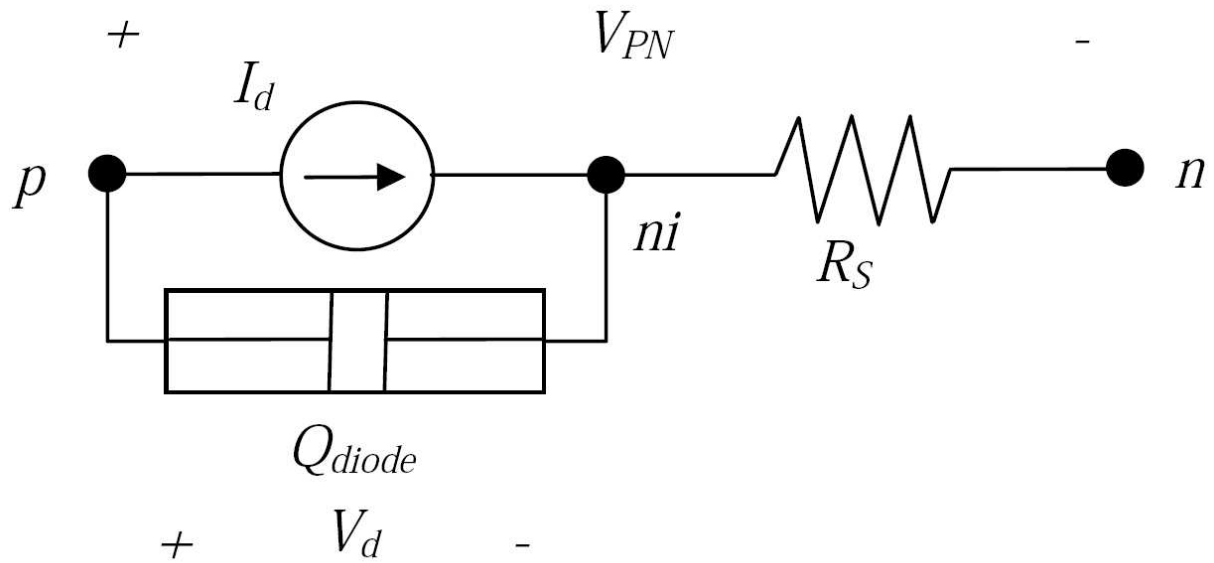


Figure 3: Large-signal model topology of a $p-n$ junction diode.

source is represented by the BCR

$$I_d = I_S \left(e^{\frac{V_d}{nV_T}} - 1 \right), \quad (10)$$

where I_S is the saturation current, V_T is the thermal voltage, V_d is the diode junction voltage, and n is known as the emission coefficient of the diode. The saturation current is a function of the geometry of the device and is given by

$$I_S = Aqn_i^2 \left(\frac{D_p}{L_p N_D} + \frac{D_n}{L_n N_A} \right), \quad (11)$$

where A is the cross-sectional area of the junction, q is the electronic charge, n_i is the intrinsic carrier concentration, N_A and N_D are the acceptor and donor concentrations, L_n (D_n) and L_p (D_p) are the diffusion lengths (diffusion constants) for electrons in p -type material and holes in n -type material, respectively. The thermal voltage is equal to

$$V_T = \frac{kT}{q}, \quad (12)$$

where $k = 1.3806 \times 10^{-23}$ J/K is Boltzmann's constant, T is the absolute temperature in Kelvin, and $q = 1.6022 \times 10^{-19}$ Coulombs is the electronic charge.

The well known relationships in (10)-(12) can be used to illustrate some the options that a model developer has when creating a model for circuit simulation. In this case, a single continuous (and continuously differentiable) equation has been used to describe the BCR. This is the most desirable scenario for the solution algorithms in traditional circuit simulators. Most solvers require continuity in the first derivative of the BCRs in order to avoid convergence problems associated with the Jacobian. Saber [63] is an exception to this requirement, but even in that case models that possess continuous first derivatives simulate faster and more robustly than those with only continuous BCRs. From a pragmatic perspective fewer convergence problems are introduced with BCRs that are not piecewise defined [55]. A second option that faces the model developer is the degree of physics to represent in the model. Most diode models simply specify the saturation current as a parameter and avoid calculating it from (11). This allows diodes to be effectively modeled without the need to know doping concentrations and other low level process information about how the device was constructed. It still remains possible to make the saturation current scalable with area by simply providing a scale factor for I_S in the model equations. Then, since I_S varies so significantly with temperature, primarily due to the n_i^2 factor in (11), an empirical relationship is commonly used to model the variation of I_S with temperature as observed from data sheets or measurements.

The governing equation for the charge in the diode in Figure 3 is

$$Q_{\text{diode}} = Q_{\text{diff}} + Q_{\text{depl}}, \quad (13)$$

where Q_{diode} is the total charge, Q_{diff} is the charge stored due to diffusion in the diode, and Q_{depl} is the charge stored in the depletion region of the junction. The relationships [51] for each of these components of charge are given by

$$Q_{\text{diff}} = \tau_T \cdot I_d, \quad (14)$$

where τ_T is known as the mean transit time of the diode and is typically a model parameter. This charge storage effect is only significant in forward bias and thus a *piecewise* definition for the charge is typically created at this point by setting $Q_{\text{diff}} = 0$ for reverse bias voltages. This charge equation is continuous as is its derivative.

The depletion charge is also defined piecewise as follows:

$$Q_{\text{depl}} = \begin{cases} V_{j0} \cdot C_{j0} \cdot \frac{\left(1 - \frac{V_d}{V_{j0}}\right)^{1-m}}{m-1}, & V_d < f_c \cdot V_{j0}, \\ V_{j0} \cdot C_{j0} \cdot \frac{(1-f_c)^{1-m}}{m-1} + \frac{C_{j0}}{(1-f_c)^{1+m}} \cdot \left[1 - f_c \cdot (1+m) \cdot (V_d - f_c \cdot V_{j0}) + m \frac{V_d^2 - (f_c \cdot V_{j0})^2}{2 \cdot V_{j0}} \right], & V_d \geq f_c \cdot V_{j0}. \end{cases} \quad (15)$$

f_c is a fitting parameter (typically equal to 0.5), m is the grading coefficient of the junction (equals 0.5 for a step junction), C_{j0} is the zero-bias junction capacitance, and V_{j0} is the built-in junction potential. In this case a great deal of care is taken to produce a continuous charge and a continuous capacitance (i.e., first derivative) for (15). The diode model possesses nonlinear behavior in the current-voltage relationships as well as the charge relationships. Details of how such behavior is "stamped" into the circuit matrices and thus used in the flow of Figure 2 will be covered in the next section.

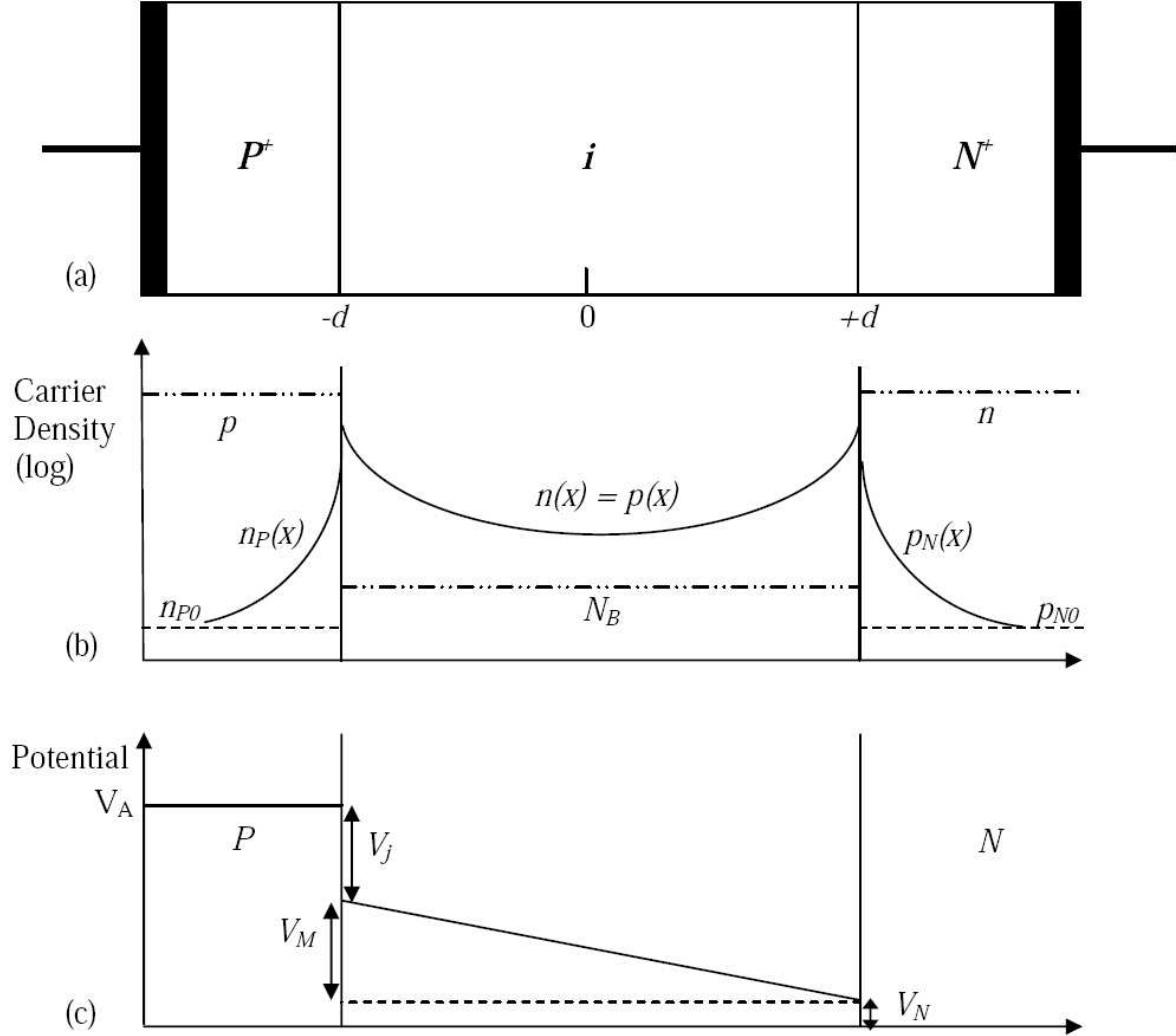


Figure 4: (a) Cross-sectional diagram of a $p-i-n$ power diode, (b) corresponding carrier distribution profile and (c) potential distribution for high-level injection conditions.

The modeling of power diodes is conceptually the same as that for low-voltage diodes. The primary differences arise from the different conditions the device is subjected to and the physical structure of the device. These differences lead to a different set of assumptions for deriving the model. In the end, this does have an impact on the nature of the equations that must be implemented. In the case of power diode models, the levels of injection are significantly higher requiring the model developer to account for ambipolar diffusion effects and pay particular attention to conductivity modulation within lightly doped regions. Some power diodes are manufactured with a lightly doped "base" region between the p and n regions making a $p-v-n$ or a $p-\pi-n$ diode. Such diodes are generically referred to as $p-i-n$ diodes where the i refers to an almost intrinsic region. Figure 4 is a cross-

sectional diagram of a $p-i-n$ diode. In these devices, holes are injected from the p region and electrons are injected from the n region into the base. The normally high resistance region of the base becomes saturated with carriers and the onstate resistance of the region becomes quite low. When the device is switched off the carriers must be removed from the base region before the device can effectively turn off.

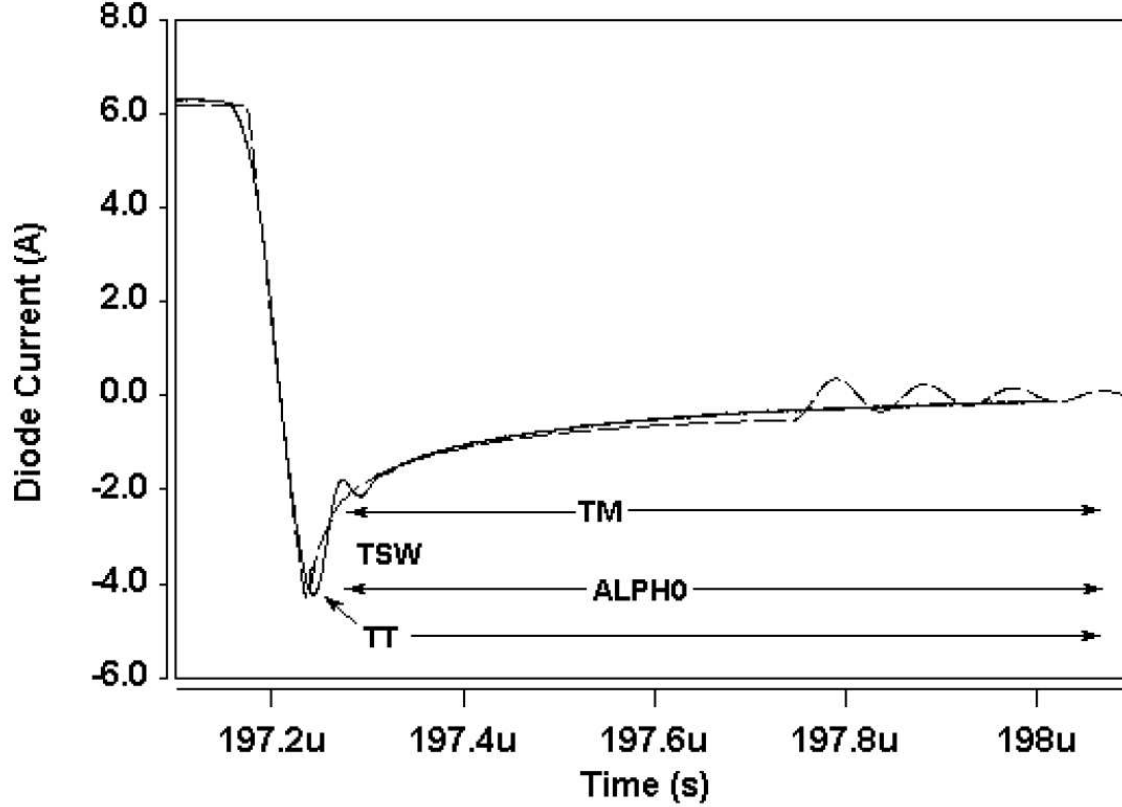


Figure 5: Reverse recovery waveform for a silicon carbide $p-i-n$ power diode [69].

Through mechanisms such as recombination and carrier sweep-out effects, the device has a significant reverse recovery where the current through the device changes from positive to negative and then recovers back to near zero (Figure 5). The depletion region that forms during this recovery process changes width producing a non-quasi-static effect that must also be accounted for to model reverse recovery accurately. The diode model in [64] possesses all of this behavior where typical SPICE diode models do not.

Figure 6 shows the large-signal topology of the power diode model in [64]. The dc characteristics are far more complicated than those described above for the low-voltage diode. Due to the injection levels there are four separate exponential relationships that are used to model low-level recombination, normal injection, high-level injection, and emitter (i.e., end region) recombination effects, respectively. All of this dc behavior is represented by the current I_d represented by the diode symbol in Figure 6. An additional series resistance effect to R_S is R_{MOD} , which represents the conductivity modulation of the base region. The nonlinear charge effects are represented by four different charges in parallel with the diode current in Figure 6. The capacitance C_j is the voltage-dependent diode junction capacitance, Q_{SW} and Q_R represent the two-time constant response of the base charge, and C_r is the moving boundary redistribution capacitance. The total diode voltage v_d is the sum of the contact voltage v_r , the junction voltage v_j , and the mid-region voltage v_m .

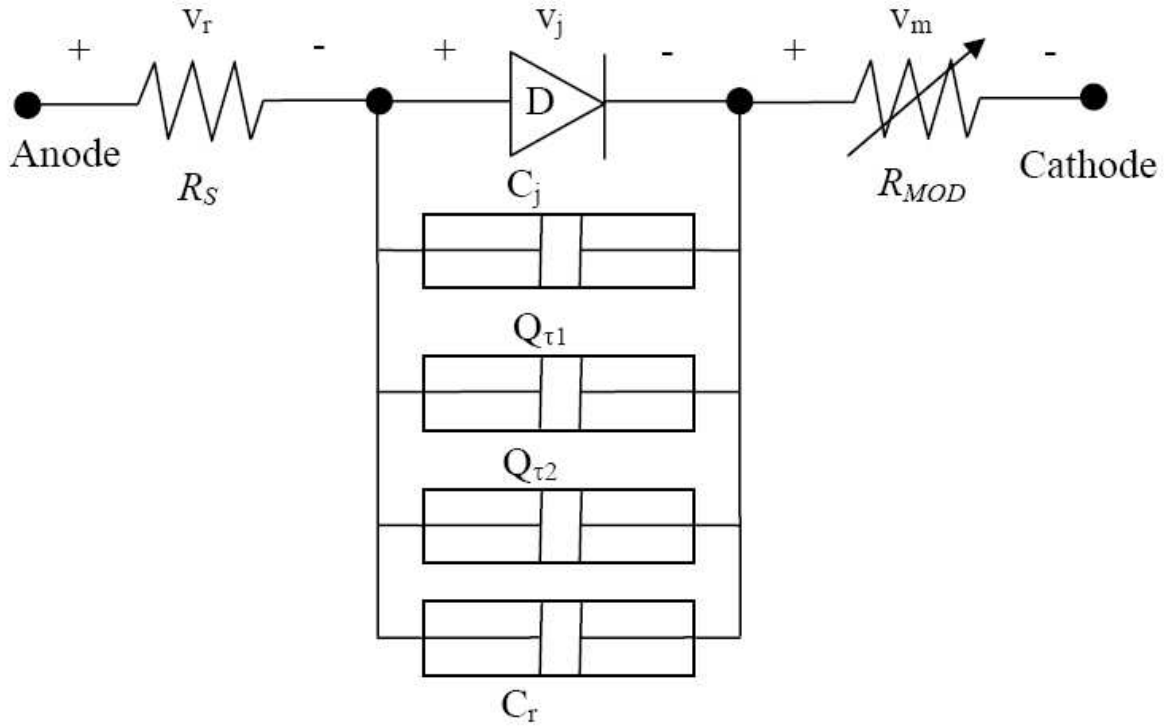


Figure 6: Large-signal model topology of power diode.

