Chapter 6
Advanced Topics in Model Order Reduction

Davit Harutyunyan, Roxana Ionutiu, E. Jan W. ter Maten, Joost Rommes, Wil H.A. Schilders, and Michael Striebel

Abstract This chapter contains three advanced topics in model order reduction (MOR): nonlinear MOR, MOR for multi-terminals (or multi-ports) and finally an application in deriving a nonlinear macromodel covering phase shift when coupling oscillators. The sections are offered in a preferred order for reading, but can be read independently.

Section 6.1, written by Michael Striebel and E. Jan W. ter Maten, deals with MOR for nonlinear problems. Well-known methods like TPWL (Trajectory PieceWise Linear) and POD (Proper Orthogonal Decomposition) are presented. Development for POD led to some extensions: Missing Point Estimation, Adapted POD, DEIM (Discrete Empirical Interpolation Method).

D. Harutyunyan (✉)
ASML, De Run 6501, 5504 DR Veldhoven, The Netherlands
e-mail: Davit.Harutyunyan@asml.com.

R. Ionutiu
ATTE, ABB Switzerland Ltd, Austraße, 5300 Turgi, Switzerland
e-mail: Roxana.Ionutiu@ch.abb.com

E.J.W. ter Maten
Chair of Applied Mathematics/Numerical Analysis, Bergische Universität Wuppertal, Gaußstraße 20, D-42119 Wuppertal, Germany

Department of Mathematics and Computer Science, CASA, Eindhoven University of Technology, P.O.Box 513, 5600 Eindhoven, The Netherlands
e-mail: Jan.ter.Maten@math.uni-wuppertal.de; E.J.W.ter.Maten@tue.nl

W.H.A. Schilders
Department of Mathematics and Computer Science, CASA, Eindhoven University of Technology, P.O.Box 513, 5600 Eindhoven, the Netherlands
e-mail: W.H.A.Schilders@tue.nl

J. Rommes
Mentor Graphics, DSM/AMS, Le Viseo – Bâtiment B, 110 rue Blaise Pacal, Inovalee, 38330 Montbonnot, France
e-mail: Joost_Rommes@mentor.com

M. Striebel
ZF Lenksysteme GmbH, Richard-Bullinger-Straße 77, D-73527 Schwäbisch Gmünd, Germany
e-mail: Michael.Striebel@zf-lenksysteme.com

M. Günther (ed.), *Coupled Multiscale Simulation and Optimization in Nanoelectronics*, Mathematics in Industry 21,
DOI 10.1007/978-3-662-46672-8_6

Section 6.2, written by Roxana Ionutiu and Joost Rommes, deals with the multi-terminal (or multi-port) problem. A crucial outcome of the research is that one should detect “important” internal unknowns, which one should not eliminate in order to keep a sparse reduced model. Such circuits come from verification problems, in which lots of parasitic elements are added to the original design. Analysis of effects due to parasitics is of vital importance during the design of large-scale integrated circuits, since it gives insight into how circuit performance is affected by undesired parasitic effects. Due to the increasing amount of interconnect and metal layers, parasitic extraction and simulation may become very time consuming or even unfeasible. Developments are presented, for reducing systems describing *R* and *RC* netlists resulting from parasitic extraction. The methods exploit tools from graph theory to improve sparsity preservation especially for circuits with multi-terminals. Circuit synthesis is applied after model reduction, and the resulting reduced netlists are tested with industrial circuit simulators. With the novel *RC* reduction method SparseMA, experiments show reduction of 95 % in the number of elements and 46x speed-up in simulation time.

Section 6.3, written by Davit Harutyunyan, Joost Rommes, E. Jan W. ter Maten and Wil H.A. Schilders, addresses the determination of phase shift when perturbing or coupling oscillators. It appears that for each oscillator the phase shift can be approximated by solving an additional scalar ordinary differential equation coupled to the main system of equations. This introduces a nonlinear coupling effect to the phase shift. That just one scalar evolution equation can describe this is a great outcome of Model Order Reduction. The motivation behind this example is described as follows. Design of integrated RF circuits requires detailed insight in the behavior of the used components. Unintended coupling and perturbation effects need to be accounted for before production, but full simulation of these effects can be expensive or infeasible. In this section we present a method to build nonlinear phase macromodels of voltage controlled oscillators. These models can be used to accurately predict the behavior of individual and mutually coupled oscillators under perturbation at a lower cost than full circuit simulations. The approach is illustrated by numerical experiments with realistic designs.

6.1 Model Order Reduction of Nonlinear Network Problems

The dynamics of an electrical circuit can in general be described by a nonlinear, first order, differential-algebraic equation (DAE) system of the form[1]:

$$\frac{d}{dt}\mathbf{q}(\mathbf{x}(t)) + \mathbf{j}(\mathbf{x}(t)) + \mathbf{B}\mathbf{u}(t) = \mathbf{0}, \tag{6.1a}$$

[1] Section 6.1 has been written by Michael Striebel and E. Jan W. ter Maten.

completed with the output mapping

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)). \tag{6.1b}$$

In the state equation (6.1a), which arises from applying modified nodel analysis (MNA) to the network graph, $\mathbf{x}(t) \in \mathbb{R}^n$ represents the unknown vector of circuit variables at time $t \in \mathbb{R}$; $\mathbf{q}, \mathbf{j} : \mathbb{R}^n \to \mathbb{R}^n$ describe the contribution of reactive and nonreactive elements, respectively and $\mathbf{B} \in \mathbb{R}^{n \times m}$ distributes the input excitation $\mathbf{u} : \mathbb{R} \to \mathbb{R}^m$. The system's response $\mathbf{y}(t) \in \mathbb{R}^p$ is a possibly nonlinear function $\mathbf{h} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^q$ of the system's state $\mathbf{x}(t)$ and inputs $\mathbf{u}(t)$.

In circuit design, (6.1a) is often not considered to describe the overall design but rather to be a model of a subcircuit or subblock. Connection to and communication with a block's environment is done via its terminals, i.e. external nodes. Therefore, we assume in the remainder of this document that the inputs $\mathbf{u}(t)$ and outputs $\mathbf{y}(t)$ denote terminal voltages and terminal currents, respectively, or vice versa, which are injected and extracted linearly, i.e., the output mapping is assumed to be of the form

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t), \tag{6.1c}$$

with $\mathbf{C} \in \mathbb{R}^{p \times n}$.

The dimension n of the unknown vector $\mathbf{x}(t)$ is of the order of the number of elements in the circuit, which can easily reach hundreds of millions. Therefore, one may solve the network equations (6.1a) and (6.1c) by means of computer algebra in an unreasonable amount of time only.

Model order reduction (MOR) aims to replace the original model (6.1a) and (6.1c) by a system

$$\begin{aligned} \frac{d}{dt}\hat{\mathbf{q}}(\mathbf{z}(t)) + \hat{\mathbf{j}}(\mathbf{z}(t)) + \hat{\mathbf{B}}\mathbf{u}(t) &= \mathbf{0}, \\ \hat{\mathbf{y}}(t) &= \hat{\mathbf{C}}\mathbf{z}(t), \end{aligned} \tag{6.2}$$

with $\mathbf{z}(t) \in \mathbb{R}^r$; $\hat{\mathbf{q}}, \hat{\mathbf{j}} : \mathbb{R}^r \to \mathbb{R}^r$ and $\hat{\mathbf{B}} \in \mathbb{R}^{r \times m}$ and $\hat{\mathbf{C}} \in \mathbb{R}^{p \times r}$, which can compute the system response $\hat{\mathbf{y}}(t) \in \mathbb{R}^p$ that is sufficiently close to $\mathbf{y}(t)$ given the same input signal $\mathbf{u}(t)$, but in much less time.

6.1.1 Linear Versus Nonlinear Model Order Reduction

So far most research effort was spent on developing and analysing MOR techniques suitable for linear problems. For an overview on these methods we refer to [1].

When trying to transfer approaches from linear MOR, fundamental differences emerge.

To see this, first consider a linear problem of the form

$$\begin{aligned} \mathbf{E}\frac{d}{dt}\mathbf{x}(t) + \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) &= \mathbf{0}, \quad \text{with } \mathbf{E}, \mathbf{A} \in \mathbb{R}^{n\times n}, \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t). \end{aligned} \tag{6.3}$$

The state $\mathbf{x}(t)$ is approximated in a lower dimensional space of dimension $r \ll n$, spanned by basis vectors which we subsume in $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_r) \in \mathbb{R}^{n\times r}$:

$$\mathbf{x}(t) \approx \mathbf{V}\mathbf{z}(t), \quad \text{with } \mathbf{z}(t) \in \mathbb{R}^r . \tag{6.4}$$

The reduced state $\mathbf{z}(t)$, i.e., the coefficients of the expansion in the reduced space, is defined by a reduced dynamical system. Applying Galerkin technique, this reduced system arises from projecting (6.3) on a test space spanned by the columns of some matrix $\mathbf{W} \in \mathbb{R}^{n\times r}$. There, $\mathbf{W}$ and $\mathbf{V}$ are chosen, such that their columns are biorthonormal, i.e., $\mathbf{W}^T\mathbf{V} = \mathbf{I}_{r\times r}$. The Galerkin projection[2] yields

$$\begin{aligned} \hat{\mathbf{E}}\frac{d}{dt}\mathbf{z}(t) + \hat{\mathbf{A}}\mathbf{z}(t) + \hat{\mathbf{B}}\mathbf{u}(t) &= \mathbf{0}, \\ \mathbf{y}(t) &= \hat{\mathbf{C}}\mathbf{z}(t) \end{aligned} \tag{6.5}$$

with $\hat{\mathbf{E}} = \mathbf{W}^T\mathbf{E}\mathbf{V}$, $\hat{\mathbf{A}} = \mathbf{W}^T\mathbf{A}\mathbf{V} \in \mathbb{R}^{r\times r}$ and $\hat{\mathbf{B}} = \mathbf{W}^T\mathbf{B} \in \mathbb{R}^{r\times m}$, $\hat{\mathbf{C}} = \mathbf{C}\mathbf{V} \in \mathbb{R}^{p\times r}$. The system matrices $\hat{\mathbf{E}}$, $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, $\hat{\mathbf{C}}$ of this reduced substitute model are of smaller dimension and constant, i.e., need to be computed only once. However, $\hat{\mathbf{E}}$, $\hat{\mathbf{A}}$ are usually dense whereas the system matrices $\mathbf{E}$ and $\mathbf{A}$ are usually very sparse.

