In this paper dedicated methods for non-linear frequency domain problems as well as for linear problems are discussed. Harmonic Balance is a well known approach for performing non-linear frequency domain analyses of analogue circuit problems. When dealing with linear(ised) problems one often is interested in stability analysis. This can be provided by pole-zero analysis.

1 INTRODUCTION

Non-linear frequency domain analysis for circuit simulation is now-a-day usually done by Harmonic Balance. Here the driving variables are approximated by a combination of waves. The method treats linear elements in the frequency domain, and non-linear elements by switching between time and frequency domains, by an appropriate Fourier transform. Compared to conventional transient simulations with Fourier output functions, it has several advantages: no need to wait for initial transients to subside, efficient evaluation of intermodulation distortion, accurate low distortion levels, frequency and signal amplitude sweep capabilities, frequency-dependent elements, etc. For a good general introduction we refer to [6, 8]. The Harmonic Balance method has been implemented in the Philips in-house developed analogue circuit simulator, Pstar. We have aimed for robustness and efficiency together with high accuracy for all Fourier components. This puts demanding requirements on the algorithm. Firstly, Pstar’s non-linear solver should converge in practice for a large class of problems, which is accomplished by a cautious starting, stopping and relaxation strategy. Secondly, internal transformations should be very efficient and stable. Here the application area allows the exploitation of a carefully constructed Discrete Fourier Transform (DFT) to achieve this. Because Pstar has a hierarchically organised datastructure, the algorithm fits such a structure. This restricts the applicability of several numerical methods, but also offers opportunities to optimize the performance of others. Much attention has been paid in the literature to solve the linear systems arizing from a Newton process to solve the non-linear systems of equations. The success of iterative methods for solving linear systems for circuit simulation was very limited when dealing with transient or steady-state problems. However, recently these methods were shown to be more successfull when applied to the linear systems arising from Harmonic Balance [3, 12].

Pole-Zero analysis allows stability analysis of small signal behaviour of electronic circuits. For such an analysis the original equations are linearised. The results can also be used in the study of behavioural models or for the verification of a filter design. The poles and zeroes define the circuit transfer function for preselected input and output variables. The determination of poles and zeroes is done by solving a generalised eigenvalue problem in either case. We discuss our experiences to solve these eigenvalue problems by methods like QR, Arnoldi and JDQR (Jacobi-Davidson QR). We already remark that for a stability analysis one is interested in finding all eigenvalues, rather than in some selection.

The paper is organized as follows. Section 2 provides some general remarks concerning the basic equations that govern circuit simulation. The sections 3-7 deal with Harmonic Balance. Section 8 is completely devoted to Pole-Zero analysis.
2 KIRCHHOFF’S LAWS

The equations to be solved or studied are based primarily on Kirchhoff’s Current Law. Let \( v = (v^1, \ldots, v^N)^T \) be a time-dependent function, which will be the unknown (the voltage vector). In addition, we introduce the similar quantities \( i = i(v) \) (the current vector) and \( q = q(v) \) (the charge vector) and \( s \) (the source vector). Let \( t \) represent the time. With \( v, i \) and \( q \), the application of Kirchhoff’s Current Law in a Nodal Analysis yields a Differential-Algebraic-Equation (DAE) in the time-domain [1, 4, 9]

\[
j(v,t) \equiv i(v(t)) + \frac{d}{dt}q(v(t)) = s(t).
\]

The left-hand side of (1), denoted by \( j(v,t) \), represents the total nodal current. The DAE nature of (1) is caused by the fact that, for instance, the number of non-zero elements in \( q \) may be less than \( N \).

Notice that quantities like \( v, i \) and \( q \) may be interpreted in a generalised sense. In a Modified Nodal Analysis [9], adding unknowns may simplify implementation, but requires additional constitutive equations to be satisfied. For instance, after adding the current through an inductor as unknown, Kirchhoff’s Voltage Law may be applied, using the flux function \( \phi \) of the inductor.

The DC (direct current) solution \( v_{DC} \) is the steady state solution of

\[
i(v_{DC}) = s_{DC}.
\]

The DC solution is important because it is usually the initial solution for transient analysis. Also in the frequency domain it provides an initial solution for the system of equations to be solved by Harmonic Balance.