3.3 MOSFET models

The most commonly used device model in circuit simulation is the MOSFET model because of the volume of integrated circuits that are designed using this workhorse device. MOS technology has undergone rapid advances that have enabled the semiconductor industry to keep pace with Moore's Law. As a result of these advances MOSFET models have been an ongoing subject of research for the past two decades. Five versions of the BSIM model have been developed from UC Berkeley's research team [65]. Many semiconductor companies have had impressive MOSFET modeling efforts including Texas Instruments, Motorola, Philips, and IBM to name a few. Other research efforts have made impacts on device modeling including the EKV model [66] and the latest efforts on the surface potential (SP) model [67]. It is well beyond the scope of this section to delve into any significant details of these or any other leading edge MOSFET model. Our intent here is merely to provide another example of how linear and nonlinear BCRs arise in device models and segue into how such nonlinearities are dealt with in simulators. For this purpose we will illustrate with a simple MOSFET model with nonlinear drain-source current and charges inspired by the work in [68].

The MOSFET equations for the three regions of drain-source current i_{DS} are given below:

$$i_{DS} = \begin{cases} 0, & v_{GS} - V_t \leq 0 \quad (\text{cut-off}), \\ \frac{\beta}{2} \cdot (v_{GS} - V_t)^2, & v_{DS} \geq v_{GS} - V_t > 0 \quad (\text{saturation}), \\ \beta \left((v_{GS} - V_t) \cdot v_{DS} - \frac{1}{2} v_{DS}^2 \right), & v_{GS} - V_t > v_{DS} \text{ and } v_{GS} - V_t > 0 \quad (\text{triode}). \end{cases} \quad (16)$$

where V_t is the effective threshold voltage and $\beta = k' \frac{W}{L} (1 + \lambda v_{DS})$.

The governing equations for the bulk-drain and bulk-source junctions are similar to those provided in the previous section, but typically do not possess series resistance effects. The governing equations for the intrinsic MOSFET charge are taken from [68]

over four regions of operation as shown below.

Below flatband: $v_{GB} \leq V_{fb}$

$$\begin{aligned} Q_{\text{gate}} &= C_{ox} (v_{GB} - V_{fb}), \\ Q_{\text{bulk}} &= -Q_{\text{gate}}, \\ Q_{\text{drain}} &= 0, \\ Q_{\text{source}} &= 0. \end{aligned} \tag{17}$$

Below threshold: $v_{GDt} \leq v_{GSf} \leq 0$

$$\begin{aligned} Q_{\text{gate}} &= C_{ox} \cdot \gamma \cdot \left(\sqrt{\frac{\gamma^2}{2} + v_{GB} - V_{fb}} - \frac{\gamma}{2} \right), \\ Q_{\text{bulk}} &= -Q_{\text{gate}}, \\ Q_{\text{drain}} &= 0, \\ Q_{\text{source}} &= 0. \end{aligned} \tag{18}$$

Saturation: $v_{GDt} \leq 0$ and $v_{GSf} > 0$

$$\begin{aligned} Q_{\text{gate}} &= C_{ox} \cdot \left(\frac{2}{3} v_{GSf} + (V_t - V_{bi}) \right), \\ Q_{\text{bulk}} &= -C_{ox} (V_t - V_{bi}), \\ Q_{\text{drain}} &= -\frac{4}{15} C_{ox} \cdot v_{GSf}, \\ Q_{\text{source}} &= -\frac{2}{5} C_{ox} \cdot v_{GSf}. \end{aligned} \tag{19}$$

Triode: $v_{GDt} \geq 0$ and $v_{GSf} > v_{GDt}$

$$\begin{aligned} Q_{\text{gate}} &= C_{ox} \left[V_t - V_{bi} + \frac{2}{3} \left(v_{GSDt} + v_{GSf} - \frac{v_{GSf} v_{GDt}}{v_{GSf} + v_{GDt}} \right) \right], \\ Q_{\text{bulk}} &= -C_{ox} (V_t - V_{bi}), \\ Q_{\text{drain}} &= -\frac{C_{ox}}{3} \left[\frac{1}{5} v_{GDt} + \frac{4}{5} v_{GSf} + v_{GDt} \left(\frac{v_{GDt}}{v_{GSDt}} \right) + \frac{1}{5} \frac{v_{GSf} v_{GDt} (v_{GDt} - v_{GSf})}{v_{GSDt}^2} \right], \\ Q_{\text{source}} &= -\frac{C_{ox}}{3} \left[\frac{1}{5} v_{GSf} + \frac{4}{5} v_{GDt} + v_{GSf} \left(\frac{v_{GSf}}{v_{GSDt}} \right) + \frac{1}{5} \frac{v_{GSf} v_{GDt} (v_{GSf} - v_{GDt})}{v_{GSDt}^2} \right]. \end{aligned} \tag{20}$$

In the above, $v_{GDt} = v_{GD} - V_t$, $v_{GSDt} = v_{GSf} - v_{GDt}$, $V_{bi} = V_{fb} + 2\phi_f$, and $C_{ox} = \epsilon_{Si} W \cdot L / t_{ox}$.

The large-signal model topology of this simple MOSFET model is given in Figure 7. In addition to the drain-source current, body diodes, and nonlinear charges, some linear capacitances are shown as well as series resistances on the drain and source.

As in the case of the power diode, several additional effects must be taken into account when dealing with power MOSFET models. The device structure is very different in that power MOSFETs are typically vertical conduction devices where the source and gate are on the top surface and the drain is the back surface of the die or wafer. Many of the same issues arise as they did in the case of the power diode. Higher levels of injection, conductivity modulation of a drift layer, and moving boundary conditions are just some of the major effects that must be modeled.

4 Basic Circuit Simulation: DC analysis

In Section 2, we noted how any circuit could be written as a set of nonlinear differential equations (1), while in Section 3, we looked into the details of some of the more complex equations involved, notably those from the semiconductor models. We now

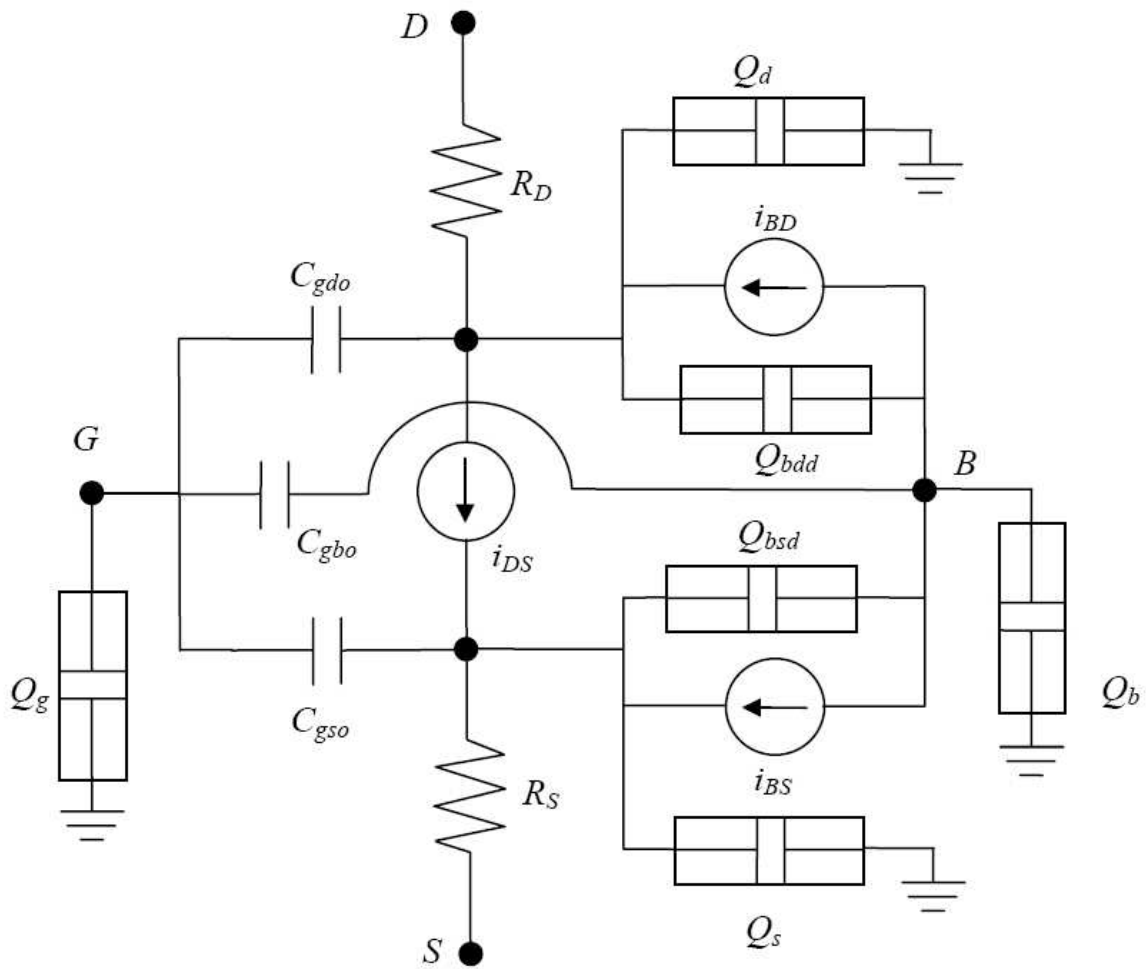


Figure 7: Large-signal model topology of a MOSFET.

look at how to solve (1) numerically for a basic design problem: finding the DC operating point of the circuit, *i.e.*, solving the circuit when all voltages/currents within it remain constant with time.

Finding the DC operating point is a fundamental and basic requirement in the design of virtually every circuit. The voltages and currents in a circuit at its DC operating point provide important information about correct functioning of the circuit. The DC operating point is essential not only as a first basic check of circuit operation, but is also a prerequisite for further analyses. Small-signal AC analysis, transient analysis, noise analysis, *etc.*, all rely on a prior DC operating point having been calculated; furthermore, the operating point is also useful for steady-state and envelope analyses.

When no quantity in (1) is changing with time, the input $b(t)$ must be a constant vector b_{DC} ; and the circuit variables $x(t)$ must similarly be unchanging, *i.e.*, $x(t) \equiv x_{DC}$. When nothing changes with time, all derivatives become zero, hence (1) becomes

$$f(x_{DC}) = b_{DC}, \quad \text{or} \quad g(x_{DC}) \equiv f(x_{DC}) - b_{DC} = 0. \quad (21)$$

This is a *nonlinear equation system*, where the unknown to be solved for is the vector x_{DC} . Solving such a system of equations is general a non-trivial numerical task. In circuit simulation applications, a numerical method known as the *Newton-Raphson algorithm* is most often used to solve this system.

4.0.1 The Newton-Raphson algorithm for nonlinear equation solution

The Newton-Raphson (NR) algorithm (*e.g.*, [50]) is a technique that is widely used in many disciplines for finding solutions of nonlinear equations. It is an *iterative* method — that is to say, it starts from a guess for the solution x_0 of (21), and keeps refining the guess until it finds a solution. It is important to note that there is no *guarantee* that NR will succeed in finding a solution, although it often does so in practice. Therefore, the algorithm is typically implemented to announce failure after some number of iterations (*e.g.*, after 100 tries).

NR relies on having available to it two functionalities related to the nonlinear function $g(x)$ in (21):

1. it needs a means to evaluate $g(x)$, given any x ; and
2. it needs a means to evaluate the derivative of $g(\cdot)$, *i.e.*, $J(x) = \frac{dg(x)}{dx}$, given any x . Note that since both x and $g(x)$ are vectors of size n , J is an $n \times n$ matrix. The matrix J is known as the *Jacobian matrix* of $g(x)$.

The following is an outline of the basic NR algorithm, in MATLAB-like pseudo-code:

```

1  function solution = NR(xguess)
2      x = xguess;
3      finished = 0;
4      epsilon = 1e-10;
5      while (~finished)
6          gx = g(x);
7          if (norm(gx) < epsilon)
8              finished = 1;
9              solution = x;
10             break;
11         end
12         Jx = dgdg(x);
13         delta_x = - inverse(Jx)*gx;
```

```

14
15             x = x + delta_x;
16         end

```

Observe that a key step is line 13, where the current guess for the solution is updated by an amount Δx , calculated by solving

$$J(x) \cdot \Delta x = -g(x), \quad \text{or} \quad \Delta x = -J^{-1}(x) \cdot g(x). \quad (22)$$

The above involves solving a *linear matrix equation*, with the Jacobian matrix $J(x)$, at every iteration of the NR algorithm. The computational ease with which this can be done depends on the structure of the Jacobian matrix, which we discuss in more detail below.

4.0.2 Derivative (“Jacobian”) matrices and device “stamps”

To illustrate the structure of the Jacobian matrix, consider again the example of Figure 1, together with its $f(x)$ defined in (8). The Jacobian matrix of $f(x)$ in (8) is

$$J(x) = \begin{pmatrix} d'(v_1 - v_2) & -d'(v_1 - v_2) & 1 \\ -d'(v_1 - v_2) & \frac{1}{R} + d'(v_1 - v_2) & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad (23)$$

where $d'(\cdot)$ is the derivative of the diode’s current-voltage relationship, defined in (2). Observe, firstly, that each element in the circuit has a characteristic pattern of entries that it contributes to the Jacobian matrix. For example, the resistor R from node 2 to ground contributes the pattern

$$\begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & \frac{1}{R} & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix}, \quad (24)$$

while the diode between nodes 1 and 2 contributes the pattern

$$\begin{pmatrix} d'(v_1 - v_2) & -d'(v_1 - v_2) & \cdot \\ -d'(v_1 - v_2) & d'(v_1 - v_2) & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix}. \quad (25)$$

The pattern of Jacobian entries contributed by each element is called its *matrix stamp*. The Jacobian matrix $J(x)$ thus consists of the addition of the stamps of all the elements in the circuit.

4.0.3 Jacobian sparsity and its importance

Observe also that several entries in the Jacobian matrix in (23) are zero. As the size of the circuit grows larger, the number of zero entries in the matrix predominates. The reason for this is easy to see – if the circuit has roughly n nodes and each node is connected to only 2 or 3 circuit elements (as is typical), the total number of nonzero matrix stamps is of the order of $2n$ or $3n$. As a result, the remaining entries of the matrix (which has a total of n^2 entries) must be zero or “unstamped”. Such matrices, where there are relatively few nonzero entries and many zero entries, are called *sparse matrices*. When a matrix has no zero entries, it is called *dense*. We have just noted that because each node in typical circuits is connected only to a few elements, Jacobian matrices of circuits tend to be sparse.

The fact that circuit matrices are usually sparse is of enormous importance for circuit simulation. The reason is that, while the Newton update step in (22) is in general very computationally expensive for arbitrary matrices J (especially if J is dense), efficient

techniques, collectively dubbed “sparse matrix technology”, exist for solving for Δx in (22) when J is sparse. More specifically, the computational complexity of general linear equation solution (*e.g.*, when J is dense) is $O(n^3)$, while that for typical sparse J is $O(n)$. When circuit sizes n reach the 1000s or 10s of 1000s, this difference in computational complexity is extremely significant from a practical standpoint. Thus, sparsity of circuit matrices is critical for enabling efficient, practically effective, linear solution for circuits. Although so far, we have noted the importance of this linear solution only for DC analysis (as part of the NR loop in line 13), we will see that exploiting sparsity is critical for virtually all other analyses related to circuits.

5 Steady-state analysis

It is often important in RF design to find the periodic steady state of a circuit driven by one or more periodic inputs.

For example, a power amplifier driven to saturation by a large single-tone input is operating in a periodic steady state. A variant is the quasiperiodic steady state, *i.e.*, when the circuit is driven by more than one signal tone; for example, an amplifier driven by two closely spaced sinusoidal tones at 1GHz and 990Mhz. Such excitations are closer approximations to real-life signals than pure tones, and are useful for estimating intermodulation distortion.

