Determination of the Steady State of an Oscillator by a Combined Time–Frequency Method

Martin H. Schwab

Abstract—**This paper presents a new method for the computation of the steady state of nonlinear oscillators including distributed elements which exploits advantages of both time-domain and frequency-domain simulation. The oscillator network is divided into a linear subnetwork described by a hybrid matrix in the frequency domain and a nonlinear subnetwork represented by a set of first-order nonlinear differential equations solved in the time-domain. The periodic steady state of the oscillator is shown to be equivalent to the solution of a boundary value problem, where the boundary conditions are given at a set of points along the time axis. For the solution of the boundary value problem the multiple shooting algorithm is applied, which may be modified in a very effective way owing to the special structure of the boundary value problem. It will be shown that the bandwidth in the nonlinear subnetwork can be chosen arbitrarily high regardless of the number of harmonics at the ports connecting the subnetworks. An error estimate for the neglected harmonics at the ports is derived, which does not require additional numerical effort. In order to demonstrate the feasibility of the method and to discuss the error mechanisms it is applied to two examples: a Clapp oscillator including a piecewise-linear characteristic and an integrated GaAs MESFET oscillator at 10.7 GHz.**

I. Introduction

MONOLITHIC integration of microwave circuits makes the accurate prediction of the behavior of oscillators an increasingly important task in modern CAD techniques. From a linear simulation such important characteristics of oscillators as frequency and output power can only be estimated, because certain dominant effects, such as the operating point shift and limitation mechanisms, are neglected. The use of traditional time-domain techniques as implemented, for example, in SPICE is the most general, but also the least advantageous, way to compute the steady state of an oscillator [1], [2]. Here no assumption on the periodicity of the solution is made and a quasi-periodic or chaotic solution may be simulated as well as a periodic one. Owing to the flexibility of modern integration algorithms the time-domain simulation may be done very effectively [1]. Nevertheless this simple method requires a high computational effort because of the unfavorable description of distributed elements and the stiffness of the oscillator network, i.e., the large difference between the time constants inherent in a typical oscillator circuit.

These problems can be avoided if a periodic solution is assumed, as is done in numerous algorithms. An excellent overview of the literature can be found in [4]. The most popular method is the harmonic balance algorithm, where the currents and voltages are described by Fourier series with a finite number of elements. Then the inner states of distributed elements can be eliminated and the differential equation describing the oscillator network is transformed into a complex algebraic equation, which is solved by using an algorithm of the Newton–Raphson type or by minimizing an objective function. Problems may arise with the harmonic balance method if the oscillator includes strong nonlinearities generating a large number of harmonics which have to be taken into consideration to achieve the required accuracy.

Another algorithm which underlies the presented algorithm is the shooting method [2], [3]. Here the state vector describing the oscillator network is varied until after one period of oscillation the state is the same. In the case of high-Q circuits this algorithm runs into severe convergence problems. Additionally the method still requires a discrete approximation of distributed elements and so is very impracticable in microwave circuits.

In this paper a new algorithm is presented which combines the advantages of both time-domain simulation and frequency-domain simulation. In Section II the network representation is described, which is based in a division of the network into a linear and a nonlinear subnetwork. The linear network part is represented by a hybrid matrix and can be computed with the methods of traditional linear network analysis. In Section III it is shown that the steady state of the oscillator network is equivalent to the solution of a nonlinear boundary value problem where the differential equation is given by the nonautonomous set of first-order differential equations and the boundary conditions are given at a set of sampling points and couple the two network subsets. The multiple shooting algorithm [5], [6], which shows a much wider basis of

Manuscript received November 5, 1990; revised April 4, 1991. This work was supported by the Deutsche Forschungsgemeinschaft DFG.

The author is with the Lehrstuhl f. Hochfrequenztechnik, Technische Universität München, Arcisstr. 21, D-8000 Munich 2, Federal Republic of Germany.

IEEE Log Number 9101026.

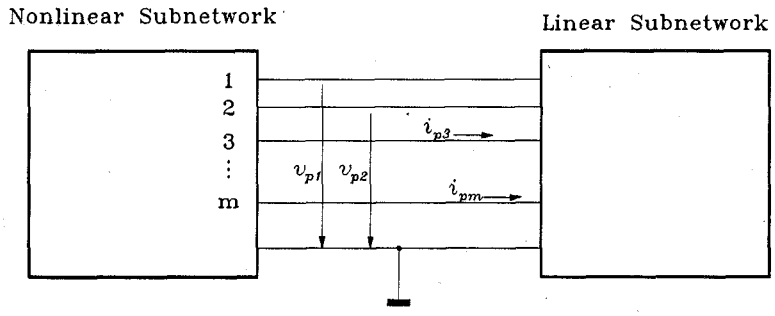


Fig. 1. Linear and nonlinear parts of oscillator network.

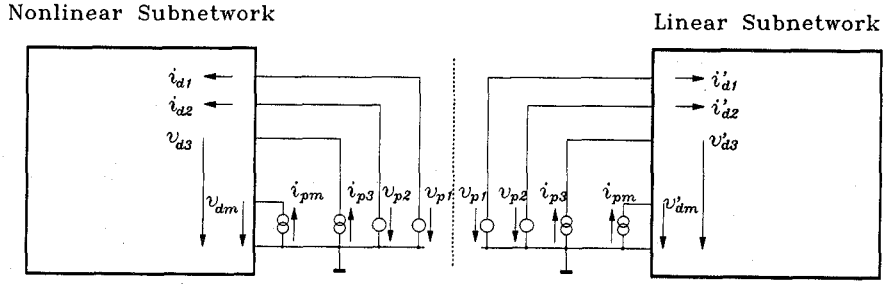


Fig. 2. Separation of the linear and nonlinear network parts.

convergence than the single shooting one, is applied to this boundary value problem in Section IV. Since the boundary conditions are no longer given at two points, as in the original multiple shooting algorithm, it must be modified appropriately. The special structure of the differential equations and boundary conditions is exploited to accelerate the numerical calculations. To demonstrate the practical use of the algorithm as a whole and to discuss its error mechanisms, it is applied to two examples in Section V. The first, a simple oscillator network, shows the advantages of the method in the analysis of strong nonlinearities. In a second example the steady state of an oscillator at a frequency of 10.7 GHz is computed.

II. Network Representation

The description of the network is based on a division into two subsets as shown in Fig. 1. A nonlinear subnetwork contains all nonlinearities of the oscillator network and some linear elements embedding the nonlinearities. These linear embedding elements show in many cases a low-pass structure owing to parasitic elements such as bond wire inductance, package capacitance, and loss resistances. They may be used to reduce the numerical effort drastically, as demonstrated later. In the other subnetwork all linear network elements which are appropriately described in the frequency domain are collected, such as distributed elements. The two network parts of the oscillator are connected by m ports. In a subsequent step the linear and the nonlinear part of the network are separated from each other as shown in Fig. 2. At each port, either the voltage or the current is replaced by an equivalent source. These sources are collected in the vector $\boldsymbol{l}$.

First we will present a description of the nonlinear subnetwork. Based on the methods of state analysis this subnetwork is described by a set of nonlinear first-order differential equations:

$$\frac{d}{dt}\boldsymbol{x} = \boldsymbol{A}(z)\boldsymbol{x} + \boldsymbol{B}(z)\boldsymbol{w}(z) + \boldsymbol{C}(z)\dot{\boldsymbol{w}}(z) + \boldsymbol{D}(z)\boldsymbol{l},$$

$$\boldsymbol{x} \in R^n, \boldsymbol{l} \in R^m \quad (1)$$

where $\boldsymbol{x}$ is the vector of the state variables and z is the vector of the variables controlling the nonlinearities. Nonlinear resistors and nonlinear conductances have been replaced by controlled voltage and current sources and are assembled together with all other sources by the vector $\boldsymbol{w} = [\boldsymbol{v}_0, \boldsymbol{i}_0]^T$. Since we have so far imposed no restriction on the network allowing, for example, nonlinear capacitors and inductors, the vector $\boldsymbol{w}$ and the matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ given by the elements and the topology of the nonlinear network depend on z, which controls the nonlinearities. A simple example will demonstrate this: the current through a diode may be estimated by $i_D = (1 - e^{u/u_T})$, where u is the voltage on the diode. In this case the controlling quantity is given by $z = u$. Equation (1), which is implicit in general since z depends on $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$, will degenerate to an explicit set of differential equations if two restrictions are imposed: First, we exclude cases where the controlling z is a function of $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ and assume that the nonlinearities are controlled by the state variables only. Second, we assume that no

loop exists including a controlled voltage source and that the loops contain only capacitors and voltage sources. By analogy we exclude the dual case, where a controlled current source is connected to a node which is connected only to current sources and inductors.