Applying the same technique directly to the nonlinear system means obtaining the reduced formulation (6.2) by defining $\hat{\mathbf{q}}(\mathbf{z}) = \mathbf{W}^T\mathbf{q}(\mathbf{V}\mathbf{z})$ and $\hat{\mathbf{j}}(\mathbf{z}) = \mathbf{W}^T\mathbf{j}(\mathbf{V}\mathbf{z})$. Clearly, $\hat{\mathbf{q}}$ and $\hat{\mathbf{j}}$ map from $\mathbb{R}^r$ to $\mathbb{R}^r$.

To solve network problems of type (6.2) numerically, usually multistep methods are used. This means that at each timepoint t_l a nonlinear equation

$$\alpha\hat{\mathbf{q}}(\mathbf{z}_l) + \hat{\boldsymbol{\beta}}_l + \hat{\mathbf{j}}(\mathbf{z}_l) + \hat{\mathbf{B}}\mathbf{u}(t_l) = \mathbf{0} \tag{6.6}$$

has to be solved for $\mathbf{z}_l$ which is the approximation of $\mathbf{z}(t_l)$. In the above equation α is the integration coefficient of the method and $\hat{\boldsymbol{\beta}}_l \in \mathbb{R}^r$ contains history from previous timesteps. Newton techniques that are used to solve (6.6) usually require an update of the system's Jacobian matrix in each iterations ν:

$$\hat{\mathbf{J}}_l^{(\nu)} = \left(\alpha\frac{\partial\hat{\mathbf{q}}}{\partial\mathbf{z}} + \frac{\partial\hat{\mathbf{j}}}{\partial\mathbf{z}}\right)\Big|_{\mathbf{z}=\mathbf{z}_l^{(\nu)}} = \mathbf{W}^T\left[\alpha\frac{\partial\mathbf{q}}{\partial\mathbf{x}} + \frac{\partial\mathbf{j}}{\partial\mathbf{x}}\right]\Big|_{\mathbf{x}^{(\nu)}=\mathbf{V}\mathbf{z}_l^{(\nu)}}\mathbf{V}. \tag{6.7}$$

[2] Most frequently $\mathbf{V}$ is constructed to be orthogonal, such that $\mathbf{W} = \mathbf{V}$ can be chosen.

The evaluation of the reduced system, i.e., $\hat{\mathbf{q}}$ and $\hat{\mathbf{j}}$, necessitates in each step the back projection of the argument $\mathbf{z}$ to its counterpart $\mathbf{Vz}$ followed by the evaluation of the full system $\mathbf{q}$ and $\mathbf{j}$ and the projection to the reduced space with $\mathbf{W}$ and $\mathbf{V}$.

Consequently, with respect to computation time no reduction will be obtained unless additional measures are taken or other strategies are pursued.

6.1.2 Some Nonlinear MOR Techniques

In MOR for linear systems especially methods based on Krylov subspaces [19] and balanced realization [30] are well understood and highly elaborated. Hence, it seems likely to adapt them to nonlinear problems, too. In the following, we shortly describe these approaches and give references for further reading.

6.1.2.1 Krylov Subspace Methods in Nonlinear MOR

In linear MOR Krylov subspace methods are used to construct reduced order models of systems (6.3) such that the moments, i.e., the coefficients in a Taylor expansion of the frequency domain transfer function of original and reduced system match up to a certain order. The transfer function $\mathbf{H} : \mathbb{C} \to \mathbb{C}^{p\times m}$ is defined by the linear equation $\mathbf{H}(s) = \mathbf{C}(s\mathbf{E} + \mathbf{A})^{-1}\mathbf{B}$.

It is not straightforward to define a transfer function for the nonlinear problem (6.1a) and (6.1c). Instead, there are Krylov based techniques that deal with bilinear systems (6.8) or linear periodically time varying (LPTV) problems (6.9).

$$\text{bilinear system:} \qquad \frac{d}{dt}\hat{\mathbf{x}}(t) + \hat{\mathbf{A}}\hat{\mathbf{x}}(t) + \hat{\mathbf{N}}\hat{\mathbf{x}}(t)\mathbf{u}(t) + \hat{\mathbf{B}}\mathbf{u}(t) = \mathbf{0} \tag{6.8}$$

$$\text{LPTV system:} \qquad \frac{d}{dt}[\mathbf{E}(t)\mathbf{x}(t)] + \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) = \mathbf{0} \tag{6.9}$$

The type of problem (6.8) arises from expanding a nonlinear problem $\dot{\mathbf{x}}(t) + \mathbf{f}(\mathbf{x}(t)) + \mathbf{B}\mathbf{u}(t) = \mathbf{0}$ around an equilibrium point. Systems of type (6.9) with matrices $\mathbf{E}(t), \mathbf{A}(t)$ that are periodic with some period T one gets when linearising the system (6.1) around a periodic steady state solution with $\mathbf{x}^0(t + T) = \mathbf{x}^0(t)$.

Volterra-series expansion, followed by multivariable Laplace-transformation and multimoment expansions are the key to apply Krylov subspace based MOR. For further reading we refer to [15] and the references therein.

In case of the LPTV systems, a timevarying system function $\mathbf{H}(s, t)$ can be defined. This plays the role of a transfer function and can be determined by a differential equation. $\mathbf{H}(s, t)$ has to be determined in terms of time- or frequency samples on $[0, T)$ for one s. Krylov techniques can then be applied to get a reduced system with which samples for different frequencies s can be constructed. We refer to [21] and the references therein.

Given a nonlinear problem (6.1a) and (6.1c) of dimension n, the bilinear system that is reduced actually has a dimension of $n + n^2 + n^3 + \cdots$, depending on the order of the expansion. Similar, the system in the LPTV case that is subject to reduction has dimension $k \cdot n$ with k being the number of timesamples in the initial determination of $\mathbf{H}(s, t)$. Therefore, it seems that these methods are suitable for small to medium sized nonlinear problems only.

6.1.2.2 Balanced Truncation in Nonlinear MOR

The energy $L_c(\mathbf{x}_0)$ that is needed to drive a system to a given state $\mathbf{x}_0$ and the energy $L_o(\mathbf{x}_0)$ the system provides to observe the state $\mathbf{x}_0$ it is in are the main terms in Balanced Truncation. A system is called balanced if states that are hard to reach are also hard to observe and vice versa, i.e. $L_c(\mathbf{x})$ large implies $L_o(\mathbf{x})$. Truncation, i.e. reduced order modelling is then done by eliminating these states.

For linear problems L_c and L_o are connected directly, by means of algebraic calculation, to the reachability and observability Gramians $\mathbf{P}$ and $\mathbf{Q}$, respectively. These can be computed from Lyapunov equations, involving the system matrices $\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C}$ of the linear system (6.3). Balancing is reached by transforming the state space such that $\mathbf{P}$ and $\mathbf{Q}$ are simultaneously diagonalised:

$$\mathbf{P} = \mathbf{Q} = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$$

with the so called Hankel singular values $\sigma_1, \ldots, \sigma_n$. From the basis that arises from the transformation only those basis vectors that correspond to large Hankel singular values are kept. The main advantage of this approach is that there exists an a priori computable error bound for the truncated system.

In transferring Balanced Truncation to nonlinear problems, three main tracks can be recognized. Energy consideration is the common ground for the three directions.

In the approach suggested in [20] the energy functions arise from solving Hamilton-Jacobi differential equations. Similar to the linear case, a state-space transformation is searched such that L_c and L_o are formulated as quadratic form with diagonal matrix. The magnitude of the entries are then basis to truncation again. The transformation is now state dependent, and instead of singular values, we get singular value functions. As the Hamilton-Jacobi system has to be solved and the varying state-space transformations have to be computed, it is an open issue, how the theory could be applied in a computer environment.

In Sliding Interval Balancing [46], the nonlinear problem is first linearised around a nominal trajectory, giving a linear time varying system like (6.9). At each state finite time reachability and observability Gramians are defined and approximated by truncated Taylor series expansion. Analytic calculations, basically the series expansions, connect the local balancing transformation smoothly. This necessary step is the limiting factor for this approach in circuit simulation.