In AC (alternating current) analysis one studies (in the frequency-domain) the linearised equations for a small perturbation of \( v_{DC} \). AC analysis offers a way to improve the initial solution provided by the DC solution when performing an Harmonic Balance analysis. In AC the sources are simple sine-waves.

3 EQUATIONS SOLVED BY HARMONIC BALANCE

We assume \( d \) independent fundamental (angular) frequencies \( \lambda_j \). Let \( (\ldots) \) denote the complex inner-product and \( Z \) be the set of integers. We write \( v \) (and similarly \( i \) and \( q \)) in an expansion of complex exponentials

\[
v = \sum_{\omega_k \in \Lambda} V_k e^{i\omega_k t}, \text{ with } \omega_k \in \Lambda \equiv \{\omega|\omega = (k,\lambda)\},
\]

\[
k \equiv (k_1, k_2, \ldots, k_d)^T \in K \subset Z^d, \lambda \equiv (\lambda_1, \lambda_2, \ldots, \lambda_d)^T, \lambda_i > 0,
\]

where the (complex) \( V_k \) satisfies \( V_{-k} = \overline{V_k} \).

Here \( \lambda \) and \( k \) are uniformly for each component of \( v \). The set \( K \), containing integer tuples, is symmetrical about 0, while also 0 \( \in K \). \( K \) is assumed to be finite. With \( K \) we denote the number of non-negative (angular) frequencies (i.e. \( \omega_k \) with \( \omega_k \geq 0 \)). We also assume that all \( \omega_k \) are different and \( \omega_0 = 0 \).

The choice of the fundamental frequencies \( \lambda_j \) will depend on the kinds of (modified) sine-wave sources used. We note that \( \Lambda \) should contain a sufficiently rich set of interdistortion frequencies \( \omega_k \) like \( 2\lambda_1, \lambda_1 \pm \lambda_2, \ldots \) that are required in a distortion analysis. In practice, too restrictive a choice of the finite set of \( K \) may give rise to aliasing problems when compared with the analytical problem. For some sine-wave sources, a 1-D set of frequencies will be sufficient. In the sequel, we will restrict ourselves to \( d = 1 \) and \( d = 2 \).

Let \( V = (V^1, V^2, \ldots, V^N)^T \) be the Fourier transform of \( v \). More specifically, using a real notation, \( V^j = (V^{j,R}_0, V^{j,R}_1, V^{j,R}_2, \ldots, V^{j,R}_{K_j-1}, V^{j,I}_{K_j-1})^T \), in which \( V^j_k \equiv V^{j,R}_k + iV^{j,I}_k \) represents the \( k \)-th Fourier coefficient of \( v^j \).
With $\mathcal{F}$, we denote the mapping of the Fourier transform, thus $V = \mathcal{F}v$ and $v = \mathcal{F}^{-1}V$. The $\mathcal{F}$-transform of $i$ (and similarly for $q$ and $j$ and $s$) is defined by $I(V) = \mathcal{F}(\mathcal{F}^{-1}V) = \mathcal{F}i(v)$.

By this Galerkin approach, the frequency-domain equivalent of (1) simply becomes

$$ J = I(V) + \Omega Q(V) = S, $$

in which

$$ \Omega = \text{Block}_\text{Diag}(\Omega_K, \ldots, \Omega_K), $$

$$ \Omega_K = \text{Block}_\text{Diag}(0, -\omega_1, \ldots, 0, -\omega_{K-1}, \ldots, 0). $$

In the terminology of circuit analysis, the method of solving (1) by solving (5) is called the Harmonic Balance method. It is clear that the system given by (5) is a non-linear algebraic set of equations in the frequency-domain. In Pstar, the system is solved by performing a Newton-Raphson iteration.

### 4 SOURCES

The choice of the fundamental frequencies $\lambda_j$ depends on the kinds of sources used. For standard amplitude, frequency or phase modulated sources, a 2-D block of frequencies will usually be necessary, as explained below.

We assume voltage and current sources. The DC-sources are time-independent, the AC-sources may involve a simple sum of (co)sine-waves (SW-source). For Harmonic Balance the sources may also show amplitude modulation (SWAM), frequency modulation (SWFM) or phase modulation (SWPM) behaviour. Denoting a source by $s(t)$ and the carrier frequency and the signal frequency by $\omega_c$ and $\omega_s$, respectively, the following cases can be distinguished (here $\theta$ simply denotes a phase shift).