The workhorse of analog verification, SPICE (and its derivatives), can of course be applied to find the (quasi)periodic steady state of a circuit, simply by performing a time-stepping integration of the circuit’s differential equations (“transient simulation”) long enough for the transients to subside and the circuit’s response to become (quasi)periodic. This approach has several disadvantages, however. In typical RF circuits, the transients take thousands of periods to die out, hence the procedure can be very inefficient. Further, harmonics are typically orders of magnitude smaller than the fundamental, hence long transient simulations are not well suited for their accurate capture, because numerical errors from time-stepping integration can mask them. These issues are exacerbated in the presence of quasiperiodic excitations, because simulations need to be much longer – *e.g.*, for excitations of 1GHz and 990Mhz, the system needs to be simulated for thousands of multiples of the common period, $\frac{1}{10\text{MHz}}$, yet the simulation timesteps must be much smaller than 1ns, the period of the fastest tone. For these reasons, more efficient and accurate specialised techniques have been developed. We will focus on two different methods with complementary properties, harmonic balance (HB) and shooting.

5.1 Harmonic Balance and Shooting

In the well-known method of *Harmonic Balance (HB)* (*e.g.*, [2, 3, 4, 5, 6, 7, 8, 9, 10]), $x(t)$ and $b(t)$ of (1) are expanded in a Fourier series. The Fourier series can be one-tone for periodic excitations (*e.g.*, 1GHz and its harmonics) or multi-tone in the case of quasiperiodic excitations (eg, 1GHz and 990Mhz, their harmonics, and intermodulation mixes). The DAE is rewritten directly in terms of the Fourier coefficients of $x(t)$ (which are unknown) and of $b(t)$; the resulting system of nonlinear equations is larger by a factor of the number of harmonic/mix components used, but are algebraic (*i.e.*, there are no differential components). Hence they can be solved numerically using, *e.g.*, the well-known Newton-Raphson method [11].

Example 5.1 We illustrate the one-tone procedure with the following scalar DAE:

$$\dot{x} + x - \epsilon x^2 - \cos(2\pi 1000t) = 0 \quad (26)$$

First, we expand all time-variations in a Fourier series of (say) $M = 3$ terms, *i.e.*, the DC component, fundamental and second harmonic components.

$$x(t) = \sum_{i=-2}^2 X_i e^{j2\pi i 10^3 t}$$

Here, X_i are the unknown Fourier coefficients of $x(t)$. For notational convenience, we express them as the vector $X = [X_2, \dots, X_{-2}]^T$.

Similarly, $x^2(t)$ is also expanded in a Fourier series in t ; the Fourier coefficients of this expansion are functions of the elements of X , which we denote by $F_i(X)$.

$$x^2(t) = \sum_{i=-2}^2 \sum_{k=-2}^2 X_i X_k e^{j2\pi(i+k)10^3 t} = \sum_{i=-2}^2 F_i(X) e^{j2\pi i 10^3 t} + \text{higher terms}$$

In this case, where the nonlinearity is a simple quadratic, F_i can be obtained analytically; but in general, numerical techniques like those used in harmonic balance need to be employed for computing these functions. For convenience, we write $F(X) = [F_2(X), \dots, F_{-2}(X)]^T$.

We also write the Fourier coefficients of the excitation $\cos(2\pi 1000t)$ as a vector $B = [0, \frac{1}{2}, 0, \frac{1}{2}, 0]^T$. Finally, we write the differential term \dot{x} also as a vector of Fourier coefficients. Because the differentiation operator is diagonal in the Fourier basis, this becomes simply ΩX , where $\Omega = j2\pi 1000 \text{diag}(2, 1, 0, -1, -2)$ is the diagonal frequency-domain differentiation matrix.

Invoking the orthogonality of the Fourier basis, we now obtain the HB equations for our DAE:

$$H(X) \equiv \Omega X + X - \varepsilon F(X) - B = 0$$

This is a set of nonlinear algebraic equations in 5 unknowns, and can be solved by numerical techniques such as Newton-Raphson.

The above example illustrates that the size of the HB equations is larger than that of the underlying DAE, by a factor of the number of harmonic/mix components used for the analysis. In fact, the HB equations are not only larger in size than the DAE, but also considerably more difficult to solve using standard numerical techniques. The reason for this is the dense structure of the derivative, or Jacobian matrix, of the HB equations. If the size of the DAE is n , and a total of N harmonics and mix components are used for the HB analysis, the Jacobian matrix has $Nn \times Nn$. Just the storage for the nonzero entries can become prohibitive for relatively moderate values of n and N ; for example, $n = 1000$ (for a medium-sized circuit) and $N = 100$ (e.g., for a two-tone problem with about 10 harmonics each) leads to 10GB of storage for the matrix alone. Further, inverting the matrix, or solving linear systems with it, requires $O(N^3 n^3)$ operations, which is usually infeasible for moderate to large-sized problems. Such linear solutions are typically required as steps in solving the HB equations, for example by the Newton-Raphson method. Despite this disadvantage, HB is a useful tool for small circuits and few harmonics, especially for microwave circuit design. Moreover, as we will see later, new algorithms have been developed for HB that make it much faster for larger problems.

Another technique for finding periodic solutions is the *Shooting* method (e.g., [12, 13, 14, 15]). Shooting works by finding an initial condition for the DAE that also satisfies the periodicity constraint. A guess is made for the initial condition, the system is simulated for one period of the excitation using time-stepping DAE solution methods, and the error from periodicity used to update the initial condition guess, often using a Newton-Raphson scheme.

More precisely, shooting computes the *state transition function* $\Phi(t, x_0)$ of (1). $\Phi(t, x_0)$ represents the solution of the system at time t , given initial condition x_0 at time 0. Shooting finds an initial condition x^* that leads to the same state after one period T of the excitation $b(t)$; in other words, shooting solves the equation $H(x) \equiv \Phi(t, x) - x = 0$.

The shooting equation is typically solved numerically using the Newton-Raphson method, which requires evaluations of $H(x)$ and its derivative (or Jacobian) matrix. Evaluation of $H(x)$ is straightforward using time-stepping, i.e., transient simulation, of (1). However, evaluating its Jacobian is more involved. The Jacobian matrix is of the same size n as the number of circuit equations, but it is dense, hence storage of its elements and linear solutions with it is prohibitive in cost for large problems. In this respect, shooting suffers from size limitations, similar to harmonic balance. In other respects, though, shooting has properties complementary to harmonic balance. The following list contrasts the main properties of HB and shooting.

- **Problem size:** The problem size is limited for both HB and shooting, due to the density of their Jacobian matrices. However, since the HB system is larger by a factor of the number of harmonics used, shooting can handle somewhat larger problems given the same resources. Roughly speaking, sizes of about 40 for HB and 400 for shooting represent practical limits.
- **Accuracy/Dynamic Range:** Because HB uses orthogonal Fourier bases to represent the waveform, it is capable of very high dynamic range – a good implementation can deliver 120dB of overall numerical accuracy. Shooting, being based on time-stepping solution of the DAE with time-steps of different sizes, is considerably poorer in this regard.
- **Handling of nonlinearities:** HB is not well suited for problems that contain strongly nonlinear elements. The main reason for this is that strong nonlinearities (*e.g.*, clipping elements) generate sharp waveforms that do not represent compactly in a Fourier series basis. Hence many harmonics/mix components need to be considered for an accurate simulation, which raises the overall problem size.

Shooting, on the other hand, is well suited for strong nonlinearities. By approaching the problem as a series of initial value problems for which it uses time-stepping DAE methods, shooting is able to handle the sharp waveform features caused by strong nonlinearities quite effectively.

- **Multitone problems:** A big attraction of HB is its ability to handle multitone or quasiperiodic problems as a straightforward extension of the one tone case, by using multitone Fourier bases to represent quasiperiodic signals. Shooting, on the other hand, is limited in this regard. Since it uses time-stepping DAE solution, shooting requires an excessive number of timepoints when the waveforms involved have widely separated rates of change; hence it is not well suited for such problems.

5.2 Fast Methods

A disadvantage of both HB and shooting is their limitation to circuits of relatively small size. This was not a serious problem as long as microwave/RF circuits contained only a few nonlinear devices. Since the mid-90s, however, economic and technological developments have changed this situation. The market for cheap, portable wireless communication devices has expanded greatly, leading to increased competition and consequent cost pressures. This has spurred on-chip integration of RF communication circuits and the reduction of discrete (off-chip) components. On-chip design techniques favour the use of many integrated nonlinear transistors over even a few linear external components. Hence the need has arisen to apply HB and shooting to large circuits in practical times.

To address this issue, so-called *fast* algorithms have arisen to enable both HB and shooting to handle large circuits. The key property of these methods is that computation/memory usage grow approximately linearly with problem size. The enabling idea behind the improved speed is to express the dense Jacobian matrices as sums and products of simpler matrices that are either sparse, or have very regular structure so can be applied/inverted efficiently. Using these expansions for the Jacobian, special solution algorithms called *preconditioned iterative linear solvers* are applied to solve linear equations involving the Jacobian without forming it explicitly.

A detailed description of fast techniques is beyond the scope of this chapter; the interested reader is referred to [7, 9, 16, 10, 15] for further information. Here we outline the main ideas behind these methods in a simplified form using Example 5.1 for illustration, and summarize their main properties.

From Example 5.1, the Jacobian matrix of the HB system is

$$J = \frac{\partial H(X)}{\partial X} = \Omega + I - \epsilon \frac{\partial F(X)}{\partial X}$$

Now, $F(X)$ in this case represents the vector of Fourier coefficients of the nonlinear term $f(x) = x^2$. One way in which these can be computed numerically is to 1) use the inverse Fast Fourier Transform (FFT) to convert the Fourier coefficients X to samples of the time-domain waveform $x(t)$, then 2) evaluate the nonlinear function $f(x) = x^2(t)$ at each of these samples in the time domain, and finally, 3) use the FFT to reconvert the time-domain samples of $f(x)$ back to the frequency domain, to obtain $F(X)$. The derivative of these operations can be expressed as:

$$\frac{\partial F(X)}{\partial X} = DGD^*,$$

where D is a block-diagonal matrix with each block equal to the Discrete Fourier Transform matrix, D^* is its inverse, and G is a diagonal matrix with entries $f'(x)$ evaluated at the time-domain samples of $x(t)$. Hence the overall Jacobian matrix can be represented as:

$$J = \Omega + I - \varepsilon DGD^*.$$

Observe that each of the matrices in this expansion is either sparse or consists of DFT matrices. Hence multiplication of J with a vector is efficient, since the sparse matrices can be applied in approximately linear time, and the DFT matrix and its inverse can be applied in $N \log N$ time using the FFT, where N is the number of harmonics. It is this key property, that multiplications of J with a vector can be performed in almost-linear computation despite its dense structure, that enables the use of preconditioned iterative linear methods for this problem.

Preconditioned iterative linear methods (*e.g.*, [17, 18, 19]) are a set of numerical techniques for solving linear systems of the form $Jc = d$. Modern iterative solvers like QMR [18] and GMRES [17] use Krylov-subspace techniques for superior performance. The key feature of these solvers is that the only way in which J is used is in matrix-vector products Jz . This contrasts with traditional methods for linear solution, which use Gaussian elimination or variants like LU factorizations directly on elements of J . Due to this property of iterative linear solvers, it is not necessary to even form J explicitly in order to solve linear systems with it, so long as a means is available for computing matrix-vector products with it.

As we have seen above, products with the HB Jacobian can be conveniently computed in almost-linear time without having to build the matrix explicitly. Hence preconditioned linear iterative techniques are well-suited to solving the linear systems that arise when solving the nonlinear HB equations using the Newton-Raphson method. If the iterative linear methods use only a few matrix-vector products with the Jacobian to compute the linear system's solution, and the Newton-Raphson is well-behaved, the overall cost of solving the HB equations remains almost-linear in problem size.

An important issue with preconditioned iterative linear solvers, especially those based on Krylov subspace methods, is that they require a good *preconditioner* to converge reliably in a few iterations. Convergence of the iterative linear method is accelerated by applying a preconditioner \tilde{J} , replacing the original system $Jc = d$ with the *preconditioned* system $\tilde{J}^{-1}Jc = \tilde{J}^{-1}d$, which has the same solution. For robust and efficient convergence, the preconditioner matrix \tilde{J} should be, in some sense, a good approximation of J , and also “easy” to invert, usually with a direct method such as LU factorization. Finding good preconditioners that work well for a wide variety of circuits is a challenging task, especially when the nonlinearities become strong.

The ideas behind the fast techniques outlined above are applicable not just to HB but also to shooting [15]. Jacobian matrices from shooting can be decomposed as products and sums of the sparse circuit Jacobians matrices. Preconditioned linear iterative techniques can then applied to invert the Jacobian efficiently.

As an example of the application of the fast methods, consider the HB simulation of an RFIC quadrature modulator reported in [10]. The circuit, of about 9500 devices, was simulated by fast HB with a three-tone excitation, with a baseband signal at 80kHz, and local oscillators at 178MHz and 1.62 GHz. The size of the circuit's DAE was $n = 4800$; the three tones, their harmonics and mixes totalled $N = 4320$ components. Simulating a circuit with these specifications is completely infeasible using traditional HB techniques.

Using fast HB, the simulation required only 350MB of memory, and took 5 days of computation on an SGI 150Mhz R4400 machine. The results of the the simulation are shown in Figure 8.

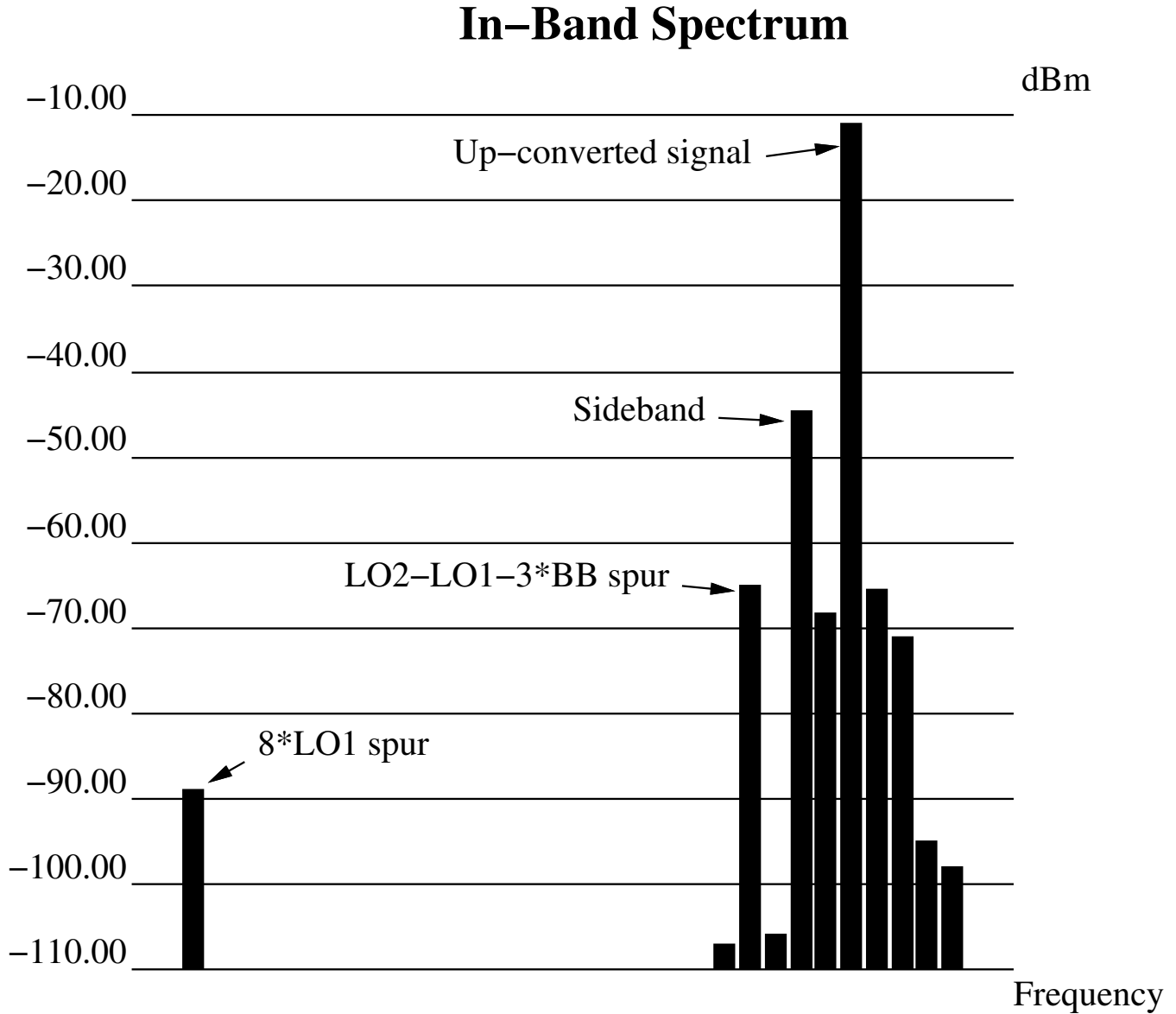


Figure 8: Quadrature modulator spectrum

6 Multitime analysis

In the previous section, we noted that HB and shooting have complementary strengths and weaknesses, stemming from their use of Fourier and time-domain bases, respectively. While HB is best suited for multitone problems that are only mildly nonlinear, shooting is best for single tone problems that can be strongly nonlinear. Neither method is suitable for circuits that have both multitone signals (*i.e.*, widely-separated time scales of variation) *and* strongly nonlinear components.