With these restrictions, the matrix $\boldsymbol{C}$ degenerates to 0 and then we obtain from (1)

$$\frac{d}{dt}\boldsymbol{x} = \boldsymbol{A}(\boldsymbol{x})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{x})\boldsymbol{w}(\boldsymbol{x}) + \boldsymbol{D}(\boldsymbol{x})\boldsymbol{l}. \tag{2}$$

Now the set of equations is in a form to be integrated with a standard algorithm. The state vector gives a single-valued representation of the nonlinear network. Therefore all network currents and network voltages can be derived from $\boldsymbol{x}$. The vector $\boldsymbol{c}$ containing the dual variables to the port source vector $\boldsymbol{l}$ is given by

$$\boldsymbol{c} = \boldsymbol{F}\boldsymbol{x} + \boldsymbol{G}\boldsymbol{w} + \boldsymbol{H}\boldsymbol{l}. \tag{3}$$

The matrices $\boldsymbol{F}$, $\boldsymbol{G}$, and $\boldsymbol{H}$ are derived in a way similar to that for the matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ from topology and elements of the network. Equations (2) and (3) provide a complete description of the nonlinear subnetwork. A computer algorithm for generating these equations can be found in [1].

At the linear subnetwork the port variables $\boldsymbol{l}$ and $\boldsymbol{c}'$ are connected by their Fourier transforms in the frequency domain:

$$\boldsymbol{V}(\omega)\boldsymbol{L}(\omega) = \boldsymbol{C}'(\omega), \qquad \boldsymbol{V}(\omega) \in C^{m\times m} \tag{4}$$

where $\boldsymbol{V}(\omega)$ is the hybrid matrix of the linear subnetwork. It is assumed that $\boldsymbol{V}(\omega)$ is given by standard network analysis or by a field-theory-based computation.

Now a favorable aspect of the network separation becomes apparent. In a pure time-domain formulation, high-Q circuits have slow time constants, leading to long transients and small variations of the state variables after one period of oscillation. This requires excessive numerical effort in integration and causes a nearly singular Jacobian in a shooting method. This will entail excessive numerical effort if the steady state is obtained by integration, and requires extremely high integrator precision if a shooting method is used. If the feedback structure is interrupted by the network separation, the slow time constants of the resonator will have no influence on the time-domain integration in the nonlinear subnetwork, while a frequency-domain approach for the linear subnetwork involves the most appropriate basis function for high-Q circuits.

III. The Steady State as the Solution to a Boundary Value Problem

In this section we deduce a boundary value problem whose solution is equal to the steady state of the oscillator. The best way to do this is to start from an oscillator network described completely in the time domain. In our method this is equivalent to the case where the nonlinear subnetwork contains the whole oscillator network and the number of ports is zero. Then we obtain from (2)

$$\frac{d}{dt}\boldsymbol{x} = \boldsymbol{A}(\boldsymbol{x})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{x})\boldsymbol{w}(\boldsymbol{x}). \tag{5}$$

This set of differential equations is autonomous and represents the entire oscillator network. In the periodic state of an oscillator the state of the system must obviously repeat itself after each period of oscillation. Since the state of the system is uniquely defined by the state vector it is necessary and sufficient that

$$\boldsymbol{x}(t) = \boldsymbol{x}(t + T_0) \tag{6}$$

be fulfilled, where T_0 is the period of oscillation and t an arbitrary initial time. Equations (5) and (6) define a two-point boundary value problem which is unsolvable for two reasons: first of all we do not know T_0 *a priori*, and second we see that for any solution $\boldsymbol{x}_s(t_i)$ there exists an infinity of solutions $\boldsymbol{x}_s(t_i + s)$, $s \in R$. By a simple transformation this boundary value problem may be transformed into a solvable one. Therefore the time interval $t \in [t_i, t_i + T_0]$ is transformed to $\tau \in [0,1]$ introducing T_0 as a new "synthetic" state variable T_0 via

$$\frac{d}{d\tau}\boldsymbol{x} = [\boldsymbol{A}(\boldsymbol{x})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{x})\boldsymbol{w}(\boldsymbol{x})]T_0$$

$$\frac{dT_0}{d\tau} = 0 \qquad \tau = \frac{t}{T_0}. \tag{7}$$

Notice, that we have now $n+1$ differential equations and therefore need one additional boundary condition, which is given by, for example,

$$\frac{d}{dt}x_\nu(\tau = 0) = 0 \quad \text{or} \quad x_\nu(\tau = 0) = 0. \tag{8}$$

This boundary condition should fix the phase of the limit cycle but must be applied cautiously as to ensure that x_ν has no local maxima or minima lying close together in the first case or not too large a dc offset in the second case.

We have derived now a boundary value problem whose solution is the steady state of a lumped oscillator network for the special case of the network description in the previous section. Treating the general case according to the network representation in the section above, the condition $\boldsymbol{x}(\tau = 0) = \boldsymbol{x}(\tau = 1)$ is necessary, but no longer sufficient. In addition to this condition the port voltages and currents of $\boldsymbol{c}(\tau)$ and $\boldsymbol{c}'(\tau)$ must coincide. By analogy to the pure time-domain method shown above, we trans-

form (2) and obtain

$$\frac{d\boldsymbol{x}}{d\tau} = (\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{w} + \boldsymbol{D}\boldsymbol{l})T_0 \qquad \frac{dT_0}{d\tau} = 0 \qquad \tau = \frac{t}{T_0}. \quad (9)$$

Since we assumed a periodic oscillation, the vector $\boldsymbol{l}(\tau)$ may be described by the Fourier series

$$\boldsymbol{l}(\tau) = \sum_{\nu=-\infty}^{+\infty} (\boldsymbol{L}_\nu^r + j\boldsymbol{L}_\nu^i) e^{j2\pi\nu\tau}. \quad (10)$$

From (4) we obtain

$$\boldsymbol{c}'(\tau) = \sum_{\nu=-\infty}^{+\infty} \boldsymbol{V}_\nu (\boldsymbol{L}_\nu^r + j\boldsymbol{L}_\nu^i) e^{j2\pi\nu\tau} \quad (11)$$

where $\boldsymbol{V}_\nu$ is the hybrid matrix of the linear multiport at the νth harmonic of the frequency $f_0 = 1/T_0$. Now the interaction between the subnetworks is limited to frequencies below $(k+1)f_0$. The Fourier series are truncated above their kth elements. Note that this does not impose a bandwidth restriction within the nonlinear subnetwork which is determined by the step size of the integrator. Then we obtain a set of $n+(2k+1)m$ differential equations:

$$\frac{d}{d\tau}\boldsymbol{x} = [\boldsymbol{A}(\boldsymbol{x})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{x})\boldsymbol{w}(\boldsymbol{x}) + \boldsymbol{D}(\tau)\boldsymbol{l}(\tau)]T_0$$

$$\frac{d}{d\tau}T_0 = 0$$

$$\frac{d}{d\tau}\boldsymbol{L} = 0 \quad (12)$$

where we added the Fourier coefficients $\boldsymbol{L} = [\boldsymbol{L}_0, \boldsymbol{L}_1^r, \boldsymbol{L}_1^i, \cdots, \boldsymbol{L}_k^i]^T$ as state variables. To fix the phase of the limit cycle we set the first element of $\boldsymbol{L}_1^i$ to zero and omit it in the vector $\boldsymbol{L}$. Contrary to (8), this condition implies no loss of generality. For the $n+(2k+1)m$ differential equations given in (12) the same number of boundary conditions is needed. The first n conditions result from the condition of periodicity for the state variables:

$$\boldsymbol{x}(\tau = 0) = \boldsymbol{x}(\tau = 1). \quad (13)$$

To obtain the resting $(2k+1)m$ conditions the following considerations are made: we have assumed periodic and bandwidth-limited port voltages and currents $\boldsymbol{l}(t)$ and $\boldsymbol{c}'(t)$. Now we assume, that $\boldsymbol{c}(t)$ is limited to k harmonics, too. Then it is necessary and sufficient for $\boldsymbol{c}'(t)$ and $\boldsymbol{c}(t)$ to coincide at $2k+1$ sampling points. This leads to $(2k+1)m$ conditions:

$$\boldsymbol{c}(\tau_\nu) - \boldsymbol{c}'(\tau_\nu) = 0, \qquad \tau_\nu = \frac{\nu-1}{2k+1}, \quad \nu = 1, 2, \cdots, 2k+1. \quad (14)$$

With (12) as differential equation and (13) and (14) as boundary conditions, we now have derived an $n+(2k+1)m$-dimensional nonlinear boundary value problem. Notice that the boundary conditions are no longer given only at $\tau = 0$ and $\tau = 1$, as in the pure time-domain boundary value problem from (6), (7), and (8), but rather at $2k+2$ points with $\tau = (\nu-1)/(2k+1)$, $\nu = 1, \cdots, 2k+2$. This makes the standard boundary value problem solver inapplicable.