<table>
<thead>
<tr>
<th>Modulation</th>
<th>$x(t) = A(t)\cos(\psi(t) + \theta)$</th>
<th>$A(t)$</th>
<th>$\psi(t)$</th>
<th>$K_{\text{min}} = \text{block}[n, m]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM</td>
<td>$a + b\sin(\omega_s t)$</td>
<td>$\omega_s t$</td>
<td></td>
<td>$[1, 1]$</td>
</tr>
<tr>
<td>FM</td>
<td>$a$</td>
<td>$\int_0^t \omega_c + c\cos(\omega_s t) , dt$</td>
<td>$[1, m]$, $m \geq c/\omega_s$</td>
<td></td>
</tr>
<tr>
<td>PM</td>
<td>$a$</td>
<td>$\omega_c t + d\sin(\omega_s t)$</td>
<td>$[1, m]$, $m \geq d$</td>
<td></td>
</tr>
</tbody>
</table>

In the last column we have added the minimum rectangular subset of $K$ in order to avoid obvious errors due to aliasing (assuming $\lambda_1 = \omega_c$, $\lambda_2 = \omega_s$). In general $\omega_c \gg \omega_s$. The following result, of which the proof is elementary, will be used in the sequel.

**THEOREM 1**

SWAM, SWFM and SWPM sources have a Fourier expansion with respect to the exponentials $e^{i(\eta \omega_c + \mu \omega_s)t}$. For SWAM and SWPM the coefficients are independent of $\omega_c$ and $\omega_s$. For SWFM the coefficients depend on $c/\omega_s$.

**REMARK**

A more careful evaluation of the coefficients reveals that the only non-zero harmonics are for $(k_1, k_2) = (1, m)$. For SWAM $m$ is also restricted to $m \leq 1$, showing that a finite expansion is obtained. For SWFM and SWPM the dominant part of the infinite expansions depends on $c/\omega_s$ and $d$, respectively.

### 5 DISCRETE FOURIER TRANSFORM

Let $\mathcal{F}_{\eta,\mu}$ denote the Fourier transform using fundamental frequencies $\eta, \mu$. We observe that in general, by the non-linearity of $i$ and $q$, $I(V)$ and $Q(V)$ depend on the specific $\lambda_1, \lambda_2$ mentioned previously. However, in practice, $I(V)$ and $Q(V)$ appear to be rather
Non-linear resistors are algebraic in the above sense when using the variables for all $\lambda$, $q$, $I$ coefficients at the other frequencies, that maps to satisfy the additional assumptions from Section 5. i.e. there is a bijective mapping \((10)\) on the larger set $K$ in the frequency domain is determined by the smaller set where $O$.

It allows to the determine the frequency-domain matrix contributions from their time-domain equivalents in process [8]. It allows to the efficient evaluation of $V$ in an efficient way. Writing $\lambda = \alpha_1 \lambda$ and $\lambda = \alpha_2 \lambda$, a frequency $\omega_{k}$ corresponding to $(k_1, k_2)$ (using $\lambda_1, \lambda_2$) becomes related to $\bar{k}\lambda$, where

$$\bar{k} \equiv k_1 \alpha_1 + k_2 \alpha_2.$$ (10)

For several sets $\Lambda$ of practical interest the $\alpha_1, \alpha_2$ and $\lambda$ can be chosen such that the $\bar{k}$ are integer. We restrict ourselves to the cases where the ‘$k$-mapping’ $(k_1, k_2) \mapsto \bar{k}$ is bijective.

Thus, the set $K$ of $(k_1, k_2)$ is bijectively mapped in a set $\bar{K}$ that corresponds to a one dimensional spectrum. For numerical efficiency reasons, we will further restrict ourselves to those cases where the resulting set of $\bar{K}$ shows no gaps (but this assumption may be relaxed), which is related to the spectrum for a periodic quantity. The choice of $\lambda_3, \lambda_4$ only serves the efficient evaluation of $I(V)$, by utilizing (8)-(9).