With greater RF integration, tools that can analyze precisely this combination of circuit characteristics effectively are required.

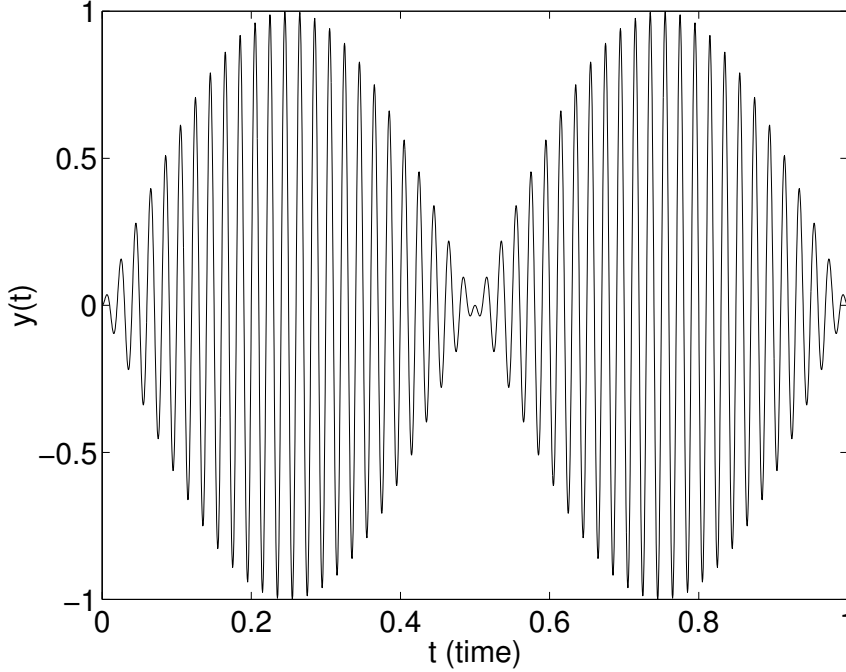


Figure 9: Example 2-tone quasi-periodic signal $y(t)$

In this section, we review a family of techniques, based on partial differential equations (PDEs) using multiple artificial time scales [20, 8, 21, 22, 23, 24], that hold this promise.

Consider the waveform $y(t)$ shown in Figure 9, a simple two-tone signal given by

$$y(t) = \sin\left(\frac{2\pi}{T_1}t\right) \sin\left(\frac{2\pi}{T_2}t\right), \quad T_1 = 0.02\text{s}, \quad T_2 = 1\text{s}. \quad (27)$$

The two tones are at frequencies $f_1 = \frac{1}{T_1} = 50\text{Hz}$ and $f_2 = \frac{1}{T_2} = 1\text{Hz}$, *i.e.*, there are 50 fast-varying cycles of period $T_1 = 0.02\text{s}$ modulated by a slowly-varying sinusoid of period $T_2 = 1\text{s}$.

If each fast cycle is sampled at n points, the total number of time-steps needed for one period of the slow modulation is $n\frac{T_2}{T_1}$. To generate Figure 9, 15 points were used per cycle, hence the total number of samples was 750. This number can be much larger in applications where the rates are more widely separated, *e.g.*, separation factors of 1000 or more are common in electronic circuits. Now consider a multivariate representation of $y(t)$ using two artificial time scales, as follows: for the ‘fast-varying’ parts of $y(t)$, t is replaced by a new variable t_1 ; for the ‘slowly-varying’ parts, by t_2 . The resulting function of two variables is denoted by

$$\hat{y}(t_1, t_2) = \sin\left(\frac{2\pi}{T_1}t_1\right) \sin\left(\frac{2\pi}{T_2}t_2\right). \quad (28)$$

The plot of $\hat{y}(t_1, t_2)$ on the rectangle $0 \leq t_1 \leq T_1$, $0 \leq t_2 \leq T_2$ is shown in Figure 10. Observe that $\hat{y}(t_1, t_2)$ does not have many undulations, unlike $y(t)$ in Figure 9. *Hence it can be represented by relatively few points, which, moreover, do not depend on the relative values of T_1 and T_2 .* Figure 10 was plotted with 225 samples on a uniform 15×15 grid – three times fewer than for Figure 9. This saving increases with increasing separation of the periods T_1 and T_2 .

Further, note that $\hat{y}(t_1, t_2)$ is periodic with respect to both t_1 and t_2 , *i.e.*, $\hat{y}(t_1 + T_1, t_2 + T_2) = \hat{y}(t_1, t_2)$. This makes it easy to recover $y(t)$ from $\hat{y}(t_1, t_2)$, simply by setting $t_1 = t_2 = t$, and using the fact that \hat{y} is bi-periodic. It is easy, from direct inspection of the three-dimensional plot of $\hat{y}(t_1, t_2)$, to visualise what $y(t)$ looks like. As t increases from 0, the path given by $\{t_i = t \bmod T_i\}$ traces the sawtooth path shown in Figure 11. By noting how \hat{y} changes as this path is traced in the t_1 - t_2 plane, $y(t)$ can be traced.

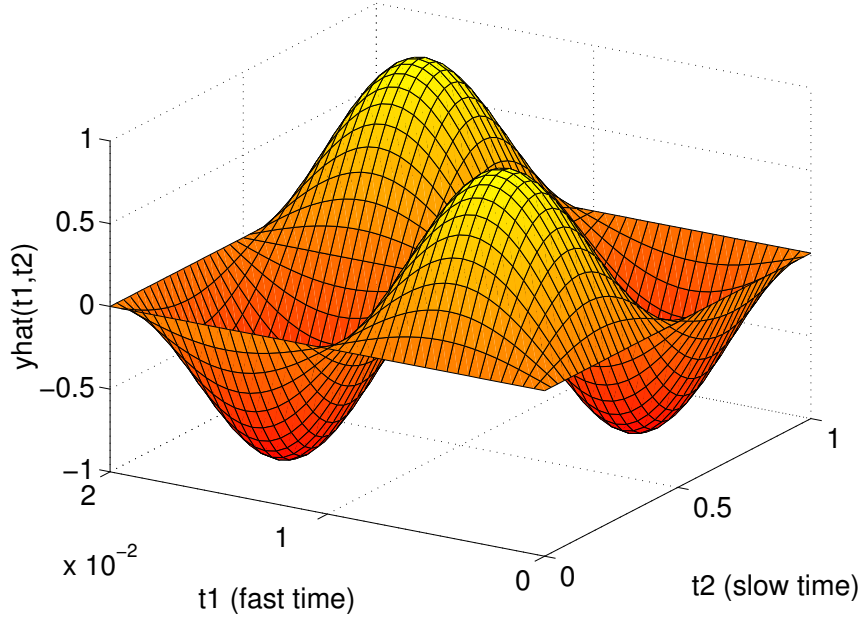


Figure 10: Corresponding 2-periodic bivariate form $\hat{y}(t_1, t_2)$

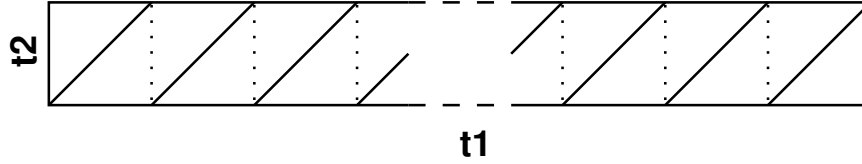


Figure 11: Path in the t_1 - t_2 plane

When the time-scales are widely separated, therefore, inspection of the bivariate waveform directly provides information about the slow and fast variations of $y(t)$ more naturally and conveniently than $y(t)$ itself.

We observe that the bivariate form can require far fewer points to represent numerically than the original quasiperiodic signal, yet it contains all the information needed to recover the original signal completely. This observation is the basis of the partial differential formulation to be introduced shortly. the waveforms in a circuit are represented in their bi-variate forms (or multivariate forms if there are more than two time scales). The key to efficiency is to solve for these waveforms directly, without involving the numerically inefficient one-dimensional forms at any point. To do this, it is necessary to first describe the circuit's equations using the multivariate functions.

If the circuit is described by the differential equations (1), then it

can be shown that if $\hat{x}(t_1, t_2)$ and $\hat{b}(t_1, t_2)$ denote the bi-variate forms of the circuit unknowns and excitations, then the following **Multitime Partial Differential Equation (MPDE)** is the correct generalization of (1) to the bi-variate case:

$$\boxed{\frac{\partial q(\hat{x})}{\partial t_1} + \frac{\partial q(\hat{x})}{\partial t_2} + f(\hat{x}) = \hat{b}(t_1, t_2).} \quad (29)$$

More precisely, if \hat{b} is chosen to satisfy $b(t) = \hat{b}(t, t)$, and \hat{x} satisfies (29), then it can be shown that $x(t) = \hat{x}(t, t)$ satisfies (1). Also, if (1) has a quasi-periodic solution, then (29) can be shown to have a corresponding bi-variate solution.

By solving the MPDE numerically in the time domain, strong nonlinearities can be handled efficiently. Several numerical methods are possible, including discretization of the MPDE on a grid in the t_1 - t_2 plane, or using a mixed time-frequency method

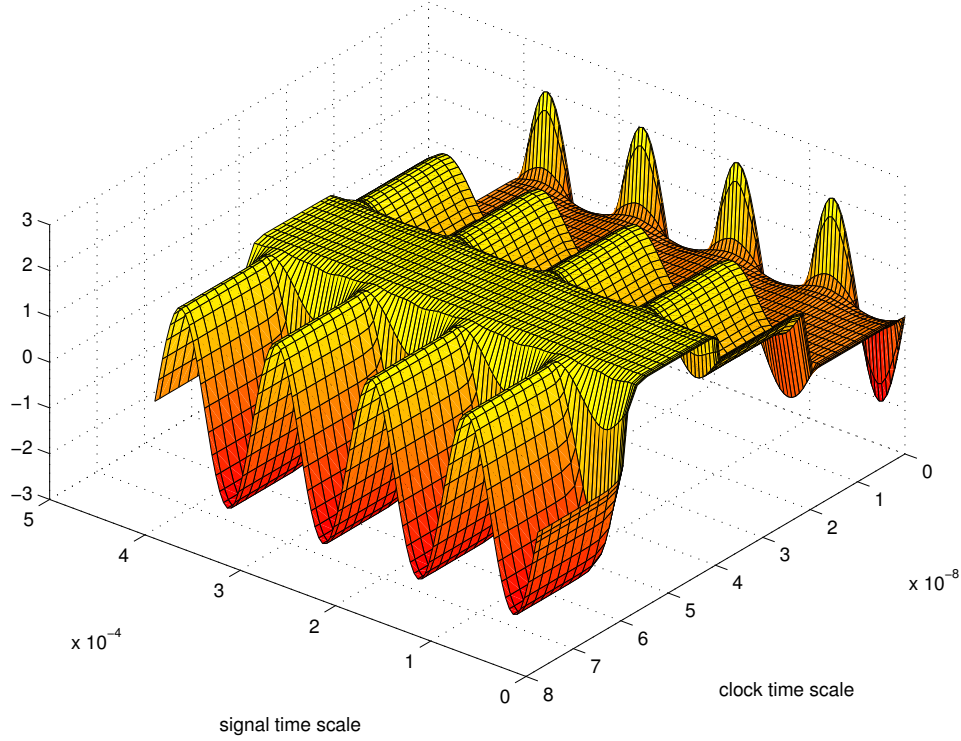


Figure 12: Multi-time output waveform of SC integrator

in which the variation along one of the time scales is expressed in a short Fourier series. Quasiperiodic and envelope solutions can both be generated, by appropriate selection of boundary conditions for the MPDE. Sparse matrix and iterative linear methods are used to keep the numerical algorithms efficient even for large systems.

As an example, Figure 12 depicts the output voltage of a switched-capacitor integrator block, obtained from a multi-time simulation based on the above concepts. The cross-section parallel to the signal time scale represents the envelope of the signal riding on the switching variations. By moving these cross-sections to different points along the clock time-scale, the signal envelope at different points of the clock waveform can be seen.

6.1 Autonomous Systems: the Warped MPDE

When the DAEs under consideration are oscillatory, frequency modulation (FM) can be generated. Unfortunately, FM cannot be represented compactly using multiple time scales as easily as the waveform in Figure 10. We illustrate the difficulty with an example. Consider the following prototypical FM signal

$$x(t) = \cos(2\pi f_0 t + k \cos(2\pi f_2 t)), \quad f_0 \gg f_2, \quad (30)$$

with instantaneous frequency

$$f(t) = f_0 - k f_2 \sin(2\pi f_2 t). \quad (31)$$

$x(t)$ is plotted in Figure 13 for $f_0 = 1\text{MHz}$, $f_2 = 20\text{KHz}$, and modulation index $k = 8\pi$. Following the same approach as for (27), a bivariate form can be defined to be

$$\hat{x}_1(t_1, t_2) = \cos(2\pi f_0 t_1 + k \cos(2\pi f_2 t_2)), \quad \text{with } x(t) = \hat{x}_1(t, t). \quad (32)$$

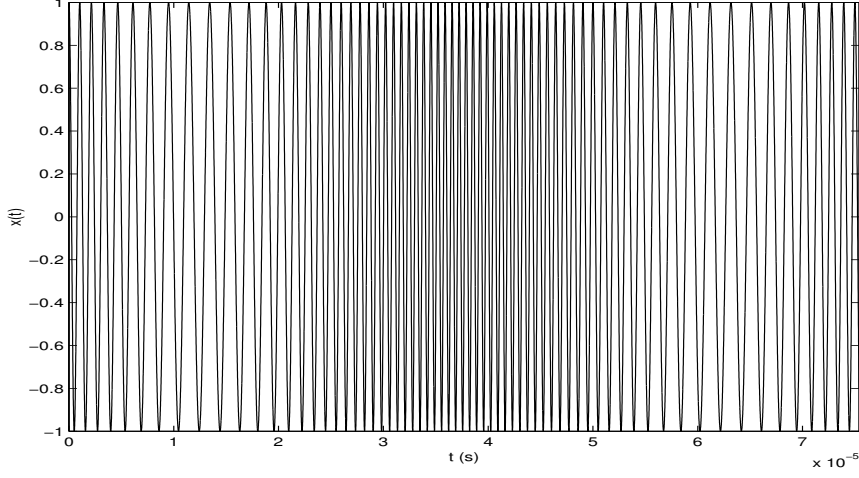


Figure 13: FM signal

Note that \hat{x}_1 is periodic in t_1 and t_2 , hence $x(t)$ is quasiperiodic with frequencies f_0 and f_2 . Unfortunately, $\hat{x}_1(t_1, t_2)$, illustrated in Figure 14, is not a simple surface with only a few undulations like Figure 10. When $k \gg 2\pi$, *i.e.*, $k \approx 2\pi m$ for some large integer m , then $\hat{x}_1(t_1, t_2)$ will undergo about m oscillations as a function of t_2 over one period T_2 . In practice, k is often of the order of $\frac{f_0}{f_2} \gg 2\pi$, hence this number of undulations can be very large. Therefore it becomes difficult to represent \hat{x}_1 efficiently by sampling on a two-dimensional grid. It turns out that resolving this problem requires the stretching, or warping, of one of the time scales. We illustrate this by returning to (30). Consider the following new multivariate representation

$$\hat{x}_2(\tau_1, \tau_2) = \cos(2\pi\tau_1), \quad (33)$$

together with the warping function

$$\phi(\tau_2) = f_0\tau_2 + \frac{k}{2\pi} \cos(2\pi f_2\tau_2). \quad (34)$$

We now retrieve our one-dimensional FM signal (*i.e.*, (30)) as

$$x(t) = \hat{x}_2(\phi(t), t). \quad (35)$$

Note that both \hat{x}_2 and ϕ , given in (33) and (34), can be easily represented with relatively few samples, unlike \hat{x}_1 in (32). What we have achieved with (34) is simply a stretching of the time axis differently at different times, to even out the period of the fast undulations in Figure 13. The extent of the stretching, or the derivative of $\phi(\tau_2)$, at a given point is simply the local frequency $\omega(\tau_2)$, which modifies the original MPDE to result in the **Warped Multirate Partial Differential Equation (WaMPDE)**:

$$\omega(\tau_2) \frac{\partial q(\hat{x})}{\partial \tau_1} + \frac{\partial q(\hat{x})}{\partial \tau_2} + f(\hat{x}(\tau_1, \tau_2)) = b(\tau_2). \quad (36)$$

The usefulness of (36) lies in that specifying

$$x(t) = \hat{x}(\phi(t), t), \quad \phi(t) = \int_0^t \omega(\tau_2) d\tau_2 \quad (37)$$

results in $x(t)$ being a solution to (1). Furthermore, when (36) is solved numerically, the local frequency $\omega(\tau_2)$ is also obtained, which is desirable for applications such as VCOs and also difficult to obtain by any other means.