IV. Numerical Solution of Boundary Value Problem

A. The Nonlinear Vector Valued Shooting Function

Having shown that the solution of the boundary value problem given by (12), (13), and (14) is equal to the steady state of the oscillator, we apply and modify the multiple shooting algorithm [5], [6]. Multiple shooting has a wider region of convergence; i.e., the method works with a worse initial guess than single shooting [6]. From now on the extended state vector, containing $\boldsymbol{x}$, T_0, and $\boldsymbol{L}$, is denoted as $\boldsymbol{y} = [\boldsymbol{x}, T_0, \boldsymbol{L}]^T$, $\boldsymbol{y} \in R^{n+(2k+1)m}$. From (13) and (14) we define a nonlinear boundary function

$$\boldsymbol{R}\left(\boldsymbol{y}(\tau=0), \boldsymbol{y}\left(\tau = \frac{1}{2k+1}\right), \cdots, \boldsymbol{y}(\tau=1)\right)$$

$$= \begin{bmatrix} \boldsymbol{x}(1) - \boldsymbol{x}(0) \\ \boldsymbol{c}'(0) - \boldsymbol{c}(0) \\ \boldsymbol{c}'\left(\frac{1}{2k+1}\right) - \boldsymbol{c}\left(\frac{1}{2k+1}\right) \\ \boldsymbol{c}'\left(\frac{2}{2k+1}\right) - \boldsymbol{c}\left(\frac{2}{2k+1}\right) \\ \vdots \\ \boldsymbol{c}'\left(\frac{2k}{2k+1}\right) - \boldsymbol{c}\left(\frac{2k}{2k+1}\right) \end{bmatrix}. \quad (15)$$

Our task will be to find a trajectory of $\boldsymbol{y}$ fulfilling $\boldsymbol{R} = 0$. We divide the interval $[0,1]$ into $2k+1$ equal sections as done previously in (14) and choose at each of these sections a starting point of $\boldsymbol{y}$: $\boldsymbol{y}(\tau = \kappa/2k+1) =: \boldsymbol{s}_\kappa$ as shown in Fig. 3. The solution of the initial value problem at $\tau = (\kappa+1)/(2k+1)$ with differential equation (12) and starting value $\boldsymbol{s}_\kappa$ will be denoted as

$$\boldsymbol{y}\left(\tau = \frac{\kappa+1}{2k+1}\right) = \boldsymbol{e}_{\kappa+1}(\boldsymbol{s}_\kappa). \quad (16)$$

We have to find now a set of starting vectors $\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_{2k+1}$ so that the discontinuities from $\boldsymbol{e}_{\kappa+1}(\boldsymbol{s}_\kappa)$ to $\boldsymbol{s}_{\kappa+1}$ vanish (see Fig. 3) as well as the boundary function $\boldsymbol{R}$:

$$\boldsymbol{R}(\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_{2k+1}, \boldsymbol{e}_{2k+2}(\boldsymbol{s}_{2k+1})) = 0. \quad (17)$$

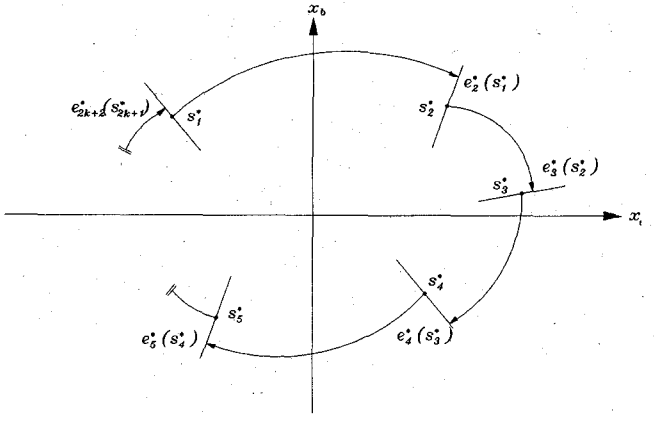


Fig. 3. The subdivision of the limit cycle into $2k+1$ sections portrayed as a two-dimensional projection into the plane x_a, x_b from state space.

Notice that in (17) we have not defined a starting vector at $\tau = 1$. The value at this point is given by the result of the initial value problem $e_{2k+2}(s_{2k+1})$. Finding a set of starting points with the required properties is equal to find the zeros of the vector valued function

$$U(s) = \begin{bmatrix} e_2(s_1) - s_2 \\ e_3(s_2) - s_3 \\ \vdots \\ e_{2k+2}(s_{2k+1}) - s_1 \\ c'(0) - c(0) \\ c'\left(\frac{1}{2k+1}\right) - c\left(\frac{1}{2k+1}\right) \\ c'\left(\frac{2}{2k+1}\right) - c\left(\frac{2}{2k+1}\right) \\ \vdots \\ c'\left(\frac{2k}{2k+1}\right) - c\left(\frac{2k}{2k+1}\right) \end{bmatrix} \tag{18}$$

with $s = [s_1, s_2, \cdots, s_{2k+1}]^T$. The special structure of the set of differential equations (12) enables us to simplify the vector valued function. With the introduction of the $(2k+1)m$ synthetic state variables we added trivial differential equations to the original system in (12). The solution of these trivial differential equations is isolated from the state vector and is easily seen to be constant in one subinterval $[\kappa/(2k+1), (\kappa+1)/(2k+1)]$. Since we desire a trajectory that is continuous at the boundaries of two subintervals, these additional state variables must be constant in the whole time interval $\tau \in [0,1]$. Therefore it is sufficient for them to be defined once. This leads to the simplified function

$$U^*(z) = \begin{bmatrix} e_2^*(s_1^*) - s_2^* \\ e_3^*(s_2^*) - s_3^* \\ \vdots \\ e_{2k+2}^*(s_{2k+1}^*) - s_1^* \\ c'(0) - c(0) \\ c'\left(\frac{1}{2k+1}\right) - c\left(\frac{1}{2k+1}\right) \\ c'\left(\frac{2}{2k+1}\right) - c\left(\frac{2}{2k+1}\right) \\ \vdots \\ c'\left(\frac{2k}{2k+1}\right) - c\left(\frac{2k}{2k+1}\right) \end{bmatrix} \tag{19}$$

with $s^* = [s_1^*, s_2^*, \cdots, s_{2k+1}^*]^T$, where s_κ^* denotes the starting point $s_\kappa^* = x(\tau = \kappa/(2k+1))$. This quantity e^* describes the first n variables of y containing the state variables and z is defined by $z = [s^*, T_0, L]$.