For sets $K$ that are rectangular, the requirements are found to be easily satisfied. We observe that for many problems, a spectrum with a diamond shape of $K$ is sufficient to analyse (5). Clearly, a diamond generates less unknowns than a rectangular set with the same bounds along the principal axes and the corresponding problem may be solved much faster for the diamond case than for the block case. However, when dealing with a diamond, we have to restrict ourselves to diamonds $K$, where the largest of the bounds is an integral multiple of the smallest (in which case the mapping in (10) is not bijective). The practical consequence is immediately clear. In all evaluations like $I(V)$, and of derivatives that are required by the non-linear solver, the Discrete Fourier Transform (DFT) can be exploited. Numerically, the DFT is very stable. In addition, it offers a fast transform property, based on a Toeplitz-Hankel block matrix structure, that is utilised to optimise the matrix assembly process [8]. It allows to the determine the frequency-domain matrix contributions from their time-domain equivalents in $O(K^2)$ operations rather than in $O(K^3)$.

### 5.1 OVERSAMPLING

Oversampling offers several enhancements. Firstly, it can be used to decrease effects due to aliasing. Secondly, it can be used to reduce the number of equations. And finally it allows the use of spectra that do not satisfy the assumptions imposed by the DFT.

When dealing with oversampling, we use two $K$ sets, the original $K$ and a new one $K_{DFT}$, such that $K \subset K_{DFT}$. Both associated spectra use the same set $\Lambda$ and satisfy the assumptions in Section 2, i.e. all frequencies should be different from each other. Only $K_{DFT}$ has to satisfy the additional assumptions from Section 5, i.e. there is a bijective mapping (10) that maps $K_{DFT}$ to a $\bar{K}$ for a one-dimensional periodic spectrum. The number of equations in the frequency domain is determined by the smaller set $K$. The DFT, however, is based on the larger set $K_{DFT}$. If we denote the Fourier coefficients for the set $K_{DFT}$ by $[V, V']$, where $V$ are the Fourier coefficients at frequencies that are associated with $K$ and $V'$ the coefficients at the other frequencies, $I(V)$ is evaluated by

$$I(V) = \theta(F_i(\mathcal{F}^{-1}\Theta(V))),$$ (11)
where the operators $\theta$ and $\Theta$ are defined by

\begin{align*}
\theta([V, V']) &= V, \\
\Theta(V) &= [V, 0].
\end{align*}

We observe that, especially when dealing with 3-D spectra, oversampling offers an interesting opportunity to balance the size of the spectrum against the size of the circuit to be studied. Also, the case of the 3-D diamond spectrum is interesting because it violates in nearly all cases the restrictions imposed by the DFT. However, efficient enveloping spectra can be found that satisfy these assumptions.

6 HIERARCHICAL ALGORITHM

6.1 NON-LINEAR SOLVER

Very often, the design of an electronic circuit is characterised by a hierarchical organisation of models with final leaves of passive and active components. In many cases, special composites of these are collected in devices (e.g. transistors) that are treated as building blocks in the modular design of the circuit. Because these devices have been characterised before, it is natural to consider them also as subblocks or leaves in a numerical algorithm. Models and devices are linked in the hierarchy by their terminal unknowns to the enclosing model. The top model is the circuit-level and has only one terminal, the fixed voltage at the ground node. The behaviour of the solution of a model is completely defined by the boundary values at its terminals (together with the internal sources). A hierarchically organised algorithm allows for a description of the circuit for a simulator that is very close to the original design.

We note that a hierarchical formulation corresponds to a particular block partitioning of the problem. Due to the linearity of the $\mathcal{F}$-transform, the Harmonic Balance algorithm, based on Newton-Raphson, can be formulated hierarchically. A form that fits such a structure is

\begin{align*}
Y^{(n)}V^{n+1} &= -J_{\text{Offset}}(V^n), \\
Y^{(n)} &= \frac{\partial I}{\partial V}(V^n) + \Omega \frac{\partial Q}{\partial V}(V^n), \\
J_{\text{Offset}}(V^n) &= J(V^n)S - Y^{(n)}V^n.
\end{align*}

The assembly of the functional matrix can be done hierarchically (bottom-up). In essence, only the non-trivial submatrices are treated. Especially when a hierarchical branch can be determined to be linear, the matrix contributions need to be assembled only once. If, in addition, no sources occur, the complete hierarchical branch will be treated only once because the contribution to $J_{\text{Offset}}$ will be zero.