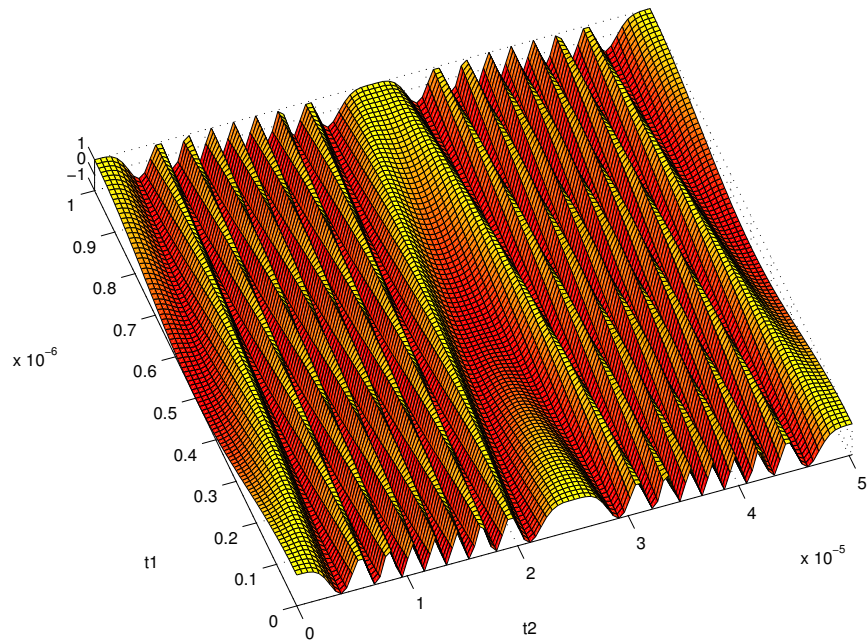


Figure 14: \hat{x}_1 : unwarped bivariate representation of FM signal

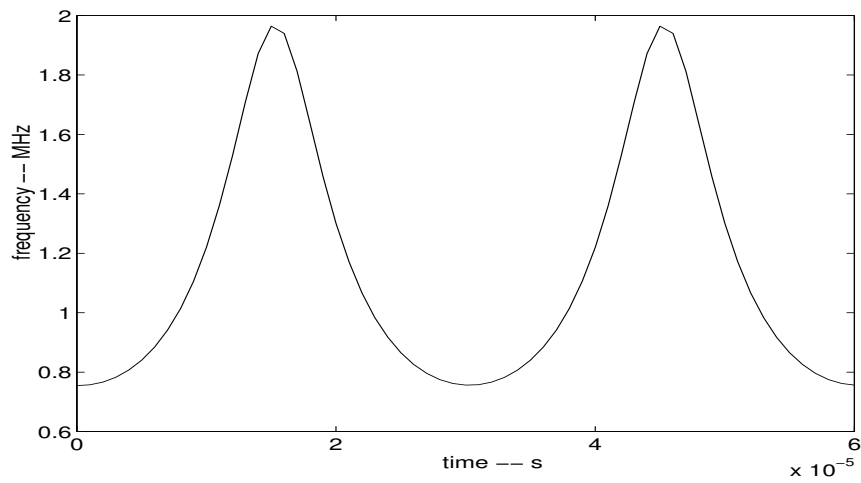


Figure 15: VCO: frequency modulation

As an example, Figure 15 shows the changing local frequency in an LC tank VCO simulated with WaMPDE-based numerical techniques. The controlling input to the VCO was about 30 times slower than its nominal frequency. Figure 16 depicts the bivariate waveform of the capacitor voltage. It is seen that the controlling voltage changes not only the local frequency, but also the amplitude and shape of the oscillator waveform.

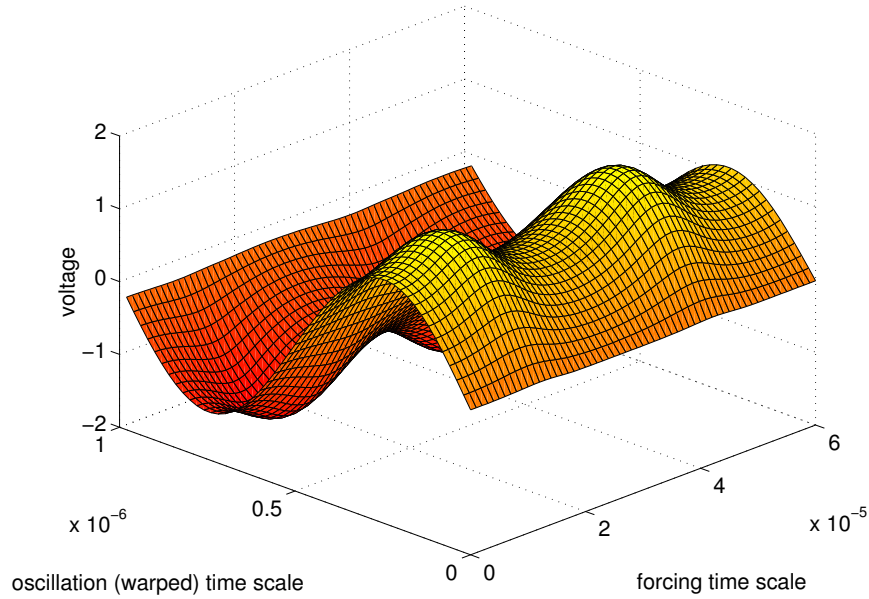


Figure 16: VCO: bivariate representation of capacitor voltage

The circuit was also simulated by traditional numerical ODE methods (“transient simulation”). The waveform from this simulation, together with the 1-dimensional waveform obtained by applying (37) to Figure 16, are shown in Figure 17. Frequency modulation can be observed in the varying density of the undulations.

6.2 Macromodelling time-varying systems

Another useful application of multiple time scales is in macromodelling linear time varying (LTV) systems [25, 26]. LTV approximations are adequate for many apparently nonlinear systems, like mixers and switched-capacitor filters, where the signal path is designed to be linear, even though other inputs (*e.g.*, local oscillators, clocks) cause “nonlinear” parametric changes to the system. LTV approximations of large systems with few inputs and outputs are particularly useful, because it is possible to automatically generate *macromodels* or *reduced-order models* of such systems. The macromodels are much smaller dynamical systems than the originals, but retain similar input-output behaviour to within a given accuracy. Such macromodels are useful in verifying systems hierarchically at different levels of abstraction, an important task in communication system design.

While mature techniques are available for the simpler task of reduced-order modelling of linear time-invariant (LTI) systems (*e.g.*, [27, 28, 29, 30, 31, 32, 33]), a difficulty in extending them to handle LTV systems has been the interference of the time-variations of the system and the input. By separating the two with artificial time variables, the MPDE provides an elegant solution to this problem.

The time-varying small-signal equations obtained by linearizing (1) around a steady-state solution are given by:

$$\begin{aligned} C(t)\dot{x}(t) + G(t)x(t) &= ru(t) \\ y(t) &= d^T x(t) \end{aligned} \tag{38}$$

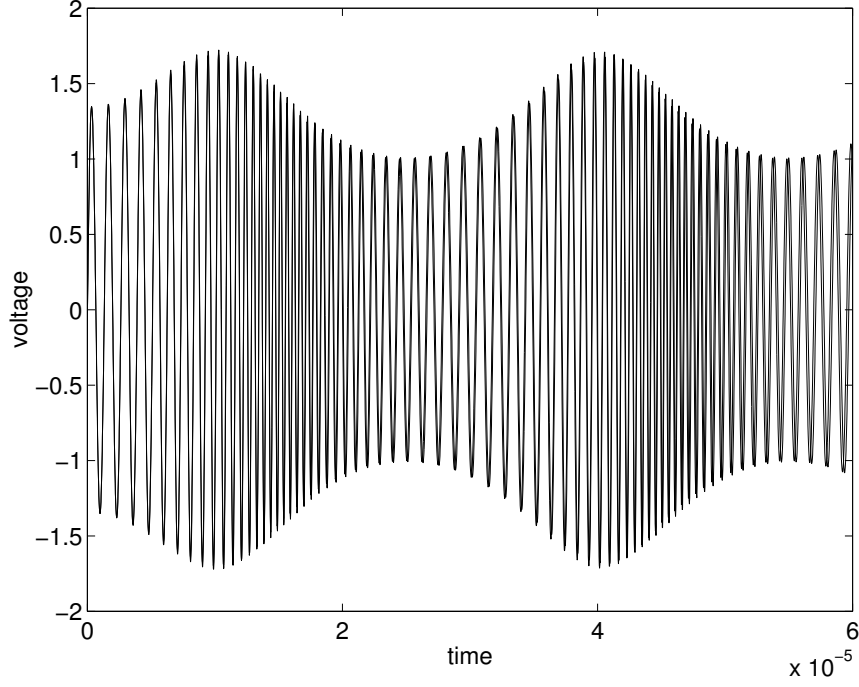


Figure 17: VCO: WaMPDE vs transient simulation

In (38), the input to the system is the scalar $u(t)$, while the output is $y(t)$. If the above equation is Laplace-transformed (following the LTI procedure), the system time variation in $C(t)$ and $G(t)$ interferes with the I/O time variation through a convolution. The LTV transfer function $H(t, s)$ is therefore hard to obtain; this is the difficulty alluded to earlier. The problem can be avoided by casting (38) as an MPDE:

$$\begin{aligned} C(t_1) \left[\frac{\partial \hat{x}}{\partial t_1}(t_1, t_2) + \frac{\partial \hat{x}}{\partial t_2}(t_1, t_2) \right] + G(t_1) \hat{x}(t_1, t_2) &= r u(t_2) \\ \hat{y}(t_1, t_2) &= d^T \hat{x}(t_1, t_2), \quad y(t) = \hat{y}(t, t) \end{aligned} \quad (39)$$

Notice that the input and system time variables are now separated. By taking Laplace transforms in t_2 and eliminating \hat{x} , the time-varying transfer function $H(t_1, s)$ is obtained:

$$Y(t_1, s) = \underbrace{\left\{ d^T \left[C(t_1) \left\{ \frac{\partial}{\partial t_1} + s \right\} + G(t_1) \right]^{-1} [r] \right\}}_{H(t_1, s)} U(s) \quad (40)$$

Observe that $H(t_1, s)$ in (40) is periodic in t_1 ; hence, discretizing the t_1 axis, it can also be represented as *several* time-invariant transfer functions $H_i(s) = H(t_{1_i}, s)$. Or, a frequency-domain discretization using harmonics of the t_1 -variation can be used. Once an equivalent system of LTI transfer functions has been obtained, existing reduced-order modelling techniques for LTI systems can be used to find a smaller system of equations, in the same form as (39), that have the same input-output relationship to within a given accuracy.

The reduced order modelling technique (dubbed *Time-Varying Padé*, or TVP) was run on a RFIC I-channel mixer circuit of size about $n = 360$ nodes, excited by a local oscillator at 178Mhz [26]. A frequency-domain discretization of the t_1 axis in (39) was employed in the model reduction process. Figure 6.2 shows frequency plots of $H_1(s)$, the upconversion transfer function (the first harmonic *wrt* t_1 of $H(t_1, s)$). The points marked '+' were obtained by direct computation of the full system, while the lines

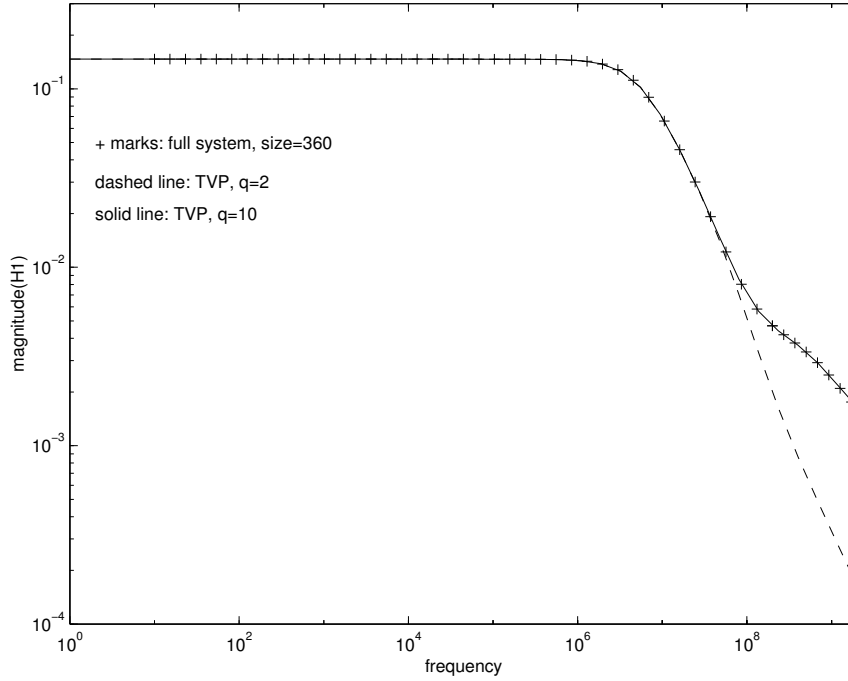


Figure 18: I-channel mixer $H_1(s)$: reduced vs full system

were computed using the reduced models of size $q = 2$ and $q = 10$, respectively¹. Even with $q = 2$, a size reduction of two orders of magnitude, the reduced model provides a good match up to the LO frequency. When the order of approximation is increased to 10, the reduced model is identical up to well beyond the LO frequency. The reduced models were more than three orders of magnitude faster to evaluate than the original system, hence they are useful for system-level verification.

The poles of the reduced models for $H_1(s)$, easily calculated on account of their small size, are shown in Table 1. These are useful in design because they constitute excellent approximations of the full system's poles, which are difficult to determine otherwise.

7 Noise in RF design

Predicting noise correctly in order to minimize its impact is central to RF design. Traditional circuit noise analysis is based on three assumptions: that noise sources and their effects are *small enough* not to change the operating point, that all noise sources are *stationary*, and that the small-signal linearization of the circuit is *time-invariant*. These assumptions break down when there are large signal variations, as is typical in RF circuits. Because of changing operating points, small-signal linearizations do not remain constant but become *time-varying*. In addition, noise sources that depend on operating point parameters (such as shot noise and flicker noise) also vary with time and no longer remain stationary. Finally, even though noise sources remain small, their impact upon circuit operation may or may not. In non-autonomous (driven) circuits, circuit effects of small noise remain small, allowing the use of linearized *mixing noise* analysis. In autonomous circuits (oscillators), however, noise creates frequency changes that lead to large deviations in waveforms over time – this phenomenon is called *phase noise*. Because of this, analysis based on linearization is not correct, and nonlinear analysis is required.

Figure 19 illustrates mixing noise. A periodic noiseless waveform in a circuit is shown as a function of time. The presence of

¹The order q of the reduced model is the number of state variables in its differential equation description.

TVP, $q = 2$	TVP, $q = 10$
-5.3951e+06	-5.3951e+06
-6.9196e+07 - j 3.0085e+05	-9.4175e+06
	-1.5588e+07 - j 2.5296e+07
	-1.5588e+07 + j 2.5296e+07
	-6.2659e+08 - j 1.6898e+06
	-1.0741e+09 - j 2.2011e+09
	-1.0856e+09 + j 2.3771e+09
	-7.5073e+07 - j 1.4271e+04
	-5.0365e+07 + j 1.8329e+02
	-5.2000e+07 + j 7.8679e+05