Since we will need the Jacobian $\partial U^*/\partial z$ when trying to find the zeros of U^*, as demonstrated next, we will first discuss its special structure and show how it can be efficiently computed. In general, most of the CPU time required by a root-finding algorithm is spent in the evaluation of the Jacobian, which is not given explicitly in most cases and has to be estimated via numerical differentiation demanding many function evaluations. Because of the special structure of U^* we can reduce the numerical effort significantly. Regarding the first row of U^* we notice that it depends only on s_1^*, s_2^*, T_0, and L. Similarly, the second row depends only on s_2^*, s_3^*, T_0, and L. Note that the partial derivative of the vector $c'(\kappa) - c(\kappa)$ with regard to s_ν^* is nonzero only for $\kappa = \nu$. We obtain, for the whole Jacobian,

$$J = \frac{\partial U^*}{\partial z} = \begin{bmatrix} \phi_1^x & -I & & & & \phi_1^k \\ & \phi_2^x & -I & & & \phi_2^k \\ & & \ddots & \ddots & & \vdots \\ & & & \phi_{2k}^x & -I & \phi_{2k}^l \\ -I & & & & \phi_{2k+1}^x & \phi_{2k+1}^k \\ \varphi_1^x & & & & & \varphi_1^k \\ & \varphi_1^x & & & & \varphi_2^k \\ & & \ddots & & & \vdots \\ & & & & \varphi_{2k+1}^x & \varphi_{2k+1}^k \end{bmatrix} \tag{20}$$

with the matrices

$$\phi_\kappa^x = \frac{\partial e_{\kappa+1}^*(s_\kappa^*)}{\partial s_\kappa^*}, \qquad \phi_\kappa^x \in R^{n\times n}$$

$$\phi_\kappa^k = \left[\frac{\partial e_{\kappa+1}^*(s_\kappa^*)}{\partial T_0}, \frac{\partial e_{\kappa+1}^*(s_\kappa^*)}{\partial L}\right], \qquad \phi_\kappa^k \in R^{n\times m(2k+1)}$$

$$\varphi_\kappa^x = \frac{\partial\left[c'\left(\frac{\kappa-1}{2k+1}\right) - c\left(\frac{\kappa-1}{2k+1}\right)\right]}{\partial s_\kappa^*}, \qquad \varphi_\kappa^x \in R^{m(2k+1)\times n}$$

$$\varphi_\kappa^k = \left[\frac{\partial\left[c'\left(\frac{\kappa-1}{2k+1}\right) - c\left(\frac{\kappa-1}{2k+1}\right)\right]}{\partial T_0}, \frac{\partial\left[c'\left(\frac{\kappa-1}{2k+1}\right) - c\left(\frac{\kappa-1}{2k+1}\right)\right]}{\partial L}\right], \quad \varphi_\kappa^k \in R^{n\times m(2k+1)}. \tag{21}$$

In (20) only the matrices ϕ_κ^x, ϕ_κ^k and the first column of φ_κ^k must be computed via numerical differentiation. The matrices φ_κ^x and φ_κ^k can be calculated directly by differentiating (15) with respect to y:

$$\frac{\partial\left[c'\left(\frac{\kappa-1}{2k+1}\right) - c\left(\frac{\kappa-1}{2k+1}\right)\right]}{\partial s_\kappa^*} = -\left(F + G\frac{\partial w}{\partial s_\kappa^*}\right)$$

$$\frac{\partial\left[c'\left(\frac{\kappa-1}{2k+1}\right) - c\left(\frac{\kappa-1}{2k+1}\right)\right]}{\partial L}$$

$$= \frac{\partial\left[\sum_{\nu=-k}^{+k} V_\nu\left(L_\nu^r + jL_\nu^i\right)e^{j2\pi\nu(\kappa-(1/2)k+1)}\right]}{\partial L}$$

$$- H\frac{\partial\left[\sum_{\nu=-k}^{+k}\left(L_\nu^r + jL_\nu^i\right)e^{j2\pi\nu(\kappa+(1/2)k+1)}\right]}{\partial L}. \tag{22}$$

The sparse matrix structure of the Jacobian allows efficient memory management and the use of sparse matrix techniques.

At this point the problems arising in a simple time-domain analysis of a very weakly damped system become apparent. The system is represented according to the network description in (2) and (3) by a 0-port: $m = 0$. The matrices ϕ_κ^k, φ_κ^x, and φ_κ^k vanish. For the weakly damped system ϕ_κ^x comes close to the identity matrix $\phi_\kappa^k \approx I$ and J is ill-conditioned or singular. The root-finding algorithm described in the next subsection, however, assumes a nonsingular Jacobian.

B. The Modified Newton–Raphson Algorithm

To find the zeros of $U^*(z)$ we apply the modified Newton–Raphson algorithm. Since this is a well-known standard algorithm, the discussion here will focus on certain details of norming and step size control.

With the Newton–Raphson algorithm the nonlinear function first is linearized, with terms of higher order neglected. From this linearized function the zeros can be found by

$$U^*(z) + J\Delta z = 0. \tag{23}$$

Solving the linear problem

$$J\Delta z = -U^*(z) \tag{24}$$

leads to Δz. The Newton–Raphson algorithm is an iterative process where at every iteration step the estimation of z is corrected via

$$z^{(\iota+1)} = z^{(\iota)} + \Delta z^{(\iota)} \tag{25}$$

where we denote $z^{(\iota)}$ as the value of z at the ιth iteration step. If the first estimation of $z^{(0)}$ is too far from the solution the algorithm will not converge to the solution. In order to enlarge the basin of convergence we introduce a relaxation factor $\lambda^{(\iota)}$ and obtain

$$z^{(\iota+1)} = z^{(\iota)} + \lambda^{(\iota)}\Delta z^{(\iota)}, \qquad 0 < \lambda^{(\iota)} < 1 \tag{26}$$

which is called the damped Newton algorithm. The literature presents a number of methods for the choice of $\lambda^{(\iota)}$. To avoid numerical problems with an ill-conditioned Jacobian we introduce a weighting matrix $W^{(\iota)}$ for the ιth iteration step:

$$W^{(\iota)} = \operatorname{diag}\left[w_1^{x(\iota)}, w_2^{x(\iota)}, \cdots, w_{2k+1}^{x(\iota)}, w_1^{l(\iota)}, w_2^{l(\iota)}, \cdots, w_{2k+1}^{l(\iota)}\right] \tag{27}$$

with

$$w_1^{x(\iota)} = \operatorname{diag}\left[\frac{1}{2(2k+1)}\sum_{\nu=1}^{2k+1}\left|\left(s^{*(\iota)}\right)^2 + \left(s^{*(i-1)}\right)^2\right|\right] \tag{28}$$

$$w_2^{x(\iota)} = w_1^{x(\iota)}$$

$$\vdots$$

$$w_{2k+1}^{x(\iota)} = w_1^{x(\iota)}$$

$$w_1^{l(\iota)} = \operatorname{diag}\left[\frac{1}{k}\sum_{\nu=1}^{k}\left[\left(L_\nu^{r(\iota)}\right)^2 + \left(L_\nu^{l(\iota)}\right)^2\right]\right]$$

$$w_2^{l(\iota)} = w_1^{l(\iota)}$$

$$\vdots$$

$$w_{2k+1}^{l(\iota)} = w_1^{l(\iota)} \tag{29}$$

where $w_1^{x(i)}, \cdots, w_{2k+1}^{x(i)}$ averages the state variables and $w^{I(i)}$ the Fourier coefficients.

We weight (20) and obtain from (24)

$$\tilde{J} = JW \qquad \tilde{z} = W^{-1}z$$
$$\tilde{J}\tilde{z} = -U^*(z). \tag{30}$$

This transformation improves the condition number of the linear equation system and avoids numerical sensitivity when, for example, one state variable is in the range of 10 V and another in the range of 1 μA.

The relaxation factor $\lambda^{(i)}$ is chosen according to [7]:

$$\lambda^{(i)} = \begin{cases} 1 & \text{if } \mu^{(i)} \geqslant 0.7 \\ \mu^{(i)} & \text{if } \mu^{(i)} < 0.7 \end{cases} \tag{31}$$

with

$$\mu^{(i)} = \frac{\|\Delta\tilde{z}^{(i)}\|_2}{\|\Delta\tilde{z}^{(i)} - \Delta\tilde{v}^{(i)}\|_2} \tag{32}$$

where $\Delta\tilde{v}^i$ comes from a "simplified Newton step":

$$\tilde{J}^{(i-1)}\Delta\tilde{v}^{(i)} = -U^{(i)}(z) \tag{33}$$

$\|\cdot\|_2$ denoting the Euclidean norm of a vector.