6.2 LINEAR SOLVER

The hierarchical procedure also requires a linear solver for solving (14) that treats this structure efficiently.

6.2.1 GAUSSIAN ELIMINATION

Gaussian elimination can deal with a hierarchical structure in a very natural way when using a block partitioning on each level between internal unknowns and terminal unknowns. Matrix assembly and (partial) Gaussian decomposition can be combined in a (bottom-up) recursive procedure, starting at the leaf models and devices. Each resulting block that deals with the rows and columns of the terminal unknowns is inserted in the matrix contribution of the enclosing model on the next higher level.

The remaining part of the solution process (solving the triangular system) is performed in
the reverse order (top-down). The ground-unknowns provide the gauge of the problem on the top-level, where they serve as interface unknowns. The internal unknowns on this level are evaluated after this. Several unknowns serve as interface unknowns on the next sub-level, and so on.

Here also a mix with phases of the non-linear solver is possible. When the boundary values of a sub-model or device are considered to be converged, one may decide to by-pass a complete hierarchical branch. The solution can also be combined with damping or relaxation with Newton-Raphson by applying such processes on each level in the hierarchy before going further on in the hierarchy.

The matrices may be defined using data-structures for either full or sparse matrices. When dealing with sparse matrices, no pivoting is allowed because this may destroy the sparse matrix structure. Because of this, we replace a small or zero-valued diagonal element by some minimum pivot value and solve the system in this way, which results in an approximation $x_0$ to the solution $x$. This approximation is improved iteratively by applying defect correction on the current level before going to the next hierarchical level. By a further block-treatment in which a current unknown and a voltage unknown are grouped together we circumvent many cases for possible zero-pivots.

The hierarchical formulation implicitly offers a rough sparse matrix structure that reduces the storage of full matrices when compared to a completely flat description. This is especially appreciated in a Harmonic Balance analysis where the matrices are much bigger than in other analyses (such as transient analysis). By the hierarchical approach, the solution phase treats lists of systems in a natural way, where for each system, the matrix is much smaller than for a flat description.

### 6.2.2 Iterative Methods

Several popular iterative methods for solving linear systems arising from partial differential equations require a suitable preconditioning, which does not cope easily with a hierarchical structure. In addition, the intermediate linear systems may have non-symmetric coefficient matrices. In many cases, well-known iterative methods (CGS, Bi-CG, Bi-CGSTAB, ILQ with CG) fail to converge. An in-house developed hierarchical preconditioned CGS proved to be most robust, but was only competitive to ordinary Gaussian elimination for circuits with very large submodels in the hierarchy.

In [3, 12] iterative methods are described that were successful when applied to the linear system of equation arising in Harmonic Balance. However, the performance of these methods, when applied to hierarchically organized matrices, has still to be considered more closely.

### 6.3 Starting Newton-Raphson

#### 6.3.1 Direct Start

The Newton-Raphson process is started with the DC-solution, i.e. the steady-state solution with all time-dependent sources (sine-wave sources, SWAM, etc.) set to 0. The DC-solution $v_{DC}$ satisfies the reduced equation (2). Eq. (2) may also be non-linear. Hence, in general, it requires an additional Newton-Raphson process to solve it. Couplings due to capacitors may give rise to floating areas. In order to gauge the solution in such cases, in solving (2) capacitors are replaced by small (linear) conductors. Within a procedure where these artificial conductors are iteratively decreased to some minimum value, the non-linear problems are solved sequentially, each next problem being initialised with the solution of the previous one.

The DC-solution immediately provides the frequency-domain equivalent $V^0$, with each $V^j_k$ given by

$$ V^j_0 = v^j_{DC}, $$

$$ V^j_k = 0, \quad k = 1, \ldots, K. $$
An efficient improvement is obtained by generating non-trivial $V_j^k$ ($k = 1, \ldots, K$) by solving a set of AC-problems in parallel, each being linearised around the same DC-solution. Because each AC-problem is linear this is very efficient. This saves one iteration with Newton-Raphson, starting with (17)-(18). Note that this approach may be interpreted as the first iteration of a non-linear block Gauss-Jacobi approach, using partitions between the components of different harmonics.