Finally, balancing is also applied to bilinear systems (6.8). Here the key tool are so called algebraic Gramians arising from generalised Lyapunov equations. However, no one-to-one connection between these Gramians and the energy functions L_c, L_o can be made, but rather they can serve to get approximative bounds for the aforementioned. Furthermore, convergence parameters have to be introduced to guarantee the solvability of the generalised Lyapunov equations. For further details we refer to [9, 14] and the references therein.

6.1.3 TPWL and POD

In view of high dimensional problems in circuit simulation and feasibility in a computational environment, Trajectory PieceWise Linearization (TPWL) and Proper Orthogonal Decomposition (POD) are amongst the most promising approaches for the time being. The basic idea of TPWL is to replace nonlinearity with a collection of linear substitute problems and apply MOR on these. The background of POD is to identify a low dimensional manifold the solution resides on and reformulate the problem in such a way that it is solved in terms of the basis of this principal manifold.

In the following we give more details on the steps done for both approaches.

6.1.3.1 Trajectory PieceWise Linearization

The idea of TPWL [33], is to represent the full nonlinear system (6.1a) and (6.1c) by a set of order reduced linear models that can reproduce the typical behaviour of the system.

Since its introduction in [33, 34], TPWL has gained a lot of interest and several adaptions have been made, see e.g., [18, 39, 49]. In the following we will basically follow the lines in the original works [33, 34] and briefly mention alternatives that have been suggested.

For extracting a model, a training input $\bar{\mathbf{u}}(t)$ for $t \in [t_{\text{start}}, t_{\text{end}}]$ is chosen and a transient simulation is run in order to get a trajectory, i.e. a collection of points $\mathbf{x}_0, \ldots, \mathbf{x}_N$, approximating $\mathbf{x}(t_i)$ at timepoints $t_{\text{start}} = t_0 < t_1 < \cdots < t_N = t_{\text{end}}$. The training input is chosen such that the trajectory it causes, reflects the typical state of the system. On the trajectory, points $\{\mathbf{x}_0^{\text{lin}}, \ldots, \mathbf{x}_s^{\text{lin}}\} \subset \{\mathbf{x}_0, \ldots, \mathbf{x}_N\}$ are chosen around which the nonlinear functions $\mathbf{q}$ and $\mathbf{j}$ are linearised:

$$\mathbf{q}(\mathbf{x}(t)) \approx \mathbf{q}(\mathbf{x}_i^{\text{lin}}) + \mathbf{E}_i \cdot \left(\mathbf{x}(t) - \mathbf{x}_i^{\text{lin}}\right); \quad \mathbf{j}(\mathbf{x}(t)) \approx \mathbf{j}(\mathbf{x}_i^{\text{lin}}) + \mathbf{A}_i \cdot \left(\mathbf{x}(t) - \mathbf{x}_i^{\text{lin}}\right), \tag{6.10}$$

with $\mathbf{E}_i = \frac{\partial \mathbf{q}}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_i^{\text{lin}}}$ and $\mathbf{A}_i = \frac{\partial \mathbf{j}}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_i^{\text{lin}}}$.

Then the nonlinear state-space equation (6.1a) can locally be replaced locally around $\mathbf{x}_i^{\text{lin}}$ for $i = 1, \ldots, s$ by

$$\frac{d}{dt}\left[\mathbf{E}_i\mathbf{x}(\mathbf{t}) + \boldsymbol{\delta}_i\right] + \mathbf{A}_i\mathbf{x}(t) + \boldsymbol{\gamma}_i + \mathbf{Bu}(t) = \mathbf{0}, \tag{6.11}$$

with $\boldsymbol{\delta}_i = \mathbf{q}(\mathbf{x}_i^{\text{lin}}) - \mathbf{E}_i\mathbf{x}_i^{\text{lin}}$ and $\boldsymbol{\gamma}_i = \mathbf{j}(\mathbf{x}_i^{\text{lin}}) - \mathbf{A}_i\mathbf{x}_i^{\text{lin}}$.

One approach, used by Rewieński [33], to get a model that represents the nonlinear problem on a larger range, is to combine the local models (6.11) to

$$\frac{d}{dt}\left(\sum_{i=0}^{s} w_i(\mathbf{x}(\mathbf{t}))\left[\mathbf{E_i}\mathbf{x}(t) + \boldsymbol{\delta}_i\right]\right) + \sum_{i=0}^{s} w_i(\mathbf{x}(\mathbf{t}))\left[\mathbf{A}_i\mathbf{x}(t) + \boldsymbol{\gamma}_i\right] + \mathbf{Bu}(t) = \mathbf{0}, \tag{6.12a}$$

where $w_i : \mathbb{R}^n \to [0, 1]$ for $s = 1, \ldots, s$ is a state-dependent weight-function. The weighting functions w_i are chosen such that $w_i(\mathbf{x}(t))$ is large for $\mathbf{x}$ close to $\mathbf{x}_i^{\text{lin}}$ and such that $w_0(\mathbf{x}(t)) + \cdots + w_s(\mathbf{x}(t)) = 1$.

A different way to define a global substitute model, suggested by Voß [49] is

$$\sum_{i=0}^{s} w_i(\mathbf{x}(t))\left(\mathbf{E}_i\frac{d}{dt}\mathbf{x}(t) + \mathbf{A}_i\mathbf{x}(t) + \gamma_i\right) + \mathbf{Bu}(t) = \mathbf{0}. \tag{6.12b}$$

Although different in definition, in deployment both approaches (6.12a) and (6.12b) are equivalent, as we will see later.

Figure 6.1 illustrates the idea: Along a training trajectory, extracted from a full dimensional simulation, a set of locally valid linear models is created. When this model is used for simulation with a different input, the existing linear models are turned on and off, adapted to the state the system is in at one moment.

Simulation of the piecewise linearized system (6.12a) or (6.12b) may already be faster than simulation of the original nonlinear system. However, the linearized system can be reduced by using model order techniques for linear systems to increase efficiency.

The main difference between linear MOR and TPWL is that the latter introduces in addition to the application of a linear MOR technique the selection of lineariza-

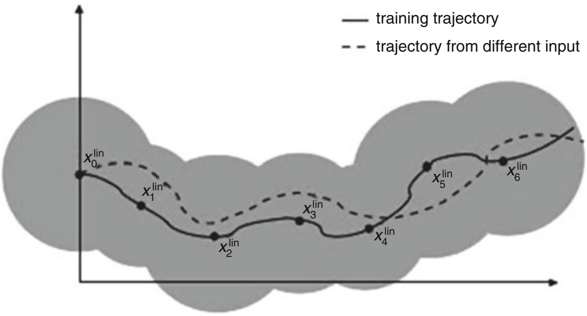


Fig. 6.1 TPWL – model extraction and usage

tion points (to get a linear problem) and the weighting of the linear submodels (to recover the global nonlinear behavior).

Reducing the System

Basically, any MOR-technique for linear problems can be applied to the linear submodels (6.11), i.e., $(\mathbf{E}_i, \mathbf{A}_i, [\mathbf{B}, \boldsymbol{\gamma}_i], \mathbf{C})$. Note that we did extend the columns of $\mathbf{B}$ with $\boldsymbol{\gamma}_i$ – thus MOR may exploit refinements for multiple terminals (see, f.i., Sect. 6.2 in this Chapter). Originally Rewieński [33] proposed the usage of Krylov-based reduction. Vasilyev, Rewieński and White [40] introduced Balanced Truncation to TPWL and Voß [49] uses Poor Man's TBR (PMTBR) as linear MOR kernel. Each of these methods creates local subspaces, spanned by the columns of projection matrices $\mathbf{V}_i \in \mathbb{R}^{n\times r_i}$ for $i = 0, \dots, s$. For some comparisons on different MOR methods used within TPWL, see [29]. For comparison between TPWL and POD (see Sect. 6.1.3.2), see [7, 42].

In a second step one global subspace is created from the information contained in the local subspaces. This is done by applying a singular value decomposition (SVD) on the aggregated matrix $\mathbf{V}_{\text{agg}} = [\mathbf{V}_0, \mathbf{x}_0^{\text{lin}}; \dots; \mathbf{V}_s, \mathbf{x}_s^{\text{lin}}]$. Note that the $\mathbf{x}_j^{\text{lin}}$ are "snapshots" in time of the nonlinear solution. Their span actually forms a POD-subspace (see Sect. 6.1.3.2) that is collected on-the-fly within TPWL. The inclusion reduces the error of the solution of the reduced model [7].