Table 1: Poles of $H_1(s)$ for the I-channel buffer/mixer

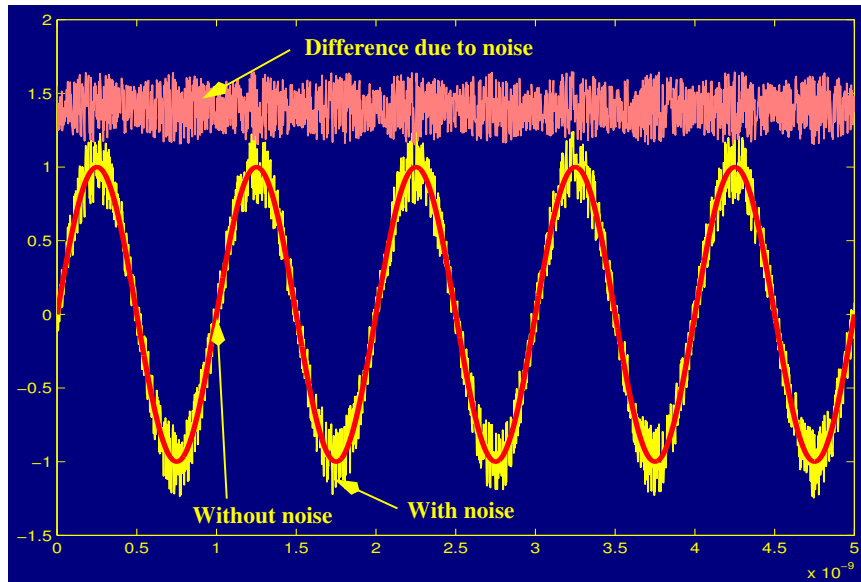


Figure 19: Time-domain view of mixing noise

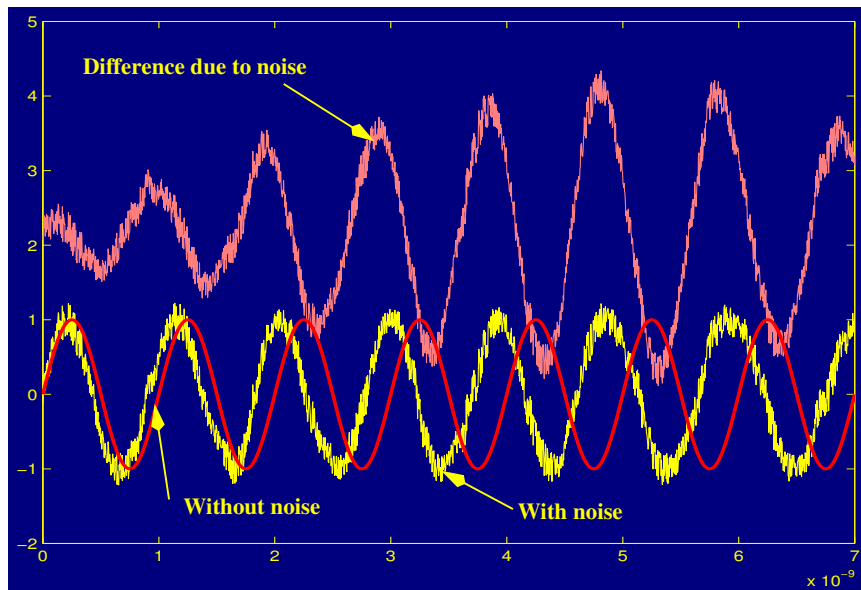


Figure 20: Time-domain view of phase noise

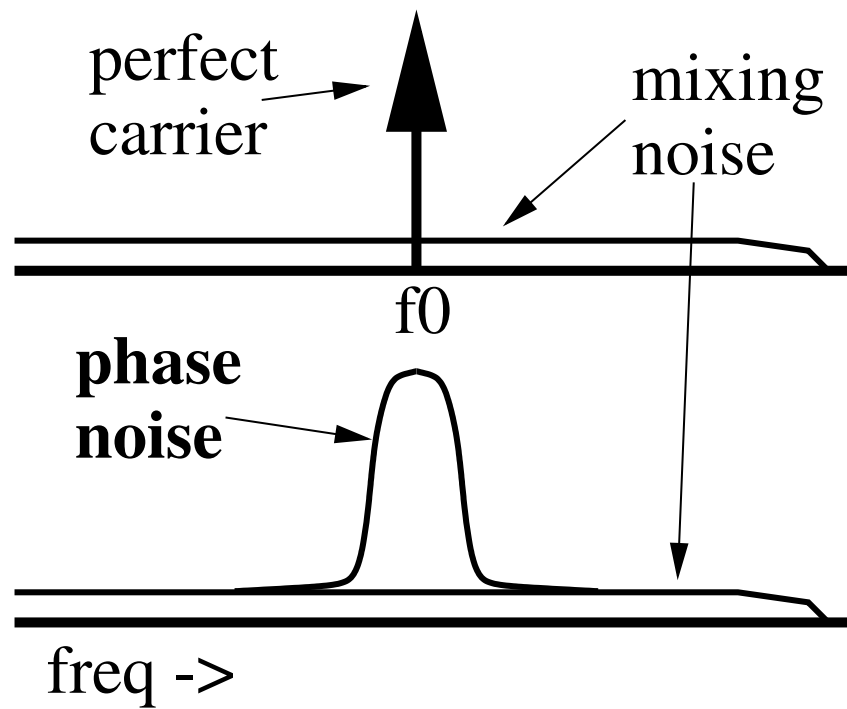


Figure 21: Frequency-domain view of mixing and phase noise

small noise corrupts the waveform, as indicated. The extent of corruption at any given time remains small, as shown by the third trace, which depicts the difference between the noiseless and noisy waveforms. The noise power can, however, vary depending on the large signal swing, as indicated by the roughly periodic appearance of the difference trace – depicting *cyclostationary* noise, where the statistics of the noise are periodic.

Figure 20 illustrates oscillator phase noise. Note that the noisy waveform’s frequency is now slightly different from the noise-free one’s, leading to increasing deviations between the two with the progress of time. As a result, the difference between the two does not remain small, but reaches magnitudes of the order of the large signal itself. Small additive corruptions remain here also just as in the mixing noise case, but the main distinguishing characteristic of oscillator noise are the frequency deviations.

The difference between mixing and phase noise is also apparent in the frequency domain, shown in Figure 21. Noise-free periodic waveforms appear as the impulse in the upper graph. If this is corrupted by small mixing noise, the impulse is not modified, but a small, possibly broadband, noise floor appears. In the case of free-running oscillators, the impulse disappears in the presence of any noise, no matter how small. It is replaced by a continuous spectrum that peaks at the oscillation frequency, and retains the power of the noise-free signal. The width and shape of this *phase noise spectrum* (i.e., the spread of power over neighbouring frequencies) is related to the amount and nature of noise in the circuit.

7.1 Mixing noise

Correct calculation of noise in nonlinear circuits with large signal swings (e.g., mixers, gain-compressed amplifiers) requires a sufficiently powerful stochastic process model. In the following, we use cyclostationary time-domain processes (e.g., [34, 35, 36, 37]), although a different but equivalent formulation, i.e., that of correlated processes in the frequency domain (e.g., [38, 5]), is often used. The statistics of cyclostationary processes (in particular the second-order statistics) are periodic or quasi-periodic, hence can be expressed in Fourier series. The coefficients of the Fourier series, termed *cyclostationary components*, capture the variations of noise power over time. The DC term of the Fourier series, or the *stationary* component, is typically the most relevant for design, since it captures the average noise power over a long time. It is important to realize, though, that calculating the correct value of the stationary component of noise over a circuit does require *all* the Fourier components to be properly accounted for. Basing calculations only on the stationary component at each node or branch current in the circuit will, in general, produce wrong results. This is analogous to computing the DC term of the product of two sinusoidal waveforms by simply multiplying the DC terms of each.

We motivate the need for cyclostationary analysis with an example. The circuit of Figure 22 consists of a mixer, followed by a bandpass filter, followed by another mixer. This is a simplification of, e.g., the bias-dependent noise generation mechanism in semiconductor devices [39]. Both mixers multiply their inputs by a local oscillator of frequency f_0 , i.e., by $\cos(2\pi f_0 t)$. The bandpass filter is centered around f_0 and has a bandwidth of $B \ll f_0$. The circuit is noiseless, but the input to the first mixer is stationary band-limited noise with two-sided bandwidth B .

A naïve attempt to determine the output noise power would consist of the following analysis, illustrated in Figure 22. The first mixer shifts the input noise spectrum by $\pm f_0$ and scales it by $1/4$. The resulting spectrum is multiplied by the squared magnitude of the filter’s transfer function. Since this spectrum falls within the pass-band of the filter, it is not modified. Finally, the second mixer shifts the spectrum again by $\pm f_0$ and scales it by $1/4$, resulting in the spectrum with three components shown in the figure. The total noise power at the output, i.e., the area under the spectrum, is $1/4$ that at the input.

This common but simplistic analysis is inconsistent with the following alternative argument. Note that the bandpass filter, which does not modify the spectrum of its input, can be ignored. The input then passes through only the two successive mixers,

resulting in the output noise voltage $o(t) = i(t) \cos^2(2\pi f_0 t)$. The output power is:

$$o^2(t) = i^2(t) \left[\frac{3}{8} + \frac{\cos(2\pi 2f_0 t) + \cos(2\pi 4f_0 t)}{2} \right]$$

The average output power consists of only the $3/8 i^2(t)$ term, since that the cosine terms time-average to zero. Hence the average output power is $3/8$ of the input power, 50% more than that predicted by the previous naïve analysis. This is, however, the correct result.

The contradiction between the arguments above underscores the need for cyclostationary analysis. The auto-correlation function of any cyclostationary process $z(t)$ (defined as $R_{zz}(t, \tau) = E[z(t)z(t + \tau)]$, $E[\cdot]$ denoting expectation) can be expanded in a Fourier series in t :

$$R_{zz}(t, \tau) = \sum_{i=-\infty}^{\infty} R_{z_i}(\tau) e^{ji2\pi f_0 t} \quad (41)$$

$R_{z_i}(\tau)$ are termed *harmonic autocorrelation functions*. The periodically time-varying power of $z(t)$ is its autocorrelation function evaluated at $\tau = 0$, i.e., $R_{zz}(t, 0)$. The quantities $R_{z_i}(0)$ represent the harmonic components of the periodically-varying power. The average power is simply the value of the DC or *stationary component*, $R_{z_0}(0)^2$. The frequency-domain representation of the harmonic autocorrelations are termed *harmonic power spectral densities* (HPSDs) $S_{z_i}(f)$ of $z(t)$, defined as the Fourier transforms:

$$S_{z_i}(f) = \int_{-\infty}^{\infty} R_{z_i}(\tau) e^{-j\pi f \tau} d\tau \quad (42)$$

Equations can be derived that relate the HPSDs at the inputs and outputs of various circuit blocks. By solving these equations, any HPSDs in the circuit can be determined.

Consider, for example, the circuit in Figure 22. The input and output HPSDs of a perfect cosine mixer with unit amplitude can be shown [40] to be related by:

$$S_{v_k}(f) = \frac{S_{u_{k-2}}(f - f_0)}{4} + \frac{S_{u_k}(f - f_0) + S_{u_k}(f + f_0)}{4} + \frac{S_{u_{k+2}}(f + f_0)}{4} \quad (43)$$

(u and v denoting the input and output, respectively). The HPSD relation for a filter with transfer function $H(f)$ is [40]:

$$S_{v_k}(f) = H(-f)H(f + kf_0)S_{u_k}(f) \quad (44)$$

The HPSDs of the circuit are illustrated in Figure 23. Since the input noise $i(t)$ is stationary, its only nonzero HPSD is the stationary component $S_{i_0}(f)$, assumed to be unity in the frequency band $[-B/2, B/2]$, as shown. From Equation 43 applied to the first mixer, *three* nonzero HPSDs (S_{x_0} , S_{x_2} and $S_{x_{-2}}$, shown in the figure) are obtained for $x(t)$. These are generated by shifting the input PSD by $\pm f_0$ and scaling by $1/4$; in contrast to the naïve analysis, the stationary HPSD is not the only spectrum used to describe the upconverted noise. From Equation 44, it is seen that the ideal bandpass filter propagates the three HPSDs of $x(t)$ unchanged to $y(t)$. Through Equation 43, the second mixer generates five nonzero HPSDs, of which only the stationary component $S_{o_0}(f)$ is shown in the figure. This is obtained by scaling and shifting not only the stationary HPSD of $y(t)$, but also the cyclostationary HPSDs, which in fact contribute an extra $1/4$ to the lobe centered at zero. The average output noise (the shaded area under $S_{o_0}(f)$) equals $3/8$ of the input noise.

We now sketch the general procedure for analyzing mixing noise in circuits. The noise sources within a circuit can be represented by a small additive term $Au(t)$ to (1), where $u(t)$ is a vector of noise sources, and A an incidence matrix capturing

²Stationary processes are a special case of cyclostationary processes, where the autocorrelation function (hence the power) is independent of the time t ; it follows that $R_{z_i}(\tau) \equiv 0$ if $i \neq 0$.

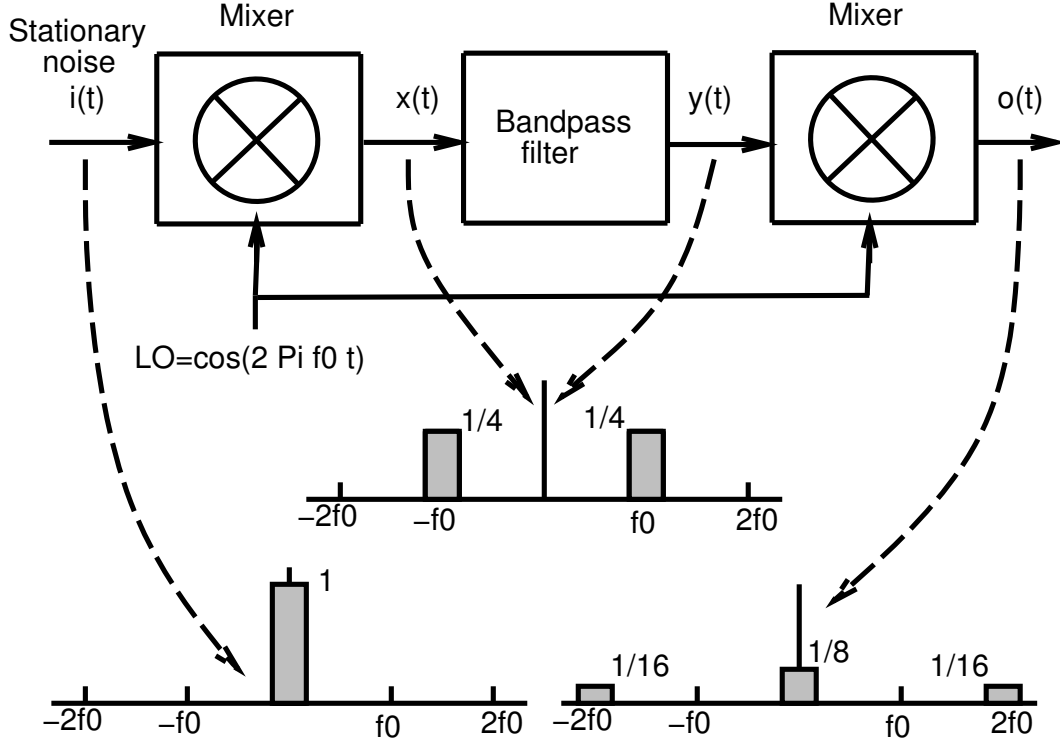


Figure 22: Mixer-filter-mixer circuit: naïve analysis

their connections to the circuit. (1) is first solved for a (quasi-)periodic steady state in the absence of noise, and then linearized as in (38), to obtain:

$$C(t)\dot{x} + G(t)x + Au(t) = 0 \quad (45)$$

where $x(t)$ represents the small-signal deviations due to noise. Equation 45 describes a linear periodically time-varying (LPTV) system with input $u(t)$ and output $x(t)$. The system can be characterized by its time-varying transfer function $H(t, f)$. $H(t, f)$ is periodic in t and can be expanded in a Fourier series similar to Equation 41. Denote the Fourier components (*harmonic transfer functions*) by $H_i(f)$, .

Since $u(t)$ and $x(t)$ are vectors, their autocorrelation functions are *matrices* $R_{zz}(t, \tau) = E[z(t)z^T(t + \tau)]$, consisting of auto- and cross-correlations. Similarly, the HPSDs $S_{z_i}(f)$ are also matrices. It can be shown [37] that the HPSD matrices of y and u are related by:

$$S_{xx}(f) = \mathcal{H}(f) S_{uu}(f) \mathcal{H}^*(f) \quad (46)$$

$\mathcal{H}(f)$ (the *conversion matrix*) is the following block-structured matrix (f^k denotes $f + kf_0$):

$$\mathcal{H}(f) = \begin{pmatrix} \vdots & \vdots & \vdots \\ \cdots & H_0(f^1) & H_1(f^0) & H_2(f^{-1}) & \cdots \\ \cdots & H_{-1}(f^1) & H_0(f^0) & H_1(f^{-1}) & \cdots \\ \cdots & H_{-2}(f^1) & H_{-1}(f^0) & H_0(f^{-1}) & \cdots \\ \vdots & \vdots & \vdots \end{pmatrix} \quad (47)$$

$S_{uu}(f)$ and $S_{xx}(f)$ are similar to $\mathcal{H}(f)$: their transposes $S_{zz}^T(f)$ have the same structure, but with $H_i(f^k)$ replaced by $S_{z_i}^T(f^k)$.