Two level functions $T_1^{(i)}$ and $T_2^{(i)}$ evaluate the success of an iteration step. $T_1^{(i)}$ judges the relative error of $U^{(i)}$:

$$T_1^{(i)} = \left\| W^{(i)-1}U(z^{(i)}) \right\|. \tag{34}$$

The level function $T_2^{(i)}$ is given by the Euclidean norm of the correction $\Delta\tilde{z}^{(i)}$:

$$T_2^{(i)} = \|\Delta\tilde{z}^{(i)}\|_2^2. \tag{35}$$

An iteration step will be accepted if at least one of these level functions has decreased. Otherwise the step has failed; the step size $\lambda^{(i)}$ is decreased. The algorithm is terminated if both $T_1^{(i)}$ and $T_2^{(i)}$ are less than a chosen error ϵ:

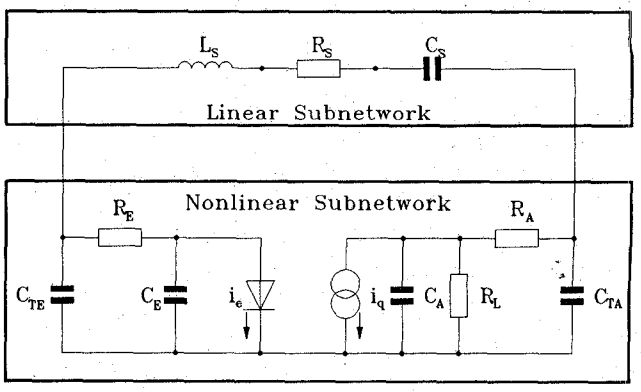
$$\epsilon \geqslant \max\left(T_1^{(i)}, T_2^{(i)}\right). \tag{36}$$

V. Examples and Error Mechanisms

In order to show the feasibility of the method two examples were chosen, the first representing a simple Clapp oscillator model including a piecewise-linear characteristic. It illustrates the formalism of network description and shows the advantage of the algorithm when analyzing oscillators with strong nonlinearities. With this simple example the error mechanisms are discussed. In a second example the algorithm is applied to a hybrid integrated GaAs oscillator with a MESFET as the active element in order to demonstrate the feasibility of the method to real microwave oscillators.

A. Clapp Oscillator

The equivalent circuit diagram of the first oscillator example is depicted in Fig. 4, including two nonlinearities: a nonlinear input resistance, $i_e((t) = f_1(v_e)$, and a nonlinear controlled current source, $i_q(t) = f_2(v_e)$. As shown in Fig. 4 the network is divided into a linear and a nonlinear subnetwork and the port current is replaced by the current source (see Fig. 5). The nonlinear subnetwork is represented by the equations



$C_{TE} = 3.5\ nF$ $C_E = 100\ pF$ $C_A = 100\ pF$ $C_{TA} = 7\ nF$
$C_S = 25\ pF$ $L_s = 1\ mH$ $R_E = 500\ \Omega$ $R_L = 1000\ \Omega$
$R_A = 1000\ \Omega$ $A_1 = 3 \cdot 10^{-5}\ A$ $A_2 = 0.5\ V$ $B_1 = 55 \cdot mS$
$B_2 = 55 \cdot mV$

Fig. 4. Linear and nonlinear parts of the oscillator network.

$$\begin{bmatrix} \dot{v}_{ce} \\ \dot{v}_{ca} \\ \dot{v}_{cte} \\ \dot{v}_{cta} \end{bmatrix} = \begin{bmatrix} \frac{-1}{C_eR_e} & 0 & \frac{1}{C_eR_e} & 0 \\ 0 & \frac{-R_a - R_l}{C_aR_aR_l} & 0 & \frac{1}{C_aR_a} \\ \frac{1}{C_{te}R_e} & 0 & \frac{-1}{C_{te}R_e} & 0 \\ 0 & \frac{1}{C_{ta}R_a} & 0 & \frac{-1}{C_{ta}R_a} \end{bmatrix} \cdot \begin{bmatrix} v_{ce} \\ v_{ca} \\ v_{cte} \\ v_{cta} \end{bmatrix}$$
$$+ \begin{bmatrix} \frac{-1}{C_e} & 0 \\ 0 & \frac{-1}{C_e} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} i_e \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{-1}{C_{te}} \\ \frac{1}{C_{ta}} \end{bmatrix} [i_p] \tag{37}$$

$$v_d = [0 \quad 0 \quad 1 \quad -1] \begin{bmatrix} v_{ce} \\ v_{ca} \\ v_{cte} \\ v_{cta} \end{bmatrix}. \tag{38}$$

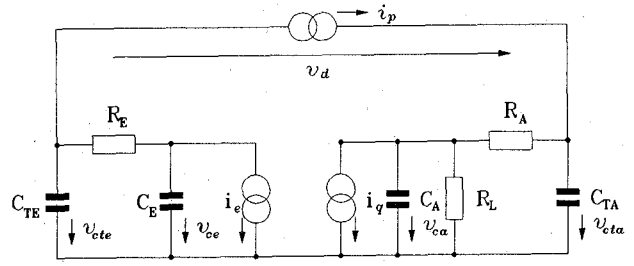


Fig. 5. Nonlinear subnetwork; the linear subnetwork is replaced with a current source.

TABLE I
STARTING VALUES FOR THE ITERATION PROCESS

$L_0 = 0$
$L_1 = 1 \cdot 10^{-2} + j0$
$L_2 = 0 + j0$
$L_3 = 0 + j0$
$L_4 = 0 + j0$
$L_5 = 0 + j0$
$T_0 = 1 \cdot 10^{-6}$
$\lambda^{(0)} = 0.01$
$k = 5$

The nonlinearities are chosen as

$$i_e(v_e) = A_1\left(\exp\frac{v_e}{A_2} - 1\right) \tag{39}$$

$$i_q(v_e) = \begin{cases} -\dfrac{B_1}{B_2} & \text{if } v_e \leqslant -B_2 \\ B_1 v_e & \text{if } -B_2 < v_E < B_2 . \\ \dfrac{B_1}{B_2} & \text{if } v_e \geqslant B_2 \end{cases} \tag{40}$$

Since we have only inserted a current source at the port, the hybrid matrix V degenerates to a simple impedance Z, which is obviously not defined at $\omega = 0$. Therefore we introduce v_0 instead of i_0 as a state variable and avoid the problem of the product of the terms $Z(\omega \to 0) \to 0$ and $I(\omega \to 0) \to \infty$ when matching $c'(t)$ and $c(t)$. The starting values and parameters for computation were chosen according to Table I. As an integrator for the solution of the initial-value problems, the Gear algorithm was applied [8]. The solution was obtained after eight iteration steps. On a Micro Vax 2000 the required CPU time was about 200 s. Fig. 6 depicts the course of the level function T_1 during the iteration process and shows a decay of about two orders of magnitude per iteration step near the solution, which is typical for the Newton algorithm.

This simple example shows a great advantage of the presented method, one which results from the flexibility of modern integration algorithms. In an integration algorithm such as Gear's, the evolution of the state variables over time is discretized and a polynomial is used to extrapolate to the next discretization point. The order of the polynom and the step size between the discretization points may be controlled using error estimates. The time

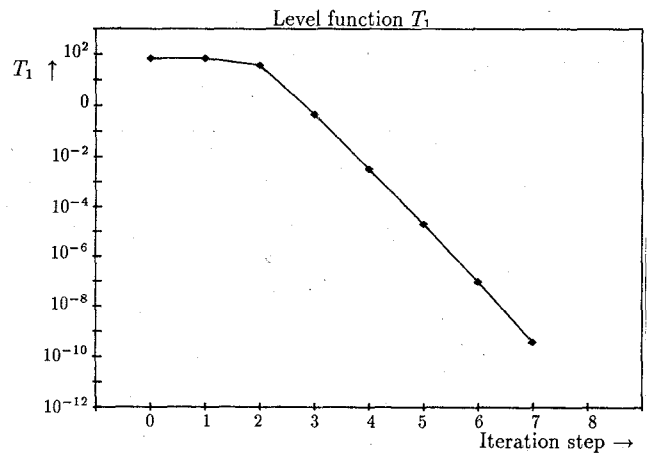


Fig. 6. Level function T_1 during the iteration process.