The final reduced subspace is then spanned by the r dominating left singular vectors, subsumed in $\mathbf{V} \in \mathbb{R}^{n\times r}$. Furthermore let $\mathbf{W} \in \mathbb{R}^{n\times r}$ be the corresponding test matrix, where often we have $\mathbf{W} = \mathbf{V}$. Then a reduced order model for the piecewise-linearized system (6.12a) is

$$\frac{d}{dt}\left(\sum_{i=0}^{s} w_i(\mathbf{V}\mathbf{z}(t))\left[\hat{\mathbf{E}}_i\mathbf{z}(t) + \hat{\boldsymbol{\delta}}_i\right]\right) + \sum_{i=0}^{s} w_i(\mathbf{V}\mathbf{z}(t))\left[\hat{\mathbf{A}}_i\mathbf{z}(t) + \hat{\boldsymbol{\gamma}}_i\right] + \hat{\mathbf{B}}\mathbf{u}(t) = \mathbf{0}, \tag{6.13}$$

with $\hat{\mathbf{E}}_i = \mathbf{W}^T\mathbf{E}_i\mathbf{V}$, $\hat{\mathbf{A}}_i = \mathbf{W}^T\mathbf{A}_i\mathbf{V}$, $\hat{\boldsymbol{\delta}}_i = \mathbf{W}^T\boldsymbol{\delta}_i$, $\hat{\boldsymbol{\gamma}}_i = \mathbf{W}^T\boldsymbol{\gamma}_i$ and $\hat{\mathbf{B}} = \mathbf{W}^T\mathbf{B}$.

Selection of Linearization Points

A crucial point in TPWL is to decide, which linearization points $\mathbf{x}_0^{\text{lin}}, \dots, \mathbf{x}_s^{\text{lin}}$ should be chosen. With a large number of such points, we could expect to find a linear model suitable to reproduce the nonlinear behaviour locally. But, this would especially cause to store huge amount of data, making the final model slow. On the other hand, if too few points are chosen to linearise around, the nonlinear behaviour will not be reflected correctly. Different strategies to decide upon adding a new linearization point, and hence a new model automatically exist:

- In the original work, Rewieński [33, 34] suggests to check at each accepted timepoint t during simulation for the relative distance of the current state $\mathbf{x}_k \approx \mathbf{x}(t_k)$ of the nonlinear problem to all yet existing i linearization states

$\mathbf{x}_0^{\text{lin}}, \ldots, \mathbf{x}_{i-1}^{\text{lin}}$. If the minimum is equal to or greater than some parameter $\alpha > 0$, i.e.

$$\min_{0 \le j \le i-1} \left(\frac{\|\mathbf{x}_k - \mathbf{x}_j^{\text{lin}}\|_\infty}{\|\mathbf{x}_j^{\text{lin}}\|_\infty} \right) \ge \alpha, \tag{6.14}$$

$\mathbf{x}_k$ becomes the $(i+1)$st linearization point. Accordingly, a new linear model, arising from linearizing around $\mathbf{x}_{i+1}^{\text{lin}} = \mathbf{x}_k$ is added to the collection. The parameter α is chosen depending on the steady state of the system (6.1a).

- In [49] the mismatch of nonlinear and linear system motivates the creation of a new linearization point and an additional linear model: at each timepoint during training both the nonlinear and a currently valid linear system are computed in parallel with the same stepsize. If the difference of the two approximations to the true solution at a timepoint t_{k+1} becomes too large, a new linear model is created from linearizing the nonlinear system around the state the system was in at the previous timepoint t_k.
- The strategy pursued by Dong an Roychowdhury [18] is similar to (6.14). Here, not deviations between states but function evaluations at the current approximation $\mathbf{x}_k$ and the linearization points $\mathbf{x}_j^{\text{lin}}$ are considered.
- In Martinez [28] an optimization criterion is used to determine the linearization points. The technique exploits the Hessian of the system as an error bound metric.

Determination of the Weights

When replacing the full nonlinear problem with the TPWL model (6.13) the weights $w_i : \mathbb{R}^n \to [0,1]$ are responsible for switching between the linear submodels, i.e., for choosing the linear model that reflects best the behaviour caused by the nonlinearity.

Besides the specifications of the desired behaviour, made before, one wants to have minimum complexity, i.e., one aims at having to deal with a combination of just a small number of linear submodels at each timepoint. It is hence obvious that the weight functions have to be nonlinear in nature. Again, different strategies exist:

- Both Rewieński [33] and Voß [49] use

$$w_i(\mathbf{x}) = e^{-\frac{\beta}{m} \cdot d_i(\mathbf{x})}, \quad \text{with } d_i(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_i^{\text{lin}}\|_2 \text{ and } m = \min_i d_i(\mathbf{x}). \tag{6.15a}$$

The constant β adjusts how abrupt the change of models is. A typical value is $\beta = 25$.

- Dong and Roychowdhury [18], however, use

$$w_i(\mathbf{x}) = \left(\frac{m}{d_i(\mathbf{x})} e^{\frac{-d_i(\mathbf{x}) - m}{M}} \right)^{\mu}, \tag{6.15b}$$

where $d_i(\mathbf{x})$ and m are the same as in (6.15a) and M is the minimum distance, taken in the 2-norm, amongst the linearization points. The parameter μ is chosen from $\{1, 2\}$.

In both cases, the weights are normalized such that $\sum_i w_i(\mathbf{x}) = 1$.

Clearly the nonlinearity of the weights causes the TPWL-model (6.13) arising from (6.12a) – and similar the reduced model that would originate from (6.12b) – to be nonlinear still. That means, after applying a numerical integration scheme to (6.13) , still a nonlinear problem has to be solved to get an approximation $\mathbf{z}_k \approx \mathbf{z}(t_k)$. To overcome this problem, both Rewieński [33] and Voß [49] decouple the evaluation of the weights from the time discretisation by replacing

$$w_i(\mathbf{V}\mathbf{z}_k) \rightsquigarrow w_i(\mathbf{V}\tilde{\mathbf{z}}_k) \quad \text{with } \tilde{\mathbf{z}}_k \approx \mathbf{z}_k,$$

i.e., for calculating $\mathbf{z}_k$ from the discretisation of (6.13) at $t = t_k$, $\mathbf{z}_k$ in the weighting is replaced by a cheaper approximation $\tilde{\mathbf{z}}_k$. It is easy to see that with this action, (6.12a) and (6.12b) are equivalent.

Note: The work of Dong and Roychowdhury [18] does actually not consider a piecewise linear a but piecewise polynomial approach, i.e., the Taylor expansions in (6.10) contain one more coefficient, leading to the need for reducing local bilinear systems. Tiwary and Rutenbar [39] look into details of implementing a TPWL-technique in an economic way.

TPWL and Time-Domain MOR

In [53] the TPWL approach is combined with wavelet expansions that are defined directly in the time-domain. For wavelets in circuit simulation we refer to [16, 17] and for technical details to [10, 11, 51, 52]. After linearizing a differential equation

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{B}\mathbf{u}(t) \tag{6.16}$$

at $\mathbf{x}_i = \mathbf{x}(t_i)$, we obtain that $\tilde{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}_i$ is approximately given by

$$\frac{d}{dt}\tilde{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}_i) + \mathbf{A}(\tilde{\mathbf{x}}(t) - \mathbf{x}_i) + \mathbf{B}\mathbf{u}(t), \tag{6.17}$$

$$= \mathbf{A}\tilde{\mathbf{x}}(t) + \mathbf{f}(\mathbf{x}_i) - \mathbf{A}\mathbf{x}_i + \mathbf{B}\mathbf{u}(t). \tag{6.18}$$

where $\mathbf{A} = \frac{\partial f(x)}{\partial x}(t_i)$. The output request $\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t)$ transfers to $\tilde{\mathbf{y}}(t) = \mathbf{C}\mathbf{x}_i + \mathbf{C}\tilde{\mathbf{x}}(t)$, in which the first term is known. Thus it is sufficient to consider on an interval $[0, T]$ the sum of the solutions of the two problems

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \tag{6.19}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \tag{6.20}$$

and

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{f}(\mathbf{x}_i) - \mathbf{A}\mathbf{x}_i, \tag{6.21}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t). \tag{6.22}$$

Assuming T being integer (see [53] for the more general case), for a wavelet order J we get $M = 2^J \cdot T + 3$ basis functions $\theta_j(t)$, $j = 1, \ldots M$. We can write $\mathbf{x}(t) = \mathbf{H}_1\theta(t)$, and $\mathbf{x}(t) = \mathbf{H}_2\theta(t)$, respectively, where $\theta(t) = (\theta_1(t), \ldots, \theta_M(t))$ and $\mathbf{H}_1, \; \mathbf{H}_2 \in \mathbb{R}^{n \times M}$. We can plug these expressions into (6.19) and into (6.21). However, note that in (6.19) the source term is time-dependent, while in (6.21) the source term is constant. Hence rather then to consider (6.19), one considers

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\delta(t), \tag{6.23}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \tag{6.24}$$

where $\delta(t)$ is an impulse excitation with the property $\int_{0_-}^{t} \delta(\tau)d\tau = 1$. Then, for (6.23), the matrix $\mathbf{H}_1$ satisfies

$$\mathbf{H}_1\frac{d}{dt}\theta(t) = \mathbf{A}\mathbf{H}_1\theta(t) + \mathbf{B}\delta(t) \tag{6.25}$$

Assuming that the wavelets have their support in $[0, T]$ we derive

$$\mathbf{H}_1\theta(t) = \mathbf{A}\mathbf{H}_1 \int_{0_-}^{t} \theta(\tau)d\tau + \mathbf{B}. \tag{6.26}$$

In [53] one applies collocation using M collocation points. Next $\mathbf{H}_1$ is found after solving the resulting Sylvester equation. A similar approach is done for (6.21). Now one determines $\mathbf{V}_i = \mathit{Orthog}(\mathbf{H}_1, \mathbf{H_2}, \mathbf{x}_i)$ (note that this is similar to the multiple terminal approach mentioned before for the frequency domain case). From $(\mathbf{V}_1, \ldots, \mathbf{V}_s)$ one determines an overall orthonormal basis $\mathbf{V} \in \mathbb{R}n \times r$ that is used for the projection as before.