6.3.2 CONTINUATION

In many cases, users are interested in series of harmonic balance analyses, for instance by varying the fundamental frequencies in modified sources, or by varying the amplitudes of the sources. In such frequency or amplitude sweeps, it is efficient to be able to step easily from one problem to the next. In order to meet this requirement, a continuation procedure has been built in which each next non-linear problem is initialised with the solution of the previous problem. When one of the fundamental frequencies is varied (from $f$ to $\tilde{f}$, say) a simple expansion of (5) shows that an improved first iterand for the next problem at $\tilde{f}$ can be obtained quite efficiently from the one at $f$ by solving

$$Y(f) \delta V = S(\tilde{f}) - S(f) - (\tilde{f} - f) \Omega'(f) Q(V(f)), \quad (19)$$

in which $Y(f)$ is the last Jacobian matrix of the converged problem at $f$, of which the decomposition is re-used, $\delta V = V(\tilde{f}) - V(f)$ and $\Omega'(f) = d\Omega/df$ (which is a constant block diagonal matrix). Note that when the Fourier coefficients of $S$ do not depend on the frequencies, the first two terms in (19) cancel, in which case the solution of (19) is simply a postprocessing result of the converged problem at $f$.

It is clear that a continuation procedure also enhances the robustness of the algorithm. We note that the expected number of Newton-Raphson iterations may also be estimated from those that were necessary for previous problems. Locally, a higher number of Newton-Raphson iterations may be necessary than in the general part of the parameter sweep. This occurs, for instance, near frequency values where oscillation occurs. An adaptive estimation may assist in completing the current problem and to step over the discontinuity to a next branch where again, a smooth convergence behaviour is met.

6.4 DAMPING AND STOPPING

In order to avoid unwanted sudden changes in the solution generated by the Newton-Raphson process, we limit $V^{(n+1)}$ on each level in the hierarchy by

$$[V^j_{k}]^{(n+1)} := [V^j_{k}]^{(n)} + \lambda_{\text{Model}}([V^j_{k}]^{(n+1)} - [V^j_{k}]^{(n)}), \quad (20)$$

where the damping-parameter $\lambda_{\text{Model}} \in [0, 1]$ is uniformly in $k$ and per model and assures that

$$||[V^j_{k}]^{(n+1)} - [V^j_{k}]^{(n)}|| \leq \tau_{\text{Rel}}||[V^j_{k}]^{(n)}|| + \tau_{\text{Abs}}. \quad (21)$$

Here $\tau_{\text{Rel}}$ and $\tau_{\text{Abs}}$ are global constants that only depend on the type of unknown.

This limiting procedure is applied on each level in the hierarchy, before going further on in the hierarchy. Thus, the procedure also immediately affects the solution of the system on the next levels, where further limiting may be necessary. When Newton converges, each $\lambda_{\text{Model}} \rightarrow 1$.

The stopping criterion is based on a combination of absolute and relative error estimations for the corrections (in the $\infty$-norm) and is applied before the limiting procedure, the tolerance parameters being dependent on the type of unknown (voltage, current, charge or flux). When a hierarchical branch has converged previously and the corrections at unknowns at the terminals remain small for the next iterands, the branch may be temporarily bypassed in the assembly procedure.

We note that the above criterion treats all harmonics in the same way and allows for achieving the same accuracy for all harmonics (ignoring effects due to aliasing).
The BFR92A is a discrete NPN transistor in a plastic SOT23 package. It is primarily intended for use in RF wideband amplifiers and oscillators. This transistor has a high power gain, a low noise figure and low intermodulation distortion. This intermodulation distortion specification can be simulated well with the distortion analysis provided by the Harmonic Balance Algorithm. The test circuit is shown in Fig. 1.