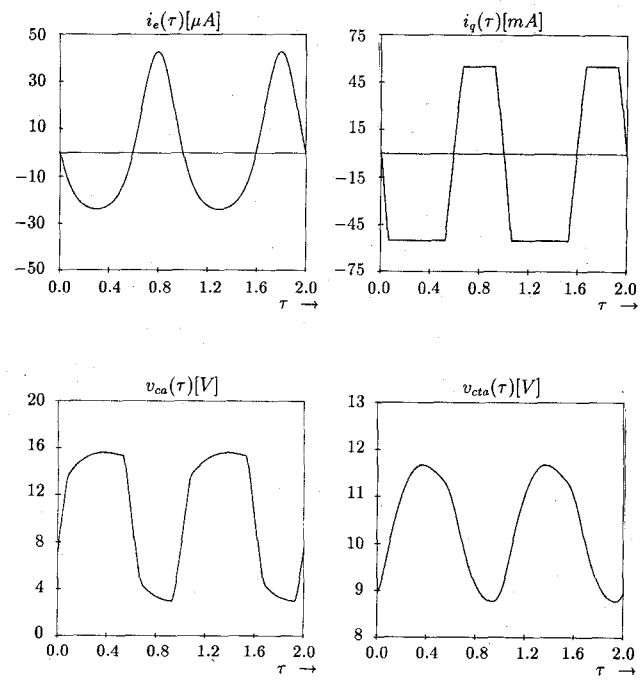


Fig. 7. Time plots of $i_e(\tau)$, $i_q(\tau)$, $v_{ca}(\tau)$, and $v_{cta}(\tau)$.

plot of i_q in Fig. 7 which was generated using three harmonics at the port shows that the bandwidth within the nonlinear subnetwork is truly independent of the bandwidth limitation at the port. With harmonic balance algorithms the time plots are interpolated by Fourier series. The order of these series must be chosen *a priori* and determines the step size between discretization points in time; i.e., all variables in the nonlinear subnetwork underlie the same bandwidth limitation as the port variables. It can easily be seen from i_q and v_{ca} in Fig. 7 that this is very inefficient in some cases. In our method the step size of the integration algorithm is independent of the number of harmonics allowed at the ports. This means that processes within the nonlinear subnetwork, in

TABLE II
RESULTS OF SIMULATION FOR VARIOUS k

k	1	2	3	4	5	6	7
ϵ_{RMS}	6.09310^{-2}	2.32910^{-2}	1.52710^{-2}	2.85110^{-3}	2.79510^{-3}	8.97010^{-4}	1.04310^{-4}
f[MHz]	1.011184	1.011283	1.011284	1.011275	1.011275	1.011276	1.011270
$\lvert L_{1,0}\rvert$[V]	−12.24	−11.43	−11.09	−11.04	−11.04	−11.03	−11.04
$\lvert L_{1,1}\rvert$[mA]	33.85	30.89	29.83	29.68	29.67	29.67	29.67
$\lvert L_{1,2}\rvert$[mA]	—	1.30710^{-2}	9.05510^{-3}	9.00810^{-3}	8.91510^{-3}	8.89310^{-3}	8.91510^{-3}
$\lvert L_{1,3}\rvert$[mA]	—	—	3.16910^{-3}	2.17210^{-3}	2.17410^{-3}	2.17210^{-3}	2.17410^{-3}
$\lvert L_{1,4}\rvert$[mA]	—	—	—	9.41410^{-4}	9.38110^{-4}	8.94310^{-4}	9.14910^{-4}
$\lvert L_{1,5}\rvert$[mA]	—	—	—	—	1.03110^{-4}	6.24810^{-5}	5.41510^{-5}
$\lvert L_{1,6}\rvert$[mA]	—	—	—	—	—	9.82210^{-5}	7.41510^{-5}
$\lvert L_{1,7}\rvert$[mA]	—	—	—	—	—	—	1.07110^{-5}

which higher harmonics affect lower harmonic contributions, can be calculated more accurately at the same number of port harmonics. Furthermore, regarding the evolution of the state variables over time, an unwanted oscillatory behavior of the active device itself may easily be detected while harmonic balance algorithms would give bad convergence as the only hint. This simple example shows the applicability of the algorithm to oscillators with strongly nonlinear behavior. We are now to discuss the error mechanisms of the algorithm, which may be divided into three main effects.

The first results from errors in the solution of the boundary value problem, especially of the solution of the zeros of (19). This error can be easily estimated by (36). The second error is caused by the subdivision of the entire oscillator network into two subnetworks. With the replacement of the linear subnetwork by current sources or voltage sources including k harmonics, the nonlinear subnetwork is terminated for harmonics of an order greater k with a short circuit when a voltage source was inserted and with an open circuit for the insertion of a current source. At this point another aspect of the replacement of the linear subnetwork with sources becomes apparent. An accurate result will be achieved if the kind of source is chosen with respect to the behavior of the linear subnetwork for $\omega \to \infty$. A parallel resonant circuit, for example, which leads to a short circuit for $\omega \to \infty$ should be replaced with a voltage source.

The third kind of error we will now discuss is caused by the truncation of the Fourier series and the violation of the sampling theorem. Recall the derivation of the boundary value problem. In a first step we replaced the linear m-port by m sources described by $\boldsymbol{I}(\tau)$. The assumption that $\boldsymbol{I}(\tau)$ is limited in bandwidth and may be described by Fourier series with a finite number of k harmonics leads to a truncation error. In a second step we assumed that the port voltages and currents $\boldsymbol{c}(\tau)$ dual to $\boldsymbol{I}(\tau)$ are limited in bandwidth to k harmonics too. Then it is sufficient for $\boldsymbol{c}(\tau)$ and $\boldsymbol{c}'(\tau)$ to coincide at $2k+1$ sampling points owing to the sampling theorem. In reality this assumption is not justified since harmonics with a higher order than k may arise in the nonlinear subnetwork. This leads to a violation of the sampling theorem.

A great advantage of the presented algorithm is that the error above may be estimated in a very simple way. To

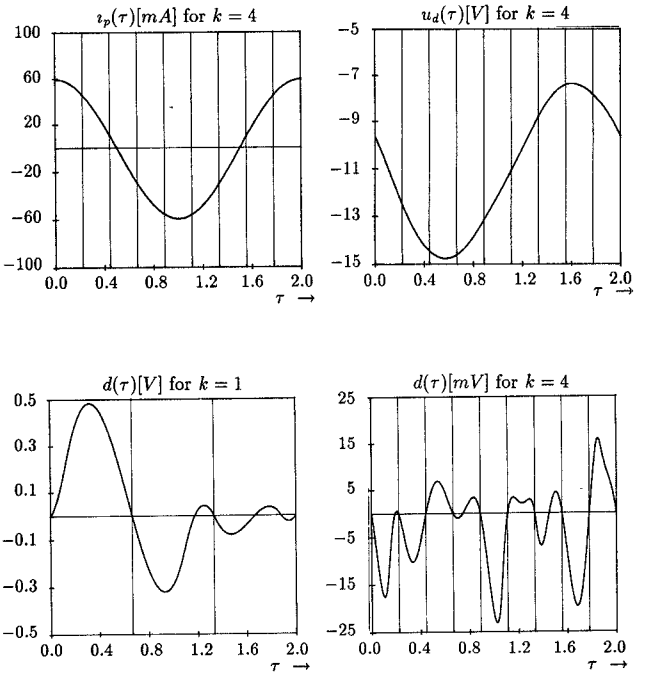


Fig. 8. Time plots of $i_p(\tau)$ and $u_d(\tau)$ for $k=4$ and of $d(\tau)$ for $k=1$ and $k=4$. The vertical lines denote the sampling points for the comparison of $\boldsymbol{c}(\tau)$ and $\boldsymbol{c}'(\tau)$.

that end we compare the port voltages and currents $\boldsymbol{c}(\tau)$ and $\boldsymbol{c}'(\tau)$ and define the difference $\boldsymbol{d}(\tau)$ as

$$\boldsymbol{d}(\tau) = \boldsymbol{c}(\tau) - \boldsymbol{c}'(\tau). \tag{41}$$