6.1.3.2 Proper Orthogonal Decomposition and Adaptions

The Proper Orthogonal Decomposition (POD) method, also known as the Principal Component Analysis and Karhunen–Loève expansion, provides a technique for analysing multidimensional data [24, 27].

In this section we briefly describe some basics of POD. For a more detailed introduction to POD in MOR we refer to [31, 47]. For further studies we point to [32], which addresses error analysis for MOR with POD and [50] where the connection of POD to balanced model reduction can be found.

POD sets work on data extracted from a benchmark simulation. In a finite dimensional setup like it is given by (6.1a), K snapshots of the state $\mathbf{x}_i \approx \mathbf{x}(t_i)$, the system is in during the training interval $[t_{\text{start}}, t_{\text{end}}]$, are collected in a snapshot matrix

$$\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_K) \quad \in \mathbb{R}^{n \times K}. \tag{6.27}$$

The snapshots, i.e., the columns of $\mathbf{x}$, span a space of dimension $k \leq K$. We search for an orthonormal basis $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ of this space that is optimal in the sense that the time-averaged error that is made when the snapshots are expanded in the space spanned by just $r < k$ basis vectors to $\tilde{\mathbf{x}}_{r,i}$,

$$\langle \|\mathbf{x} - \tilde{\mathbf{x}}_r\|_2^2 \rangle \quad \text{with the averaging operator} \quad \langle \mathbf{f} \rangle = \frac{1}{K} \sum_{i=1}^{K} \mathbf{f}_i \tag{6.28}$$

is minimised. This least squares problem is solved by computing the eigenvalue decomposition of the state covariance matrix $\frac{1}{K}\mathbf{x}\mathbf{x}^T$ or, equivalently by the singular value decomposition (SVD) of the snapshot matrix (assuming $K > n$)

$$\mathbf{x} = \mathbf{UST} \quad \text{with} \quad \mathbf{U} \in \mathbb{R}^{n \times n}, \mathbf{T} \in \mathbb{R}^{K \times K} \text{ and } \mathbf{S} = \left(\begin{array}{ccc|c} \sigma_1 & & & \\ & \ddots & & \mathbf{0}_{n \times (K-n)} \\ & & \sigma_n & \end{array} \right), \tag{6.29}$$

where $\mathbf{U}t$ and $\mathbf{T}$ are orthogonal and the singular values satisfy $\sigma_1 \geq \sigma_2 \geq \cdots \sigma_n \geq 0$. The matrix $\mathbf{V} \in \mathbb{R}^{n \times r}$ whose columns span the reduced subspace is now build from the first r columns of $\mathbf{u}$, where the truncation r is chosen such that

$$\frac{\sum_{i=1}^{r} \sigma_i^2}{\sum_{i=1}^{n} \sigma_i^2} \geq \frac{d}{100}, \tag{6.30}$$

where usually $d = 99$ is usually a reasonable choice. For the, in this way constructed matrix, it holds $\mathbf{V}^T\mathbf{V} = \mathbf{I}_{r \times r}$. Therefore, Galerkin projection as described above can be applied to create a reduced system (6.2).

However, as mentioned in Sect. 6.1.1 the cost for evaluating the nonlinear functions $\mathbf{q}$, $\mathbf{j}$ is not reduced. In the following we describe some adaptions to POD that have been made to overcome this problem.

6.1.3.3 Missing Point Estimation

The Missing Point Estimation (MPE) was proposed by Astrid [2, 4] to reduce the cost of updating system information in the solution process of time varying systems arising in computational fluid dynamics. Verhoeven and Astrid [3] brought the MPE approach forward to circuit simulation.

Once a POD basis is constructed, there is no Galerkin projection deployed. Instead a numerical integration scheme is applied which in general leads to system of n nonlinear equations, analogue to (6.6), for the r dimensional unknown $\mathbf{z}_l$, that approximate $\mathbf{z}(t_l)$. In MPE this system is reduced to dimension g with $r \leq g < n$ by discarding $n-g$ equations. Formally this can be described by multiplying the system with a selection matrix[3] $\mathbf{P}_g \in \{0, 1\}^{g\times n}$, stating a g-dimensional overdetermined problem

$$\alpha\bar{\mathbf{q}}(\mathbf{V}\mathbf{z}_l) + \mathbf{P}_g\boldsymbol{\beta}_l + \bar{\mathbf{j}}(\mathbf{V}\mathbf{z}_l) + \mathbf{P}_g\mathbf{B}\mathbf{u}(t_l) = \mathbf{0}, \tag{6.31}$$

with $\bar{\mathbf{q}}(\mathbf{V}\mathbf{z}_l) = \mathbf{P}_g\mathbf{q}(\mathbf{V}\mathbf{z}_l)$ and $\bar{\mathbf{j}}(\mathbf{V}\mathbf{z}_l) = \mathbf{P}_g\mathbf{j}(\mathbf{V}\mathbf{z}_l)$. The system (6.31) is solved at each timepoint t_l for $\mathbf{z}_l$ in the least-squares sense [3, 41, 44, 45].

The effect of $\mathbf{P}_g$ operating on $\mathbf{q}(\cdot)$ and $\mathbf{j}(\cdot)$ is the same as evaluating only the $g \ll n$ components of $\mathbf{q}$ and $\mathbf{j}$ corresponding to the columns $\mathbf{P}_g$ has a 1 in.

The choice of $\mathbf{P}_g$ is motivated by identifying the g most dominant state variables, i.e., components of $\mathbf{x}$. In terms of the POD basis this is connected to restricting the orthogonal $\mathbf{V}$ to $\tilde{\mathbf{V}} = \mathbf{P}_g\mathbf{V} \in \mathbb{R}^{g\times r}$ in an optimal way. This in turn goes down to

$$\min_{\mathbf{P}_g} \| \left(\tilde{\mathbf{V}}^T\tilde{\mathbf{V}}\right)^{-1} - \mathbf{I}_{r\times r}\|. \tag{6.32}$$

Details on reasoning and solving (6.32) can be found in [4].

6.1.3.4 Adapted POD

A second approach to reduce the work of evaluating the nonlinear functions, Adapted POD, was proposed in [41, 43–45]. Having done an SVD (6.29) on the snapshot matrix, not directly a projection matrix $\mathbf{V}$ is defined from the singular values and vectors. Instead the matrix $\mathbf{L} = \mathbf{u}\Sigma \in \mathbb{R}^{n\times n}$, with $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$ is defined. Hence, $\mathbf{L}$ arises from scaling the left-singular vectors with the corresponding singular values. Although $\mathbf{L}$ is not orthogonal, its columns are. Next we transform the original system (6.1a) by writing $\mathbf{x}(t) = \mathbf{L}\mathbf{w}(t)$ with $\mathbf{w}(t) \in \mathbb{R}^n$ and using the Galerkin approach:

$$\frac{d}{dt}\left[\mathbf{L}^T\mathbf{q}(\mathbf{L}\mathbf{w}(t))\right] + \mathbf{L}^T\mathbf{j}(\mathbf{L}\mathbf{w}(t)) + \mathbf{L}^T\mathbf{B}\mathbf{u}(t) = \mathbf{0}. \tag{6.33}$$

At this point, $\mathbf{L}$ and $\mathbf{L}^T$ are treated as two different matrices, one acting on the parameter of the function, the other on the value. For both $\mathbf{L}$ and $\mathbf{L}^T$ we identify the r and g, most dominant columns. A measure for the significance of a column vector $v \in \mathbb{R}^n$ is its 2-norm $\|v\|_2$.

[3] This means, the matrix has exactly one non-zero entry per row at most one non-zero per column.

As the columns of $\mathbf{L}$ are ordered according to the singular values, we will pick the first r columns in this case. Now $\mathbf{L}$ and $\mathbf{L}^T$ are approximated by matrices that agree with the respective matrix in the selected r and g selected columns but have the $n-r$ and $n-g$, respectively, remaining columns set to $\mathbf{0} \in \mathbb{R}^n$. This can be expressed with the help of selection matrices $\mathbf{P}_r \in \{0,1\}^{r\times n}$ and $\mathbf{P}_g \in \{0,1\}^{g\times n}$, respectively:

$$\mathbf{L} \approx \mathbf{L}\mathbf{P}_r^T\mathbf{P}_r \quad \text{and} \quad \mathbf{L}^T \approx \mathbf{L}^T\mathbf{P}_g^T\mathbf{P}_g. \tag{6.34}$$

We may conclude $\mathbf{L}^T \approx \mathbf{P}_r^T\mathbf{P}_r\mathbf{L}^T\mathbf{P}_g^T\mathbf{P}_g$, insert these approximations in (6.33) and multiply with $\mathbf{P}_r$, bearing in mind that $\mathbf{P}_r\mathbf{P}_r^T = \mathbf{I}_{r\times r}$:

$$\frac{d}{dt}\left[\mathbf{P}_r\mathbf{L}^T\mathbf{P}_g^T\mathbf{P}_g\mathbf{q}(\mathbf{L}\mathbf{P}_r^T\mathbf{P}_r\tilde{\mathbf{w}})\right]+\mathbf{P}_r\mathbf{L}^T\mathbf{P}_g^T\mathbf{P}_g\mathbf{j}(\mathbf{L}\mathbf{P}_r^T\mathbf{P}_r\tilde{\mathbf{w}})+\mathbf{P}_r^T\mathbf{L}^T\mathbf{B}\mathbf{u} = \mathbf{0}. \tag{6.35}$$