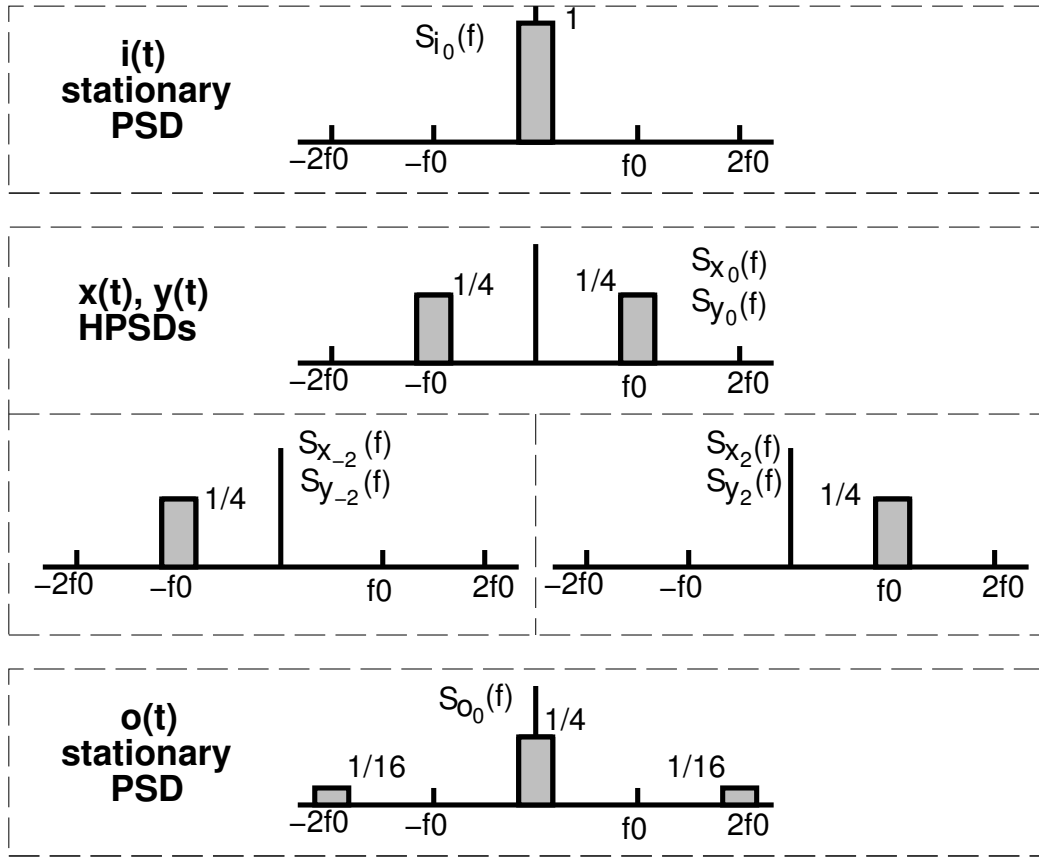


Figure 23: HPSDs of mixer-filter-mixer circuit

Equation 46 expresses the output HPSDs, contained in $\mathcal{S}_{xx}(f)$, in terms of the input HPSDs (contained in $\mathcal{S}_{uu}(f)$) and the harmonic transfer functions of the circuit (contained in $\mathcal{H}(f)$). The HPSDs of a single output variable $x_p(t)$ (both auto- and cross-terms with all other output variables) are available in the p^{th} column of the central block-column of $\mathcal{S}_{xx}^T(f)$. To pick this column, $\mathcal{S}_{xx}^T(f)$ is applied to a unit vector E_{0p} , as follows ($\overline{}$ denotes the conjugate):

$$\mathcal{S}_{xx}^T(f) E_{0p} = \overline{\mathcal{H}(f)} \mathcal{S}_{uu}^T(f) \mathcal{H}^T(f) E_{0p} \quad (48)$$

Evaluating Equation 48 involves two kinds of matrix-vector products: $\mathcal{H}(f)z$ and $\mathcal{S}_{uu}(f)z$ for some vectors z . Consider the latter product first. If the inputs $u(t)$ are stationary, as can be assumed without loss of generality [40], then $\mathcal{S}_{uu}(f)$ is block-diagonal. In practical circuits, the inputs $u(t)$ are either uncorrelated or sparsely correlated. This results in each diagonal block of $\mathcal{S}_{uu}(f)$ being either diagonal or sparse. In both cases, the matrix-vector product can be performed efficiently.

The product with $\mathcal{H}(f)$ can also be performed efficiently, by exploiting the relation $\mathcal{H}(f) = J^{-1}(f) \mathcal{A}$ [4]. \mathcal{A} is a sparse incidence matrix of the device noise generators, hence its product with a vector can be computed efficiently. $J(0)$ is the harmonic balance Jacobian matrix [9] at the large-signal solution $x^*(t)$. $J(f)$ is obtained by replacing kf_0 by $kf_0 + f$ in the expression for the Jacobian.

The product $J^{-1}z$ can therefore be computed efficiently using the fast techniques outlined in Section 5.2. As a result, Equation 48 can be computed efficiently for large circuits to provide the auto- and cross-HPSDs of any output of interest.

For example, a portion of the Lucent W2013 RFIC, consisting of an I-channel buffer feeding a mixer, was simulated using (48). The circuit consisted of about 360 nodes, and was excited by two tones — a local oscillator at 178Mhz driving the mixer, and a strong RF signal tone at 80Khz feeding into the I-channel buffer. Two noise analyses were performed. The first analysis included both LO and RF tones (sometimes called a three-tone noise analysis). The circuit was also analysed with only the LO tone to determine if the RF signal affects the noise significantly. The two-tone noise simulation, using a total of 525 large-signal mix components, required 300MB of memory and for each frequency point, took 40 minutes on an SGI machine (200MHz R10000 CPU). The one-tone noise simulation, using 45 harmonics, needed 70MB of memory and took 2 minutes per point.

The stationary PSDs of the mixer output noise for the two simulations are shown in Fig. 24. It can be seen that the presence of the large RF signal increases the noise by about 1/3. This is due to *noise folding*, the result of devices being driven into nonlinear regions by the strong RF input tone. The peaks in the two waveforms, located at the LO frequency, are due to up- and down-conversion of noise from other frequencies.

7.2 Phase Noise

Even small noise in an oscillator leads to dramatic changes in its frequency spectrum and timing properties, *i.e.*, to phase noise.

This effect can lead to interchannel interference and increased bit-error-rates (BER) in RF communication systems. Another manifestation of the same phenomenon, jitter, is important in clocked and sampled-data systems: uncertainties in switching instants caused by noise can affect synchronisation.

Although a large body of literature is available on phase noise³, treatments of the phenomenon from the design perspective have typically been phenomenological, *e.g.*, the well-known treatment of Leeson [42]. Most analyses have been based on linear time-invariant or time-varying approaches, which, though providing useful design guidelines, contain qualitative inaccuracies — *e.g.*, they can predict infinite noise power. Recently, however, the work of Kärtner [43] and Demir et al [41] have provided a more correct understanding of phase noise. Here, we sketch the approach in [41].

³[41] contains a list of references.

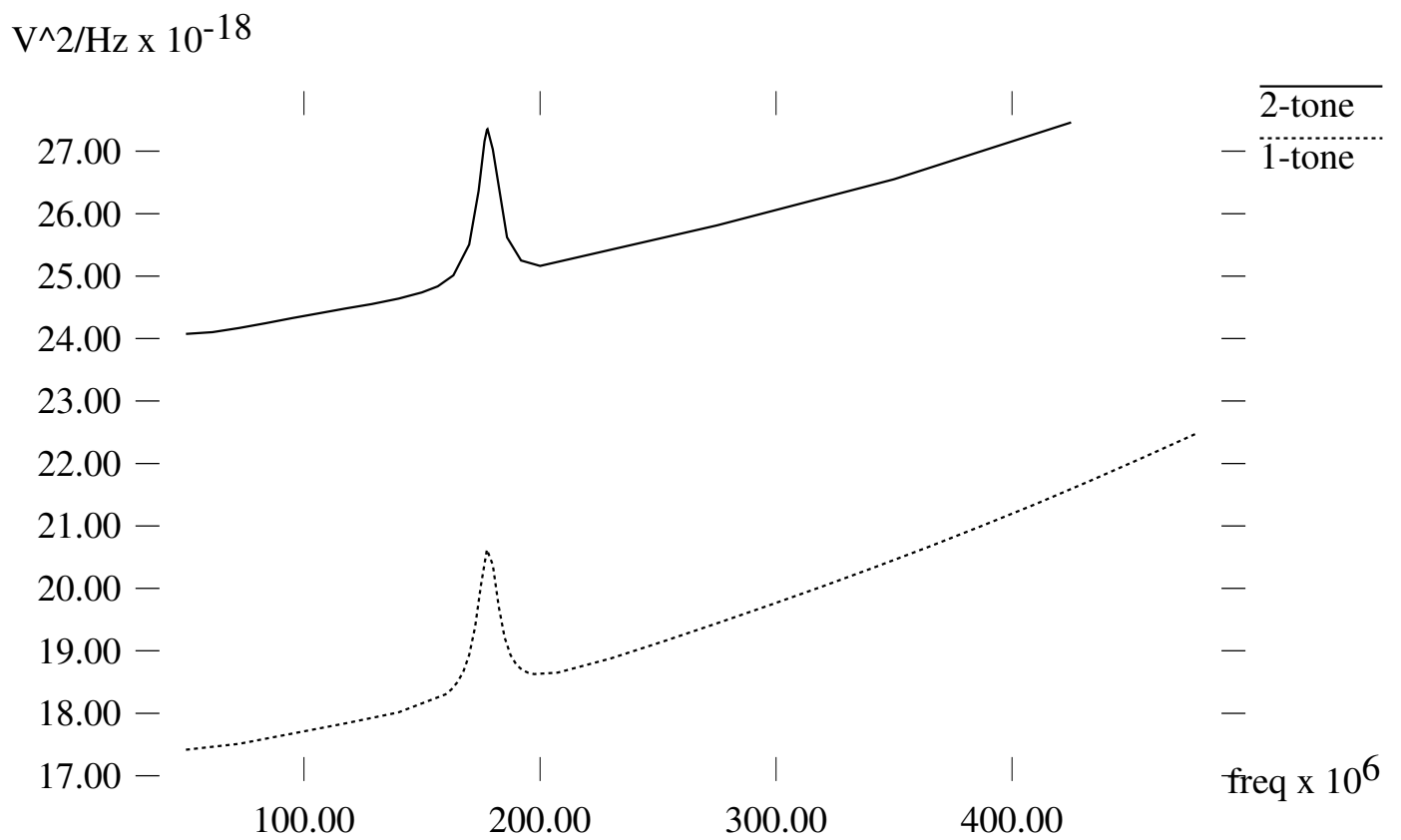


Figure 24: Stationary PSDs for the I-Q mixer/buffer circuit

The starting point for phase noise analysis is (1), reproduced here for oscillators with no external forcing:

$$\dot{q}(x) + f(x) = 0 \quad (49)$$

We assume (49) to be the equations for an oscillator with an orbitally stable⁴, nontrivial periodic solution, *i.e.*, an oscillation waveform $x_s(t)$. With small noise generators in the circuit, possibly dependent on circuit state, the equation becomes

$$\dot{q}(x) + f(x) = B(x)b(t), \quad (50)$$

where $b(t)$ now represents small perturbations.

When $b(t)$ is small, it can be shown [45] that the originally periodic oscillation $x_s(t)$ changes to

$$x(t) = x_s(t + \alpha(t)) + y(t), \quad (51)$$

where $y(t)$ remains small, but $\alpha(t)$ (a time/phase deviation) can grow unboundedly with time, no matter how small the perturbation $b(t)$ is (see Figure 25). For driven circuits (the mixing noise case) $\alpha(t)$ remains bounded and small and its effects can therefore be lumped into the $y(t)$ term. This is the difference illustrated in Figure 19, Figure 20 and Figure 21. The underlying reason for this difference is that oscillators by their very definition are phase-unstable, hence phase errors build up indefinitely.

Furthermore, it can be shown that $\alpha(t)$ is given by a nonlinear scalar differential equation

$$\dot{\alpha} = v_1^T(t + \alpha(t))B(x_s(t + \alpha(t)))b(t), \quad (52)$$

where $v_1(t)$ is a periodic vector function dubbed the Perturbation Projection Vector (PPV). The PPV, which is characteristic of an oscillator in steady state and does not depend on noise parameters, is an important quantity for phase noise calculation. Roughly speaking, it is a “transfer function” that relates perturbations to resulting time or phase jitter in the oscillator. The PPV can be found through only a linear time-varying analysis of the oscillator around its oscillatory solution, and simple techniques to calculate it using HB or shooting are available [46].

In general, (52) can be difficult to solve analytically. When the perturbation $b(t)$ is white noise, however, it can be shown that $\alpha(t)$ becomes a Gaussian random walk process with linearly increasing variance ct , where c is a scalar constant given by

$$c = \frac{1}{T} \int_0^T v_1^T(t)B(x_s(t))B^T(x_s(t))v_1(t)dt, \quad (53)$$

with T the period of the unperturbed oscillation.

This random walk stochastic characterization of the phase error $\alpha(t)$ implies that:

1. the average spread of the jitter (mean-square jitter) increases *linearly* with time, with cT being the jitter per cycle.
2. the spectrum of the oscillator’s output, *i.e.*, the power spectrum of $x_s(t + \alpha(t))$, is *Lorenzian*⁵ about each harmonic. For example, around the fundamental (with angular frequency $\omega_0 = \frac{2\pi}{T}$ and power P_{fund}), the spectrum is

$$S_p(f) = P_{\text{fund}} \frac{\omega_0^2 c}{\frac{w_0^4 c^2}{4} + (2\pi f - \omega_0)^2}. \quad (54)$$

This means that the spectrum decays as $\frac{1}{f^2}$ beyond a certain knee distance away from the original oscillation frequency and its harmonics, as is well-known for white noise in oscillators [42]. The $\frac{1}{f^2}$ dependence does not, however, continue as $f \rightarrow 0$, *i.e.*, close to and at the oscillation frequency; instead, the spectrum reaches a finite maximum value.

⁴see, *e.g.*, [44] for a precise definition; roughly speaking, an orbitally stable oscillator is one that eventually reaches a unique, periodic waveform with a definite magnitude.

⁵A Lorenzian is the shape of the squared magnitude of a one-pole lowpass filter transfer function.

3. the oscillator's output is a *stationary* stochastic process.

The Lorentzian shape of the spectrum also implies that the power spectral density at the carrier frequency and its harmonics has a finite value, and that the total carrier power is preserved despite spectral spreading due to noise.

(52) can also be solved for coloured noise perturbations $b(t)$ [45], and it can be shown that if $S(f)$ is the spectrum of the coloured noise, then the phase noise spectrum generated falls as $\frac{S(f)}{(f-f_0)^2}$, away from f_0 .

Numerical methods based on the above insights are available to calculate phase noise.

The main effort is calculating the PPV; once it is known, c can be calculated easily using (53) and the spectrum obtained directly from (54). The PPV can be found from the time-varying linearization of the oscillator around its steady state. Two numerical methods can be used to find the PPV. The first calculates the time-domain monodromy (or state-transition) matrix of the linearized oscillator explicitly, and obtains the PPV by eigendecomposing this matrix [41]. A more recent method [46] relies on simple postprocessing of internal matrices generated during the solution of the operator's steady state using HB or shooting, and as such, can take advantage of the fast techniques of Section 5.2. The separate contributions of noise sources, and the sensitivity of phase noise to individual circuit devices and nodes, can be obtained easily.

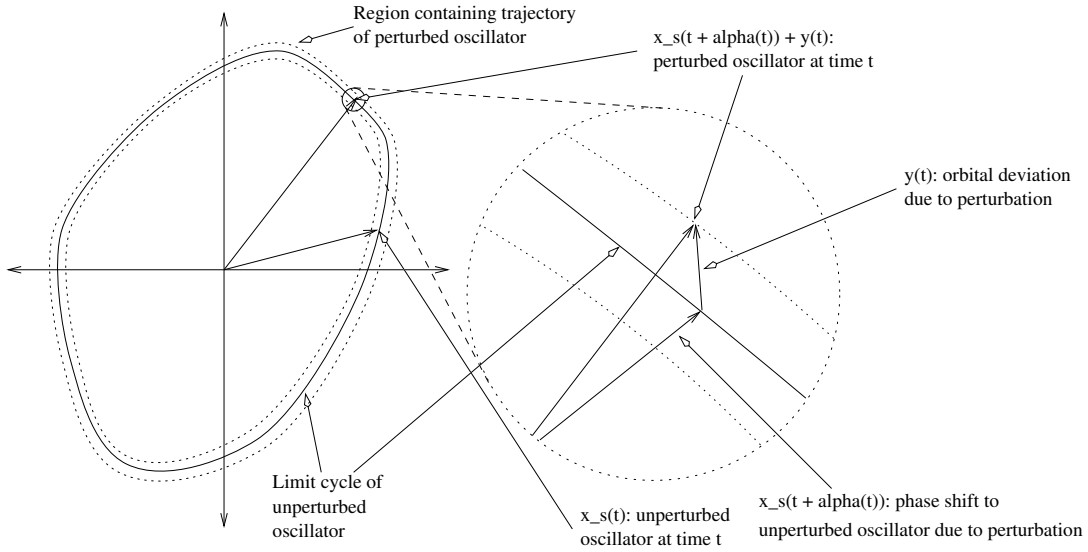


Figure 25: Oscillator trajectories

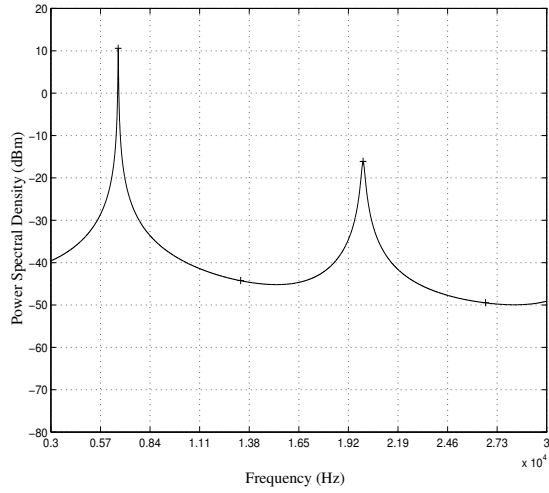
As an example, the oscillator in Figure 26 consists of a Tow-Thomas second-order bandpass filter and a comparator [47]. If the OpAmps are considered to be ideal, it can be shown that this oscillator is equivalent (in the sense of the differential equations that describe it) to a parallel RLC circuit in parallel with a nonlinear voltage-controlled current source (or equivalently a series RLC circuit in series with a nonlinear current-controlled voltage source). In [47], authors breadboarded this circuit with an external white noise source (intensity of which was chosen such that its effect is much larger than the other internal noise sources), and measured the PSD of the output with a spectrum analyzer. For $Q = 1$ and $f_o = 6.66$ kHz,

a phase noise characterisation of this oscillator was performed

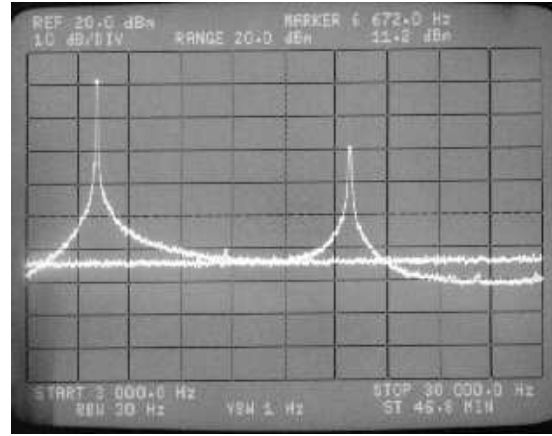
to yield the periodic oscillation waveform $x_s(t)$ for the output and $c = 7.56 \times 10^{-8} \text{ sec}^2.\text{Hz}$. Figure 27(a) shows the PSD of the oscillator output

and Figure 27(b) shows the spectrum analyzer measurement⁶. Figure 27(c) shows a blown up version of the PSD around the first harmonic. The single-sideband phase noise spectrum is in Figure 27(d). The oscillator model that was simulated has two

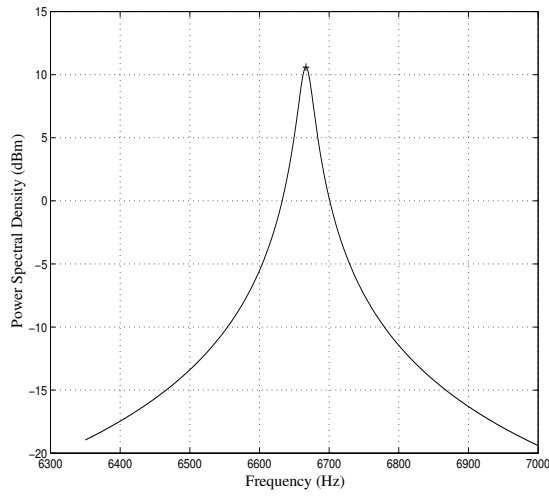
⁶The PSDs are plotted in units of dBm.