The solution of the boundary value problem implies that $\boldsymbol{d}(\tau)$ becomes zero at the sampling points. The existence of harmonics of higher order than k and the violation of the sampling theorem induces a $\boldsymbol{d}(\tau)$ unequal to zero off the sampling points. A simple criterion for error is the root mean square ϵ_{RMS} of $\boldsymbol{d}(\tau)$ normalized to $\boldsymbol{c}(\tau)$ and $\boldsymbol{c}'(\tau)$ by

$$\epsilon_{RMS} = \sum_{\nu=1}^{m} \frac{\int_{\tau=0}^{1} d_\nu^2(\tau)\, d\tau}{\int_{\tau=0}^{1} \left(c_\nu(\tau) - \bar{c}_\nu\right)^2 d\tau}, \qquad \bar{c}_\nu = \int_{\tau=0}^{1} c_\nu(\tau)\, d\tau. \tag{42}$$

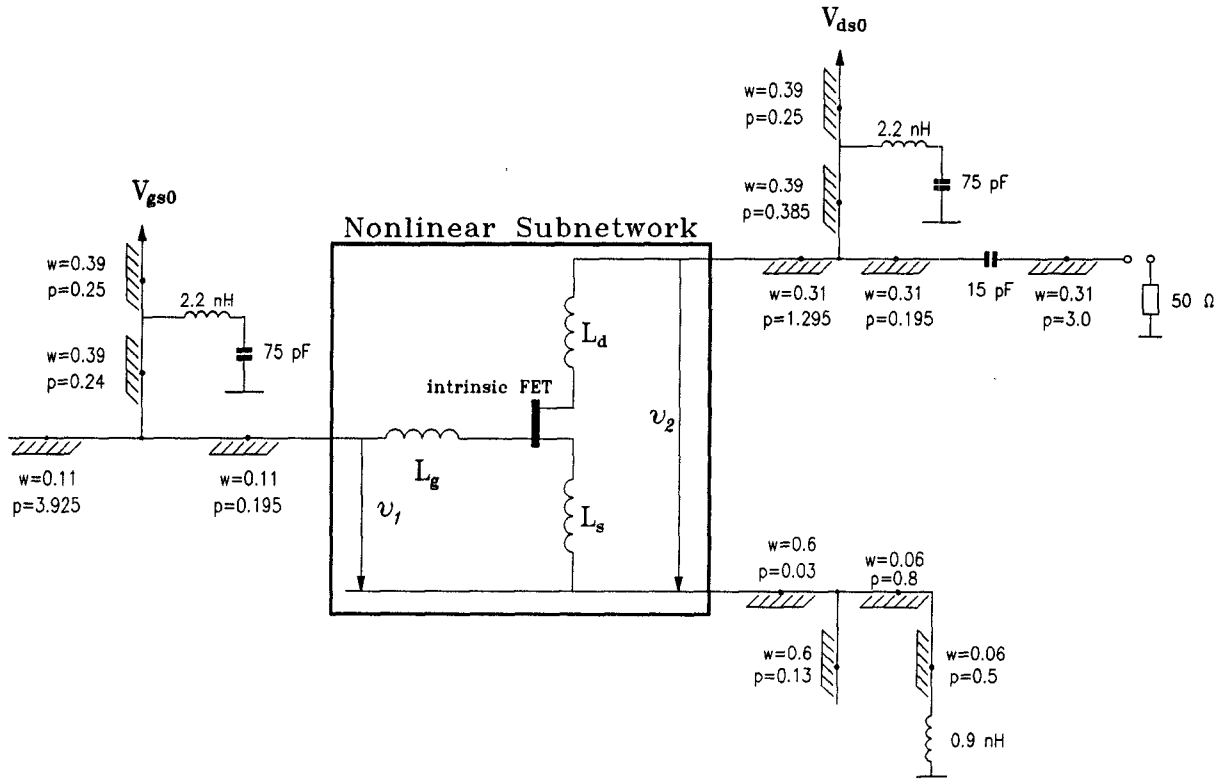
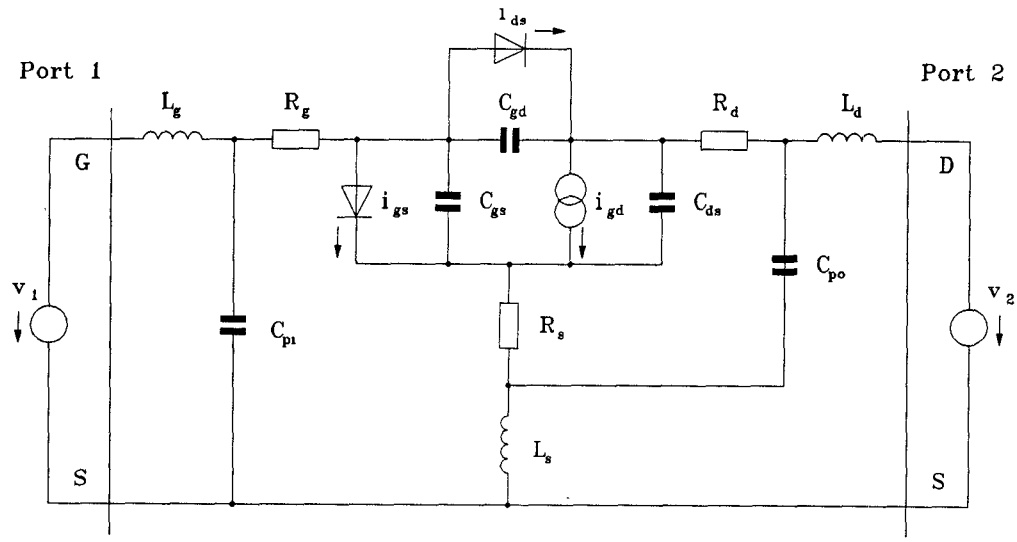


Fig. 9. Block circuit diagram of the GaAs MESFET oscillator.



$C_{pi} = 31\ fF$	$C_{po} = 1.5\ fF$	$C_{ds} = 0.17\ pF$	$C_{ds0} = 0.32\ pF$
$C_{gs0} = 12\ fF$	$L_g = 0.224\ nH$	$L_d = 0.1226\ nH$	$L_s = 95\ pH$
$R_g = 6.0\ \Omega$	$R_s = 4.1\ \Omega$	$R_d = 7.4\ \Omega$	$A_0 = 0.07384\ A$
$A_1 = 0.06066\ \frac{A}{V}$	$A_2 = -0.02553\ \frac{A}{V^2}$	$A_3 = -0.02250\ \frac{A}{V^3}$	$\beta = 0.05183\ \frac{1}{V}$
$\gamma = 2.84177\ V$			

Fig. 10. Network separation and MESFET model.

The quantity ϵ_{RMS} may be used to control the error discussed above. The order of the Fourier series is increased until ϵ_{RMS} is less than a specified tolerance. To illustrate the mechanisms discussed, the example was computed with $k = 1,2,\cdots,7$. Table II compiles the results. We see that port voltages and currents are almost completely determined by the harmonics of order 0 and 1. Therefore ϵ_{RMS} decreases slowly for $k \geqslant 1$. This can be seen in Fig. 8 as well, where the time plots of $d(\tau)$ and the sampling points are shown for $k = 1$ and $k = 4$. We see that $d(\tau)$ becomes 0 at the sampling points.