Note that due to the approximations to $\mathbf{L}$ and $\mathbf{L}^T$ in the above equation $\mathbf{w}$ has changed to $\tilde{\mathbf{w}}$ which can merely be an approximation to the former. We introduce $\boldsymbol{\Sigma}_r = \mathrm{diag}(\sigma_1,\ldots,\sigma_r)$ and let $\mathbf{V} \in \mathbb{R}^{n\times r}$ be the first r columns of $\mathbf{u}$. In this wa we have $\mathbf{L}\mathbf{P}_r^T = \mathbf{V}\mathbf{S}_r$. Finally we scale (6.35) with $\boldsymbol{\Sigma}_r^{-1}$ and introduce a new unknown $\mathbf{z} = \boldsymbol{\Sigma}_r\mathbf{P}_r\tilde{\mathbf{w}} \in \mathbb{R}^r$ from which we can reconstruct the full state by approximation $\mathbf{x} \approx \mathbf{V}\mathbf{z}$. We end up with

$$\frac{d}{dt}\left[\mathbf{W}_{r,g}\bar{\mathbf{q}}(\mathbf{V}\mathbf{z})\right] + \mathbf{W}_{r,g}\bar{\mathbf{j}}(\mathbf{V}\mathbf{z}) + \tilde{\mathbf{B}}\mathbf{u}(t) = \mathbf{0}, \tag{6.36}$$

with $\bar{\mathbf{q}}(\mathbf{V}\mathbf{z}) = \mathbf{P}_g\mathbf{q}(\mathbf{V}\mathbf{z})$, $\bar{\mathbf{j}}(\mathbf{V}\mathbf{z}) = \mathbf{P}_g\mathbf{j}(\mathbf{V}\mathbf{z})$, $\mathbf{W}_{r,g} = \mathbf{V}^T\mathbf{P}_g^T \in \mathbb{R}^{r\times g}$ and $\tilde{\mathbf{B}} = \mathbf{V}^T\mathbf{B}$.

Here $\mathbf{P}_g$ has the same effect as noted in the previous subsection: not the full nonlinear functions $\mathbf{q}$ and $\mathbf{j}$ have to be evaluated but g components only.

6.1.3.5 Discrete Empirical Interpolation

Recently, Chaturantabut and Sorensen [12, 13] did present the Discrete Empirical Interpolation Method (DEIM) as a further modification of POD. It originates from partial differential equations (PDEs) where the nonlinearities exhibit a special structure. It can, however, be applied to general nonlinearities as well. We give a brief introduction of how this may look like in circuit simulation problems.

Given a nonlinear function $\mathbf{f}: \mathbb{R}^n \to \mathbb{R}^n$, the essential idea of DEIM is to approximate $\mathbf{f}(\mathbf{x})$ by projecting it on a subspace, spanned by the basis $\{\mathbf{u}_1,\ldots,\mathbf{u}_g\} \subset \mathbb{R}^n$:

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{U}\mathbf{c}(\mathbf{x}), \tag{6.37}$$

where $\mathbf{U} = (\mathbf{u}_1,\ldots,\mathbf{u}_g) \in \mathbb{R}^{n\times g}$ and $\mathbf{c}(\mathbf{x}) \in \mathbb{R}^g$ is the coefficient vector. Forcing equality in (6.37) would state an overdetermined system for the $g < n$ coefficients

$\mathbf{c}t(\mathbf{x})$. Instead accordance in g rows is required, which can be expressed by

$$\mathbf{P}_g\mathbf{f}(\mathbf{x}) = (\mathbf{P}_g\mathbf{U})\mathbf{c}(\mathbf{x}), \tag{6.38}$$

with a selection matrix $P_g \in \{0,1\}^{g\times n}$. If $\mathbf{P}_g\mathbf{U}$ is non-singular, (6.38) has a unique solution $\mathbf{c}(\mathbf{x})$ and, hence $\mathbf{f}(\mathbf{x})$ can be approximated by

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{U}\left(\mathbf{P}_g\mathbf{U}\right)^{-1}\mathbf{P}_g\mathbf{f}(\mathbf{x}), \tag{6.39}$$

which means that $\mathbf{f}(\mathbf{x})$ is interpolated at the entries specified by $\mathbf{P}_g$.

In (6.39), $\mathbf{U}\left(\mathbf{P}_g\mathbf{U}\right)^{-1}$ can be computed in advance, and, again, the multiplication $\mathbf{P}_g\mathbf{f}(\mathbf{x})$ corresponds to evaluating only those entries of $\mathbf{f}$, addressed by $\mathbf{P}_g$.

Using the notations introduced before, POD with the DEIM modification yields a reduced model (6.2) with

$$\hat{\mathbf{q}}(\mathbf{z}(t)) = \hat{\mathbf{W}}\,\bar{\mathbf{q}}(\mathbf{z}(t)), \quad \hat{\mathbf{j}}(\mathbf{z}(t)) = \hat{\mathbf{W}}\,\bar{\mathbf{j}}(\mathbf{z}(t)), \quad \hat{\mathbf{B}} = \mathbf{V}^T\mathbf{B}, \quad \hat{\mathbf{C}} = \mathbf{C}\mathbf{V}^T, \tag{6.40}$$

with $\hat{\mathbf{W}} = \mathbf{V}^T\mathbf{U}\left(\mathbf{P}_g\mathbf{U}\right)^{-1}$ and $\bar{\mathbf{q}}(\cdot) = \mathbf{P}_g\mathbf{q}(\cdot)$ and $\bar{\mathbf{j}}(\cdot) = \mathbf{P}_g\mathbf{j}(\cdot)$. Here POD provides the state-space part of the reduction, i.e., $\mathbf{V}$. And DEIM determines the subspace on which $\mathbf{q}$ and $\mathbf{j}$ is projected, hence the columns of the matrix $\mathbf{U} \in \mathbb{R}^{n\times g}$ and the selection $\mathbf{P}_g$.

The reduced subspace, suitable for representing a nonlinear function $\mathbf{f}$ on, is constructed from an SVD on a matrix $\mathbf{F} = (\mathbf{f}(\mathbf{x}_1),\ldots,\mathbf{f}(\mathbf{x}_K)) \in \mathbb{R}^{n\times K}$ whose columns are snapshots of the function evaluations. The matrix $\mathbf{U}$ in (6.39) consists then of the g most dominant left singular vectors of $\mathbf{F}$.

The core of DEIM is the construction of the selection $\mathbf{P}_g \in \{0,1\}^{g\times n}$. A set of indices $\{\rho_1,\ldots,\rho_g\} \subset \{1,\ldots,n\}$, determined by the DEIM-algorithm, define the selection matrix, meaning that $\mathbf{P}_g$ has a 1 in the ith row and ρ_ith column (for $i = 1,\ldots,g$) and 0 elsewhere.

The first index, ρ_1 is chosen to be the index of the largest (in absolute value) entry in $\mathbf{u}_1$. In step $l = 2,\ldots,g$ the residual

$$\mathbf{r}_{l+1} = \mathbf{U}_{l+1} - \mathbf{U}_l\,(\mathbf{P}_l\mathbf{U}_l)^{-1}\,\mathbf{P}_l\mathbf{U}_{l+1}$$

is computed where $\mathbf{U}_l = (\mathbf{u}_1,\ldots,\mathbf{u}_l)$ and $\mathbf{P}_l \in \{0,1\}^{l\times n}$ is constructed from the indices $\rho_1,\ldots,\rho_l$ (cp. (6.39)). Then, the index corresponding to entry of the residual $\mathbf{r}_{l+1}$ the largest magnitude of is taken as index ρ_{l+1}.

Setting up the selection matrix with this algorithm, $\mathbf{P}_g\mathbf{U}$ in (6.37) is guaranteed to be regular. For a detailed description and discussion, including error estimates we refer to [12, 13].

Note: Originally, DEIM is constructed in the context of discretisation and approximation of PDEs with a special structure of the nonlinearity involved. Considering network problem (6.1a) that leads to the reduced problem (6.40), we constructed a uniform DEIM-approximation, i.e., $\mathbf{U}$ and $\mathbf{P}_g$ for the both nonlinearities, $\mathbf{q}$ and $\mathbf{j}$ involved. This could probably be approached in a different way, too.

6.1.4 Other Approaches

We shortly address some other approaches. In [5, 6, 8, 48] Krylov-subspace methods are applied to bilinear and quadratic-bilinear ODE-systems. One exploits the observation that several nonlinear functions can be generated by extending the system first with additional unknowns for which simple differential equations are introduced. In [48] also the application to DAEs is discussed. In [22] a transformation from a set of nonlinear differential equations to another set of equivalent nonlinear differential equations that involve only quadratic terms of state variables is described to which Volterra analysis is applied to derive a reduced model.

We already mentioned [20] for nonlinear balancing in which the energy functions arise from solving Hamilton-Jacobi differential equations. Related work is on cross Gramians for dissipative and symmetric nonlinear systems [25, 26].