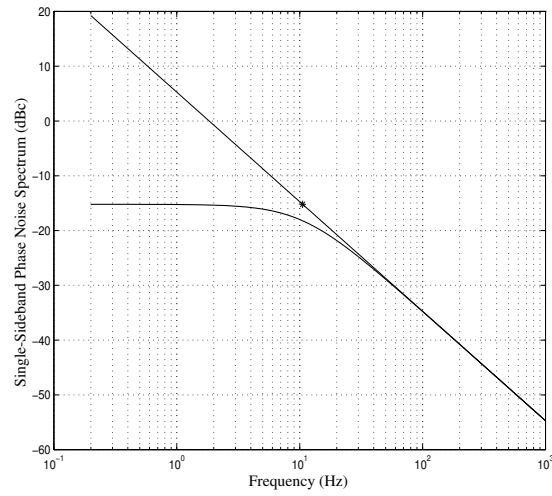
(a) Computed PSD (4 harmonics)



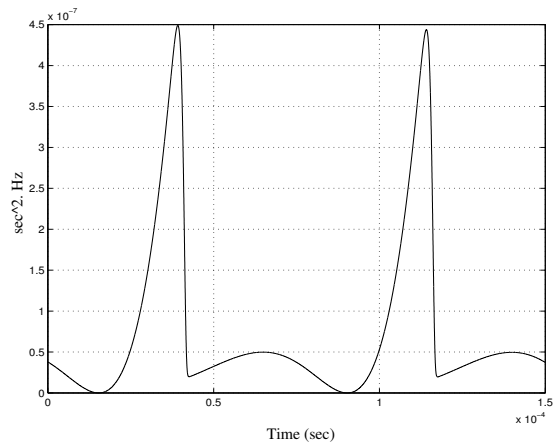
(b) Spectrum analyzer measured PSD [47]



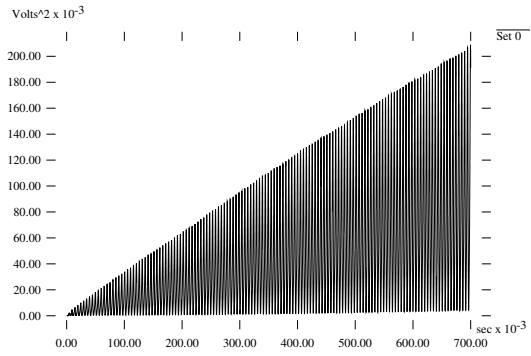
(c) Computed PSD (first harmonic)



(d) $\mathcal{L}(f_m)$



(e) $(v_1^T(t)B)^2$



(f) Variance of total deviation (Monte-Carlo method)

Figure 27: Phase noise characterisation for the oscillator in Figure 26

and Section 7.

References

- [1] L.O. Chua and P-M. Lin. *Computer-aided analysis of electronic circuits : algorithms and computational techniques*. Prentice-Hall, Englewood Cliffs, N.J., 1975.
- [2] K.S. Kundert, J.K. White, and A. Sangiovanni-Vincentelli. *Steady-state methods for simulating analog and microwave circuits*. Kluwer Academic Publishers, 1990.
- [3] M.S. Nakhla and J. Vlach. A piecewise harmonic balance technique for determination of periodic responses of nonlinear systems. *IEEE Trans. Ckts. Syst.*, CAS-23:85, 1976.
- [4] S.A. Haas. *Nonlinear Microwave Circuits*. Artech House, Norwood, MA, 1988.
- [5] V. Rizzoli and A. Neri. State of the art and present trends in nonlinear microwave CAD techniques. *IEEE Trans. MTT*, 36(2):343–365, February 1988.
- [6] R.J. Gilmore and M.B. Steer. Nonlinear circuit analysis using the method of harmonic balance – a review of the art. Part I. Introductory concepts. *Int. J. on Microwave and Millimeter Wave CAE*, 1(1), 1991.
- [7] Markus Rösch. *Schnell Simulation des stationären Verhaltens nichtlinearer Schaltungen*. PhD thesis, Technischen Universität München, 1992.
- [8] R. Mickens. *Oscillations in Planar Dynamic Systems*. World Scientific, 1995.
- [9] R.C. Melville, P. Feldmann, and J. Roychowdhury. Efficient multi-tone distortion analysis of analog integrated circuits. In *Proc. IEEE CICC*, pages 241–244, May 1995.
- [10] D. Long, R.C. Melville, et al. Full chip harmonic balance. In *Proc. IEEE CICC*, May 1997.
- [11] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes – The Art of Scientific Computing*. Cambridge University Press, 1989.
- [12] T.J. Aprille and T.N. Trick. Steady-state analysis of nonlinear circuits with periodic inputs. *Proc. IEEE*, 60(1):108–114, January 1972.
- [13] S. Skelboe. Computation of the periodic steady-state response of nonlinear networks by extrapolation methods. *IEEE Trans. Ckts. Syst.*, CAS-27(3):161–175, March 1980.
- [14] A. Nayfeh and B. Balachandran. *Applied Nonlinear Dynamics*. Wiley, 1995.
- [15] R. Telichevesky, K. Kundert, and J. White. Efficient steady-state analysis based on matrix-free krylov subspace methods. In *Proc. IEEE DAC*, pages 480–484, 1995.
- [16] P. Feldmann, R.C. Melville, and D. Long. Efficient frequency domain analysis of large nonlinear analog circuits. In *Proc. IEEE CICC*, May 1996.
- [17] Y. Saad. *Iterative methods for sparse linear systems*. PWS, Boston, 1996.

- [18] R.W. Freund. Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation. Technical Report 11273-980217-02TM, Bell Laboratories, 1998.
- [19] R.W. Freund. Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation. *Applied and Computational Control, Signals, and Circuits, Volume 1*, pages 435–498, 1999.
- [20] J. Kevorkian and J.D. Cole. *Perturbation methods in Applied Mathematics*. Springer-Verlag, 1981.
- [21] E. Ngoya and R. Larchevêque. Envelop transient analysis: a new method for the transient and steady state analysis of microwave communication circuits and systems. In *Proc. IEEE MTT Symp.*, 1996.
- [22] H.G. Brachtendorf, G. Welsch, R. Laur, and A. Bunse-Gerstner. Numerical steady state analysis of electronic circuits driven by multi-tone signals. *Electrical Engineering (Springer-Verlag)*, 79:103–112, 1996.
- [23] J. Roychowdhury. Efficient methods for simulating highly nonlinear multi-rate circuits. In *Proc. IEEE DAC*, 1997.
- [24] O. Narayan and J. Roychowdhury. Multi-time simulation of voltage-controlled oscillators. In *Proc. IEEE DAC*, New Orleans, LA, June 1999.
- [25] J. Phillips. Model Reduction of Time-Varying Linear Systems Using Approximate Multipoint Krylov-Subspace Projectors. In *Proc. ICCAD*, November 1998.
- [26] J. Roychowdhury. Reduced-order modelling of time-varying systems. *IEEE Trans. Ckts. Syst. – II: Sig. Proc.*, 46(10), November 1999.
- [27] L.T. Pillage and R.A. Rohrer. Asymptotic waveform evaluation for timing analysis. *IEEE Trans. CAD*, 9:352–366, April 1990.
- [28] X. Huang, V. Raghavan, and R.A. Rohrer. AWESim: A program for the efficient analysis of linear(ized) circuits. In *Proc. ICCAD*, pages 534–537, November 1990.
- [29] E. Chiprout and M.S. Nakhla. *Asymptotic Waveform Evaluation*. Kluwer, Norwell, MA, 1994.
- [30] P. Feldmann and R.W. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. *IEEE Trans. CAD*, 14(5):639–649, May 1995.
- [31] P. Feldmann and R.W. Freund. Reduced-order modeling of large linear subcircuits via a block Lanczos algorithm. In *Proc. IEEE DAC*, pages 474–479, 1995.
- [32] P. Feldmann and R.W. Freund. Circuit noise evaluation by Padé approximation based model-reduction techniques. In *Proc. ICCAD*, pages 132–138, November 1997.
- [33] A. Odabasioglu, M. Celik, and L.T. Pileggi. PRIMA: passive reduced-order interconnect macromodelling algorithm. In *Proc. ICCAD*, pages 58–65, November 1997.
- [34] W. Gardner. *Introduction to Random Processes*. McGraw-Hill, New York, 1986.
- [35] T. Ström and S. Signell. Analysis of Periodically Switched Linear Circuits. *IEEE Trans. Ckts. Syst.*, CAS-24(10):531–541, Oct 1977.

- [36] M. Okumura, H. Tanimoto, T. Itakura, and T. Sugawara. Numerical Noise Analysis for Nonlinear Circuits with a Periodic Large Signal Excitation Including Cyclostationary Noise Sources. *IEEE Trans. Ckts. Syst. – I: Fund. Th. Appl.*, 40(9):581–590, Sep 1993.
- [37] J. Roychowdhury, D. Long, and P. Feldmann. Cyclostationary noise analysis of large RF circuits with multitone excitations. *IEEE J. Solid-State Ckts.*, 33:324–336, Mar 1998.
- [38] V. Rizzoli, F. Mastri, and D. Masotti. General Noise Analysis of Nonlinear Microwave Circuits by the Piecewise Harmonic-Balance Technique. *IEEE Trans. MTT*, 42(5):807–819, May 1994.
- [39] A.R. Kerr. Noise and loss in balanced and subharmonically pumped mixers: Part 1 – Theory. *IEEE Trans. MTT*, MTT-27:938–943, Dec 1979.
- [40] J. Roychowdhury and P. Feldmann. A new linear-time harmonic balance algorithm for cyclostationary noise analysis in RF circuits. In *Proc ASP-DAC*, pages 483–492, 1997.
- [41] A. Demir, A. Mehrotra, and J. Roychowdhury. Phase noise in oscillators: a unifying theory and numerical methods for characterization. *IEEE Trans. Ckts. Syst. – I: Fund. Th. Appl.*, 47:655–674, May 2000.
- [42] D.B. Leeson. A simple model of feedback oscillator noise spectrum. *Proceedings of the IEEE*, 54(2):329, February 1966.
- [43] F. Kärtner. Analysis of white and $f^{-\alpha}$ noise in oscillators. *International Journal of Circuit Theory and Applications*, 18:485–519, 1990.
- [44] M. Farkas. *Periodic Motions*. Springer-Verlag, 1994.
- [45] A. Demir. Phase Noise in Oscillators: DAEs and Colored Noise Sources. In *Proc. ICCAD*, pages 170–177, 1998.
- [46] A. Demir, D. Long and J. Roychowdhury. Computing Phase Noise Eigenfunctions Directly from Steady-State Jacobian Matrices. In *Proc. ICCAD*, November 2000.
- [47] A. Dec, L. Toth, and K. Suyama. Noise analysis of a class of oscillators. *IEEE Transactions on Circuits and Systems*, 1998.
- [48] Kenneth S. Kundert and Alberto Sangiovanni-Vincentelli. Simulation of nonlinear circuits in the frequency domain. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, CAD-5(4):521–535, October 1986.
- [49] Kenneth S. Kundert, Gregory B. Sorkin, and Alberto Sangiovanni-Vincentelli. Applying harmonic balance to almost-periodic circuits. *IEEE Transactions on Microwave Theory and Techniques*, MTT-36(2):366–378, February 1988.
- [50] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes – The Art of Scientific Computing*. Cambridge University Press, 1989.
- [51] P. Antognetti and G. Massobrio, *Semiconductor Device Modeling with SPICE*, Chapter 1, McGraw-Hill, 1988.
- [52] W. Liu, *MOSFET Models for SPICE Simulation Including BSIM3v3 and BSIM4*, John Wiley & Sons, New York, NY, 2001.
- [53] Y. P. Tsividis, *Operation and Modeling of the MOS Transistor*, McGraw-Hill, New York, NY, 1987.
- [54] I. E. Getreu, *Modeling the Bipolar Transistor*, Tektronix, Inc., 1976.
- [55] D. Foty, *MOSFET Modeling with SPICE: Principles and Practice*, Prentice Hall, Upper Saddle River, NJ, 1997.

- [56] K. Kundert and O. Zinke, *The Designer's Guide to Verilog-AMS*, Kluwer Academic Publishers, Norwell, MA, 2004.
- [57] D. Fitzpatrick and I. Miller, *Analog Behavioral Modeling with the Verilog-A Language*, Kluwer Academic Publishers, Norwell, MA, 1998.
- [58] H. A. Mantooth and M. Fiegebaum, *Modeling with an Analog Hardware Description Language*, Kluwer Academic Publishers, Norwell, MA, 1995.
- [59] P. Ashenden, G. Peterson and D. Teegarden, *System Designer's Guide to VHDL-AMS*, Morgan-Kaufman, 2003.
- [60] P. Su, S. K. H. Fung, S. Tang, F. Assaderaghi and C. Hu, "BSIMPD: A partial-depletion SOI MOSFET model for deep-submicron CMOS designs," *IEEE Proc. of Custom Integrated Circuits Conference*, pp. 197-200, 2000.
- [61] B. G. Streetman, *Solid State Electronic Devices*, Prentice-Hall, pp. 172-173, 1980.
- [62] A. S. Sedra and K. C. Smith, *Microelectronic Circuits*, 5th Edition, Chapter 3, Oxford University Publishing, New York, NY, 2004.
- [63] H. A. Mantooth and M. Vlach, "Beyond SPICE with Saber and MAST," *IEEE Proc. of Int. Symposium on Circuits Syst.*, vol. 1, pp. 77-80, May 1992.
- [64] H. A. Mantooth and J. L. Duliere, "A unified diode model for circuit simulation," *IEEE Trans. on Power Electronics*, pp. 816-823, vol. 12, no. 5, Sept. 1997.
- [65] BSIM Homepage, <http://www-device.eecs.berkeley.edu/bsim3/>.
- [66] C. Enz, F. Krummenacher and E. Vittoz, "An analytical MOS transistor model valid in all regions of Operation and dedicated to low-voltage and low-current applications," *Journal on Analog Integrated Circuits and Signal Processing*, Kluwer Academic Publishers, pp. 83-114, July 1995.
- [67] G. Gildenblat, H. Wang, T.-L. Chen, X. Gu and X. Cai, "SP: An advanced surface-potential-based compact MOSFET model," *IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1394-1406, Sept. 2004.
- [68] K. A. Sakallah, Y. T. Yen and S. S. Greenberg, "First Order Charge Conserving MOS Capacitance Model," *IEEE Trans. on Computer-Aided Design*, vol. 9, no. 1, pp. 99-108, Jan. 1990.
- [69] T. R. McNutt, A. R. Hefner, H. A. Mantooth, J. L. Duliere, D. Berning and R. Singh, "Silicon carbide PiN and merged PiN schottky power diode models implemented in the Saber circuit simulator," *IEEE Trans. on Power Electronics*, vol. 19, no. 3, pp. 573-581, May 2004.