The BFR92A specification sheet [10] contains data for comparing the simulated intermodulation distortion with measured results. In this example, we have a sign wave at $E_2$ with three independent fundamental frequencies $f_1 = 795.25\,\text{MHz}$, $f_2 = 803.25\,\text{MHz}$, and $f_3 = 805.25\,\text{MHz}$ and associated amplitudes $A, \frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}$, respectively, where $A = 0.150\sqrt{2}/1.24$ (and all phases equal to 0). The measured relative distortion value $d_{\text{im}}$ equals at the specific node $j = \text{out}$, the value $\text{dB}(\frac{|V_k|}{V_k^{'}})$, where $k'$ indicates the index of the Fourier coefficient at frequency 793.25MHz ($= f_1 + f_2 - f_3$), and $k$ the index at $f_1$ (Note: $\text{dB}(x) = 20 \log_{10}(x)$).

A 3-D diamond analysis spectrum comprises 63 components, but this spectrum does not satisfy the assumptions necessary for the DFT. The enveloping block[3,3,3] does, but it generates 343 components and the resulting system of equations is too large to simulate. Here, we applied oversampling on the 3-D diamond spectrum using an efficient enveloping spectrum that satisfies the DFT-restrictions. Thus, the size of the non-linear equations is based on the diamond spectrum, whereas the DFT is based on the enveloping spectrum. The value of $d_{\text{im}}$, obtained by the simulation, $-6.002\times10^1$, agrees very well with the value as specified in the BRF92A datasheet specification ($d_{\text{im}} = -60\,\text{dB}$).

Fig. 2 shows the result of the time-profile of the voltage at node ‘out’.

8 POLE-ZERO ANALYSIS

In contrast to the equations solved by Harmonic Balance, Pole-Zero Analysis only applies to linearized equations in the frequency-domain. In Pole-Zero Analysis one has to solve generalized eigenvalue problems for poles and zeroes. Here the poles allow stability analysis of the small signal behaviour. The circuit transfer function of the small signal equations can be described completely in terms of poles and zeroes.
8.1 BASIC BACK-GROUND

The approach for Pole-Zero Analysis is well-known [1]. For convenience, we recapitulate some basic results.

Figure 2: Time-profile of the voltage at the point of measurement.

The small signal equations deal with \( x(t) \equiv v(t) - v_{DC} \) and neglecting higher order terms, \( x(t) \) satisfies the linearised equations

\[
Gx + C \frac{dx}{dt} = 0,
\]

\( G = \frac{\partial i}{\partial v_{DC}} \), \( C = \frac{\partial q}{\partial v_{DC}} \). (22)

After applying the Laplace Transform, the frequency-domain equivalent becomes

\[
[G + sC]X = Cx(0),
\]

(23)

where \( X \) is the Laplace Transform of \( x \) and \( s \in \mathbb{C} \).

The matrix \( \mathcal{H}(s) \) defined by

\[
\mathcal{H}(s) = [G + sC]^{-1},
\]

(24)

describes the circuit response transfer function. We note that the entry \( \mathcal{H}_{ij}(s) \), being the response of the \( i \)-th output unknown to a unit variation of the \( j \)-th unknown, satisfies (Cramer’s rule)

\[
\mathcal{H}_{ij}(s) = \frac{\text{Det}(\mathcal{A}[G + sC, i, j])}{\text{Det}(G + sC)},
\]

(25)

where \( \mathcal{A}[A, i, j] \) is constructed from a matrix \( A \) by replacing the \( i \)-th column by the \( j \)-th unit vector \( e_j \). Hence, \( \mathcal{H}_{ij}(s) \) is a rational function in \( s \) that can be written as

\[
\mathcal{H}_{ij}(s) = h_{ij} \frac{\prod_k (s - z_k)}{\prod_k (s - p_k)},
\]

(26)

for a suitable factor \( h_{ij} \). Here, the poles \( p_k \) and zeroes \( z_k \) are generalised eigenvalues of

\[
(G + pC)x = 0, \quad x \neq 0,
\]

(27)

\[
(\mathcal{A}[G + zC, i, j])\bar{x} = 0, \quad \bar{x} \neq 0.
\]

(28)

Note that \( h_{ij} \) can be determined from

\[
h_{ij} \frac{\prod_k (-z_k)}{\prod_k (-p_k)} = \mathcal{H}_{ij}(0) = (G^{-1})_{ij}.
\]

(29)
For real $s$, the terms $\mathcal{H}_{ij}(s)$ and $h_{ij}$ and the products $\prod_k(-z_k)$ and $\prod_\ell(-p_\ell)$ are real. The location of the poles determines the stability of the circuit. In several applications, one is interested in the variation of poles and zeroes as a function of some parameter [5].