In [37, 38] interpolating input-output behavior of nonlinear systems is studied. This is related to table modelling.

6.1.5 Numerical Experiments

For testing purposes, a time-simulator, has been implemented in `octave`. The underlying DAE integration scheme used here is CHORAL [23], a Rosenbrock-Wanner type of method, adapted to circuitry problems. Besides performing transient-analysis, TPWL and POD models can be extracted and reused in simulations.

To show the performance of TPWL and POD when applied to an example from circuit design, the nonlinear transmission line in Fig. 6.2, taken from [33] is chosen. Only the diodes introduce the designated nonlinearity to the circuit, as the current $\imath_d$ traversing a diode is modeled by $\imath_d(v) = \exp(40 \cdot v) - 1$ where v is the voltage drop between the diode's terminals. The resistors and capacitors contained in the model have unit resistance and capacitance ($R = C = 1$), respectively. The current

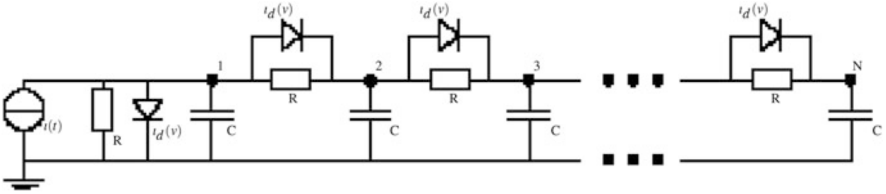


Fig. 6.2 Nonlinear transmission line

source between node 1 and ground marks the input to the system $u(t) = \imath(t)$ and the output of the system is chosen to be the voltage at node 1: $y(t) = v_1(t)$.

Introducing the state vector $\mathbf{x}(t) = (v_1(t), \ldots, v_N(t))^T \in \mathbb{R}^N$, where $v_i(t)$ describes the voltage at node $i \in \{1, \ldots, N\}$ modified nodal analysis yields:

$$\begin{aligned} \frac{d}{dt}\mathbf{x}(t) + \mathbf{j}(\mathbf{x}(t)) + \mathbf{B}u(t) &= \mathbf{0} \\ y(t) &= \mathbf{C}\mathbf{x}(t), \end{aligned} \tag{6.41}$$

where $\mathbf{B} = \mathbf{C}^T = (1, 0, \ldots, 0)^T \in \mathbb{R}^N$ and $\mathbf{j} : \mathbb{R}^N \to \mathbb{R}^N$ with

$$\mathbf{j}(\mathbf{x}) = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{pmatrix} \cdot \mathbf{x} - \begin{pmatrix} 2 - e^{40x_1} - e^{40(x_1-x_2)} \\ e^{40(x_1-x_2)} - e^{40(x_2-x_3)} \\ \vdots \\ e^{40(x_{N-2}-x_{N-1})} - e^{40(x_{N-1}-x_N)} \\ e^{40(x_{N-1}-x_N)} - 1 \end{pmatrix}$$

We choose $N = 100$, causing a problem of dimension $n = 100$.

For extracting a model a shifted Heaviside function was used as training input. Resimulation was done both with the training input and with a cosine function on the interval $[t_{\text{start}}, t_{\text{end}}] = [0, 10]$:

$$u_{\text{train}}(t) = H(t-3) = \begin{cases} 0 & t < 3 \\ 1 & t \geq 3 \end{cases} \qquad u_{\text{resim}}(t) = \frac{1}{2}\left(1 + \cos\left(\frac{2\pi}{10}t\right)\right).$$

The TPWL-model was extracted with the Arnoldi-method as suggested in [33], leading to a order reduced model of dimension 10. For choosing linearization points, the strategy proposed by Rewieński with $\alpha = 0.0167$ in (6.14) has been tested. With this setting, 27 linear models are constructed. Also the extended strategy described in Voß [49] is implemented, but does not show much different results for the transmission line. A more detailed discussion on the model extraction and statistics on which models are chosen can be found in [35, 36].

For the transmission line, also a POD model as well as a POD model that has been modified with the Discrete Empirical Interpolation Method (DEIM) algorithm is constructed. By choosing $d = 99.9$ in (6.30) a reduced model of dimension 4 is constructed. Applying the DEIM algorithm the nonlinear $\mathbf{q}$ and $\mathbf{j}$ where reduced to order 5. Figure 6.3 displays the singular values form snapshots collected during a training run and the behaviour of the coverage function (6.30). Note, that only 38 singular values are shown, although the full system is of dimension 100. This is caused by the time domain simulation: with tolerances specified for the timestepping mechanism, only 38 time steps where necessary to resolve the system. However, also with more snapshots, the gradient of the singular values does not change remarkably.

Figures 6.4 and 6.5 show the trajectories, i.e., the behaviour in time, of the voltages at nodes 1 and ten, when the training signal is and when the cosine like

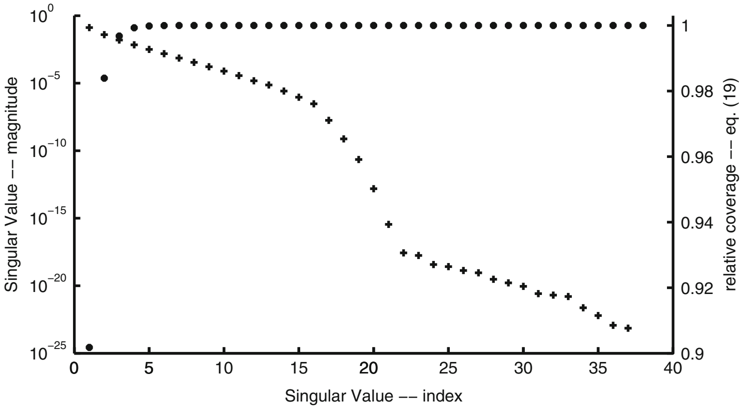


Fig. 6.3 Transmission line: singular values (+) & coverage (∗)

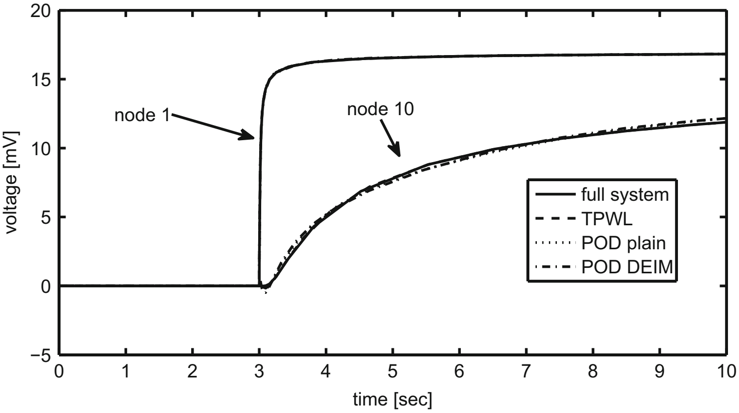


Fig. 6.4 Nonlinear transmission line: resimulation results

signal is applied at the input, respectively. The plots show the signals reproduced by using the full model, the TPWL-model and the plain POD and DEIM-adapted POD model. Slight deviations from the reference solution are obvious, but, in total, a good matching is observable. However, the TPWL-model seems to have problems following the reference solution, when a signal, different to the input is applied. This indicates that there are still improvements possible.

Finally, Table 6.1 gathers the performance of the models, measured in time consumption. Clearly, simulation with the TPWL model is cheaper than using the full network as not the full nonlinearity has to be evaluated. Still, POD, adapted with DEIM is superior, as no decision has to be made, which model to use. Furthermore, as predicted in Sect. 6.1.1 applying only projection without taking care of the nonlinearity, does not guarantee cheaper to evaluate model: the plain POD model, used for simulation, causes equal or even increased computational expenses.

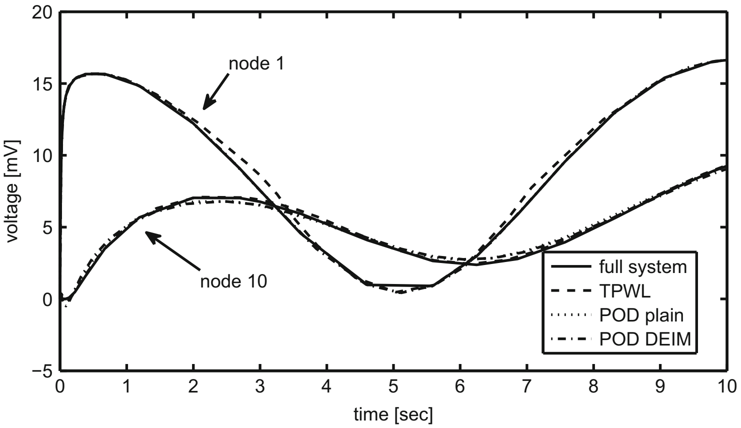


Fig. 6.5 Transmission line: different input

Table 6.1 Transmission line: performance of nonlinear MOR techniques

	Resimulation (s)	Changed input (s)
Full problem	6.67	4.66
TPWL model	4.35	3.47
POD model	6.51	5.23
POD-DEIM model	1.98	1.63

6.2 Model Order Reduction for Multi-terminal Circuits

Analysis of effects due to parasitics is of vital importance during the design of large-scale integrated circuits and derived products.[4] One way to model parasitics is by means of parasitic extraction, which results in large linear $RCL(k)$ networks. In ESD analysis [65, 75], for instance, the interconnect network is modeled by resistors with resistances that are based on the metal properties. In other (RF) applications one needs RC or even $RCLk$ extractions to deal accurately with higher frequencies as well.