B. Hybrid GaAs Oscillator

In a second example the algorithm is applied to a hybrid GaAs oscillator at a frequency of 10.7 GHz. The block circuit diagram is depicted in Fig. 9. For the active element, a GaAs MESFET, a model according to [9] and [10], was chosen. The parameters were extracted by matching the model to S-parameter and dc measurements. The network separation and the MESFET model are shown in Fig. 10. The transistor nonlinearities are given by

$$i_{ds} = \left(A_0 + A_1 v_1 + A_2 v_1^2 + A_3 v_1^3\right) \tanh\left(\gamma v_{ds}\right) \tag{43}$$

$$c_{gs} = \begin{cases} C_{gs0} \dfrac{1}{\sqrt{1 - \dfrac{v_{cgs}}{v_B}}} & \text{if } v_{cgs} < F_c v_B \\ C_{gs0}\left(1 + \dfrac{v_{cgs} - F_c v_B}{2\left(v_B - F_c v_B \sqrt{1 - F_c}\right)}\right) & \text{if } v_{cgs} \geqslant F_c v_B \end{cases} \tag{44}$$

$$c_{gd} = \begin{cases} C_{gd0} \dfrac{1}{\sqrt{1 - \dfrac{v_{cgd}}{v_B}}} & \text{if } v_{cgd} < F_c v_B \\ C_{gd0}\left(1 + \dfrac{v_{cgd} - F_c v_B}{2\left(v_B - F_c v_B \sqrt{1 - F_c}\right)}\right) & \text{if } v_{cgd} \geqslant F_c v_B \end{cases} \tag{45}$$

$$i_{gs} = \begin{cases} I_s\left(\exp\left(\dfrac{v_{gs}}{n v_T} - 1\right)\right) + G_{\min} v_{gs} & \text{if } v_{gs} \geqslant 5 n v_t \\ -I_s + G_{\min} v_s & \text{if } -v_{br} + 50 v_T < v_{gs} < 5 n v_T \\ -I_s\left(\exp\left(-\dfrac{v_{gs} + v_{br}}{v_T} - 1\right)\right) + G_{\min} v_{gs} & \text{if } v_{gs} \leqslant -v_{br} + 50 v_t \end{cases} \tag{46}$$

$$i_{ds} = \begin{cases} I_s\left(\exp\left(\dfrac{v_{ds}}{n v_T} - 1\right)\right) + G_{\min} v_{ds} & \text{if } v_{ds} \geqslant 5 n v_t \\ -I_s + G_{\min} v_s & \text{if } -v_{br} + 50 v_T < v_{ds} < 5 n v_T \\ -I_s\left(\exp\left(-\dfrac{v_{ds} + v_{br}}{v_T} - 1\right)\right) + G_{\min} v_{ds} & \text{if } v_{ds} \leqslant -v_{br} + 50 v_t \end{cases} \tag{47}$$

(48)

with

$$v_1 = v_{gs}\left(1 + \beta\left(v_{ds} - v_{ds0}\right)\right). \tag{49}$$

As integrator the Gear algorithm was used. The solution for the operating point $V_g s0 = -0.5$ V and $V_{ds0} = 3.0$ V with eight significant numbers was obtained after eight iteration steps. The results for this operation point compared with measurements are represented in Table III. The output power was measured with a Hewlett Packard HP8569a spectrum analyzer. The steady state of the oscillator was computed as a function of the dc operating point, i.e., of V_{ds0} (see Fig. 11). The solution for one operating point was used here as a starting value for the neighboring operating point, so the algorithm was always near the solution. Since we made only small parameter steps, the algorithm converged within a few iteration steps. Note that for $V_{ds0} = 1.05$ in simulation and for $V_{ds0} = 1.35$ V in measurement, the oscillation vanishes. For the presented algorithm this means that the vector valued function (19) possesses no zeros except the trivial one, which is the stable dc operating point given by

$$\frac{d}{d\tau} \boldsymbol{x}_{st} = 0 \qquad \boldsymbol{L} = 0. \tag{50}$$

TABLE III
OUTPUT POWER AND FREQUENCY, WITH P_k DENOTING THE POWER OF THE kth HARMONIC

	Measured	Simulated
f	10.689 GHz	10.433 GHz
P_1	5.800 dBm	6.45 dBm
P_2	−2.5 dBm	−2.143 dBm

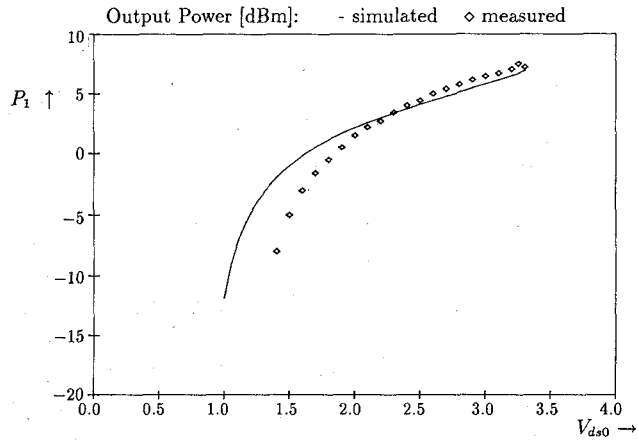


Fig. 11. Simulated and measured output power over various V_{ds0}.

The CPU time required for one singular dc operating point was about 600 s on a MicroVax 2000 if $k = 4$ is chosen.

VI. CONCLUSION

A novel method combining time-domain and frequency-domain approaches to determine the periodic steady state of oscillators has been presented. The method was derived in a general manner and then tested in cases which are difficult to handle either for pure time or frequency-domain simulators. Results on oscillators similar to the second example indicate that today harmonic balance implementations are still faster by a factor of between 2 and 10. However, the method described here implies no bandwidth limitation within the nonlinear subnetwork, since the bandwidth there is determined by the step size of the integrator, which is independent of the number of harmonics assumed at the ports. As impedance levels at higher harmonics are not reliable in common models for passive circuits, a restriction to a small number of harmonics in the linear subnetwork is reasonable. However, owing to fast processes in the semiconductor, a bandwidth restriction within the nonlinear subnetwork has a more serious effect on the results, since it excludes contributions of up-conversion and consequent down-conversion. Therefore a pure frequency-domain method may require the inclusion of higher harmonics and increased numerical effort to reach the same accuracy. Furthermore, the error discussion of the above model resulted in a definite improvement over pure frequency-domain methods, as a good error estimate for the Fourier series truncation is available at almost no cost. So there is no need for variation of the number of harmonics to gain confidence in the validity of the solution. While time-domain simulations are compulsory when no assumptions on the qualitative behavior of the oscillation can be made and while harmonic balance may be faster in well-behaved problems, the method presented here is expected to find its place in applications where fast processes in the active devices have to be modeled and where reliable error control is required.

ACKNOWLEDGMENT

The author wishes to thank Prof. Dr. P. Russer and Prof. Dr. P. Rentrop as well as Dipl. Ing. T. Göller and Dr. Ing. F.X. Kärtner for many helpful discussions. The hybrid integrated GaAs MESFET oscillator was designed and measured by Dipl. Ing. V. Güngerich. For the implementation of the algorithm the author is indebted to Dipl. Ing. B. Gruber.

REFERENCES

[1] L. O. Chua and P. Lin, *Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques*. Englewood Cliffs, NJ: Prentice-Hall, 1975.
[3] T. J. Aprille and T. N. Trick, "Steady-state analysis of nonlinear circuits with periodic inputs," *Proc. IEEE*, vol. 60, pp. 108–114, Jan. 1972.
[3] T. J. Aprille and T. N. Trick, "A computer algorithm to determine the steady state response of nonlinear oscillators," *IEEE Trans. Circuit Theory*, vol. CT-19, pp. 131–139, July 1972.
[4] V. Rizzoli and A. Neri, "State of the art and present trends in nonlinear microwave CAD techniques," *IEEE Trans. Microwave Theory Tech.*, vol. 36, pp. 343–365, Feb. 1988.
[5] R. Bulirsch, "Die Mehrzielmethode zur numerischen Lösung von nichtlinearen Randwertproblemen und Aufgaben der optimalen Steuerung," Heidelberg, West Germany: Carl-Cranz-Gessellschaft 1971 (reprinted Munich, West Germany: Technische Universität, Mathematisches Institut 1985).
[6] R. Bulirsch and J. Stoer, *Introduction to Numerical Analysis*. New York: Springer Verlag, 1980.
[7] P. Deuflhard, "A stepsize control for continuation methods with special application to multiple shooting techniques," Technical University Munich, Germany, Report TUM-MATH-7627, Dec. 1976.
[8] C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*. Englewood Cliffs, NJ: Prentice-Hall, 1971.
[9] W. R. Curtice and M. Ettenberg, "A nonlinear GaAs FET model for use in the design of output circuits for power amplifiers," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-33, pp. 1383–1391, Dec. 1985.
[10] H. Statz, R. A. Pucel, and P. Newman, "GaAs-FET device and circuit simulation in SPICE," *IEEE Trans. Electron Devices*, vol. ED-34, pp. 160–168, Feb. 1987.

Martin H. Schwab was born in Munich, Germany, in 1961. He received the Dipl. Ing. degree in 1988 from the Technical University of Munich. Since then he has worked as a Research Assistant there on the nonlinear computer-aided design of oscillator circuits.