8.2 BODE PLOTS

The transfer response described by (26) is valid for all complex $s$ (for which the small signal assumption is valid). For $s = 2\pi f$ the same response can be determined by performing a frequency swept series of AC-analyses (and, in general, in a much cheaper way). In this case, the graph of the modulus of the resulting transfer response (in dB), as a function of $f$, results in a familiar Bode plot [1]. By comparing the AC-Bode plot to the PZ-Bode plot, one can check the accuracy of the poles and zeroes.

In general, in a transfer response, cancellation of poles and zeroes will occur. Cancellation of nearest (complex) poles $p$ and zeroes $z$ is governed by the rule

$$|p - z| \leq \varepsilon_{\text{Rel}} \frac{|\text{Re}(p) + \text{Re}(z)|}{2} + \varepsilon_{\text{Abs}}$$  \hspace{1cm} (30)

(assuming imaginary parts of equal sign), with the restriction that real poles can only cancel to real zeroes. A pair of conjugate poles can only be canceled by a pair of conjugate zeroes. In (30), the relative criterion, says that the distance between $p$ and $z$ is small compared to the distance of $p$ and $z$ to the imaginary axis. This keeps the error in the transfer function $\mathcal{H}_{ij}(s)$ small for $s = 2\pi f$.

We note that the PVL-algorithm [5] circumvents the cancellation of the poles and zeroes occurring in the transfer function automatically. However this algorithm is less suited for stability analysis than the algorithms in the next sections.

8.3 QR AND ARNOLDI EIGENVALUE ALGORITHMS

For a stability analysis one is interested in finding all poles, while for a particular response one may be interested in a selection only (see again [5]). We currently implemented QR as well as Arnoldi for solving the generalised eigenvalue problems [11]. QR is implemented for flat matrices, i.e. for matrices that are not hierarchically organised. The method, of course, is very accurate and stable and one finds all poles and zeroes. For QR, however, the circuit is limited, say, to have at most 600 unknowns.

For this reason, a basic Arnoldi was also implemented. This allows for larger circuits, because it can deal with the hierarchical block matrix structure, which is possible because it only uses products of a matrix with a vector. Arnoldi is an iterative Krylov subspace method. Limiting the size of the Krylov subspaces implies that only a limited number of poles and zeroes may be approximated. The Arnoldi algorithm proved to be less robust than QR and also less accurate. The problems were due to scaling, loss of orthogonality and occurrence of Jordan blocks, etc. To improve the algorithm, enhancements are needed.

8.4 JACOBI-DAVIDSON QR (JDQR)

Recently, the Jacobi-Davidson eigenvalue algorithm was developed at Utrecht University [13, 14]. It was originally meant as a method for only a few (generalised) eigenvalues. It had very interesting convergence properties. The question was whether it could be suitable for our industrial application. A first test with the available original Matlab implementation of the QR variant was rather disappointing: it was no better (robust or accurate) than Arnoldi for our purposes. Consequently, in [2] the problem was studied more closely. Several modifications improved the algorithm. A more general remark (that also applies to other algorithms) is that the sparsity pattern of $G$ and $C$ allows the definition of a restriction matrix $\mathcal{S}$ and it suffices to consider only the eigenvalues of $\mathcal{S}^T G^{-1} C \mathcal{S}$. This reduces the problemsize, but this may also eliminate several Jordan blocks (that otherwise could give rise to ill-conditioned eigenvalues). The resulting algorithm is now much more robust and accurate.
than Arnoldi. However, things still have to be worked out further. The computational costs strongly depend on the number of eigenvalues that are important to the specified frequency range. If this number is large, JDQR will be more expensive than QR. In contrast, JDQR needs less memory than QR. A shift-and-invert approach within JDQR can be interesting for very large circuit problems.

8.5 EXAMPLE

Fig. 3 [2] shows the AC and the PZ-Bode plots for a circuit for which Arnoldi failed. The original size was 504 unknowns, after size reduction 365 remained. With JDQR 299 poles and 300 zeroes were determined.

References