The resulting parasitic networks may contain up to millions of resistors, capacitors, and inductors, and hundreds of thousands of internal nodes, and thousands of external nodes (nodes with connections to active elements such as transistors). Simulation of such large networks within reasonable time is often not possible [62, 63], and including such networks in full system simulations may be even unfeasible. Hence, there is need for much smaller networks that accurately or even exactly describe the behavior of the original network, but allow for fast analysis.

[4] Section 6.2 has been written by Roxana Ionutiu and Joost Rommes. For an extended treatment on the topics of this section see also the Ph.D. Thesis of the first author [68].

In this section we describe recently developed methods for the reduction of large R networks, and present a new approach for the reduction of large RC networks. We show how insights from graph theory, numerical linear algebra, and matrix reordering algorithms can be used to construct a reduced network that shows sparsity preservation especially for circuits with multi-terminals (ports). Hence it allows for the same number of external nodes, but needs much fewer internal nodes and circuit elements (resistors and capacitors). Circuit synthesis is applied after model reduction, and the resulting reduced netlists are tested with industrial circuit simulators. For related literature we refer to [55–57].

The section is organized as follows. Section 6.2.1 revisits recent work on reduction of R networks [83, 84]. It provides the basis for understanding how graph theoretical tools can be used to significantly improve the sparsity of the reduced models, which are later synthesized [70] into reduced netlists. Section 6.2.2 deals with the reduction of RC networks. Section 6.2.2.1 first reviews an existing method which employs *Pole Analysis* via *Congruence Transformations* (*PACT*) [73] to reduce RC netlists with multi-terminals. In Sect. 6.2.2.2 the new method *Sparse Modal Approximation (SparseMA)* is presented, where graph-theoretical tools are brought in to enhance sparsity preservation for the reduced models. The numerical results for both R and RC netlist reduction are presented in Sect. 6.2.3. Section 6.2.4 concludes.

6.2.1 Reduction of R Networks

In this section we review the approach for reducing R networks, as developed in [83, 84]. Reduction of R networks, i.e., networks that consist of resistors only, is needed in electro-static discharge analysis (ESD), where large extracted R networks are used to model the interconnect. Accurate modeling of interconnect is required here, since the costs involved may vary from a few cents to millions if, due to interconnect failures, a respin of the chip is needed. An example of a damaged piece of interconnect that was too small to conduct the amount of current is shown in Fig. 6.6.

6.2.1.1 Circuit Equations and Matrices

Kirchhoff's Current Law and Ohm's Law for resistors lead to the following system of equations for a resistor network with N resistors (resistor i having resistance r_i) and n nodes ($n < N$):

$$\begin{bmatrix} R & P \\ -P^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{i}_b \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{i}_n \end{bmatrix}, \tag{6.42}$$

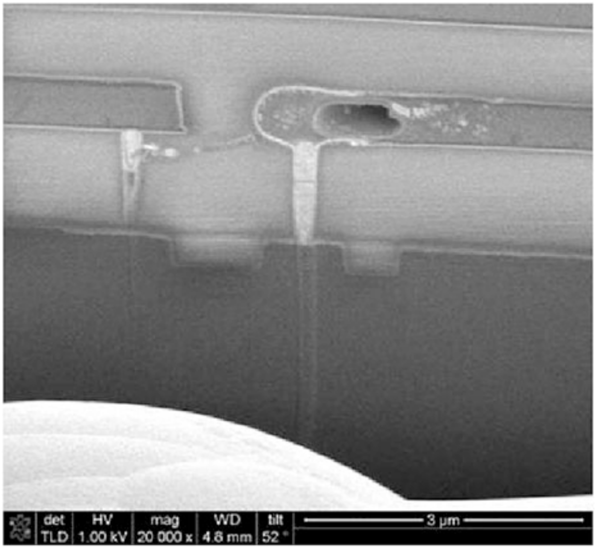


Fig. 6.6 Example of a piece of interconnect that was damaged because it was too small to conduct the amount of current caused by a peak charge

where $R = \text{diag}(r_1, \ldots, r_N) \in \mathbb{R}^{N \times N}$ is the resistor matrix, $P \in \{-1, 0, 1\}^{N \times n}$ is the incidence matrix, $\mathbf{i}_b \in \mathbb{R}^N$ are the resistor currents, $\mathbf{i}_n \in \mathbb{R}^n$ are the injected node currents, and $\mathbf{v} \in \mathbb{R}^n$ are the node voltages.

The MNA (modified nodal analysis) formulation [60, 76] can be derived from (6.42) by eliminating the resistor currents $\mathbf{i}_b = -R^{-1}P\mathbf{v}$:

$$G\mathbf{v} = \mathbf{i}_n, \tag{6.43}$$

where $G = P^T R^{-1} P \in \mathbb{R}^{n \times n}$ is symmetric positive semidefinite. Since currents can only be injected in external nodes, and not in internal nodes of the network, system (6.43) has the following structure:

$$\begin{bmatrix} G_{11} & G_{12} \\ G_{12}^T & G_{22} \end{bmatrix} \begin{bmatrix} \mathbf{v}_e \\ \mathbf{v}_i \end{bmatrix} = \begin{bmatrix} B \\ 0 \end{bmatrix} \mathbf{i}_e, \tag{6.44}$$

where $\mathbf{v}_e \in \mathbb{R}^{n_e}$ and $\mathbf{v}_i \in \mathbb{R}^{n_i}$ are the voltages at external and internal nodes, respectively ($n = n_e + n_i$), $\mathbf{i}_e \in \mathbb{R}^n_e$ are the currents injected in external nodes, $B \in \{-1, 0, 1\}^{n_e \times n_e}$ is the incidence matrix for the current injections, and $G_{11} = G_{11}^T \in \mathbb{R}^{n_e \times n_e}$, $G_{12} \in \mathbb{R}^{n_e \times n_i}$, and $G_{22} = G_{22}^T \in \mathbb{R}^{n_i \times n_i}$. The block G_{11} is also referred to as the terminal block.

A current source (with index s) between terminals a and b with current j results in contributions $B_{a,s} = 1$, $B_{b,s} = -1$, and $\mathbf{i}_e(s) = j$. If current is only injected in a terminal a (for instance if a connects the network to the top-level circuit), the contributions are $B_{a,s} = 1$ and $\mathbf{i}_e(s) = j$.

Finally, systems (6.42)–(6.44) must be made consistent by grounding a node gnd, i.e., setting $\mathbf{v}(gnd) = 0$ and removing the corresponding equations. In the following we will still use the notation G for the grounded system matrix, if this does not lead to confusion.

6.2.1.2 Problem Formulation

The problem is: given a very large resistor network described by (6.42), find an equivalent network with (a) the same external nodes, (b) exactly the same path resistances between external nodes, (c) $\hat{n} \ll n$ internal nodes, and (d) $\hat{r} \ll r$ resistors. Additionally, (e) the reduced network must be realizable as a netlist so that it can be (re)used in the design flow as subcircuit of large systems.

Simply eliminating all internal nodes will lead to an equivalent network that satisfies conditions (a)–(c), but violates (d) and (e): for large numbers m of external nodes, the number of resistors $\hat{r} = (m^2 - m)/2$ in the dense reduced network is in general much larger than the number of resistors in the sparse original network (r of $O(n)$), leading to increased memory and CPU requirements.

6.2.1.3 Existing Approaches

There are several approaches to deal with large resistor networks. In some cases the need for an equivalent reduced network can be circumvented in some way: due to sparsity of the original network, memory usage and computational complexity are *in principle* not an issue, since solving linear systems with the related conductance matrices is typically of complexity $O(n^\alpha)$, where $1 < \alpha \leq 2$, instead of the traditional $O(n^3)$ [79]. Of course, α depends on the sparsity and will rapidly increase as sparsity decreases. This also explains why eliminating all internal nodes does not work in practice: the large reduction in unknowns is easily undone by the enormous increase in number of resistors, mutually connecting all external nodes.

However, if we want to (re)use the network in full system simulations, a reduced equivalent network is needed to limit simulation times or make simulation possible at all. In [77] approaches based on large-scale graph partitioning packages such as (h)METIS [72] are described, but only applied to small networks. Structure preserving projection methods for model reduction [66, 86], finally, have the disadvantage that they lead to dense reduced-order models if the number of terminals is large. There is commercial software [59, 64] available for the reduction of parasitic reduction networks.

6.2.1.4 Improved Approach

Knowing that eliminating all internal nodes is not an option and that projection methods lead to dense reduced-order models, we use concepts from matrix reordering algorithms such as AMD [54] and BBBD [88], usually used as preprocessing step for (parallel) LU- or Cholesky-factorization, to determine which nodes to eliminate. The fill-in reducing properties of these methods also guarantee sparsity of the reduced network. Similar ideas have also been used in [77, 89].

Our main motivation for this approach is that large resistor networks in ESD typically are extracted networks with a structure that is related to the underlying (interconnect) layout. Unfortunately, the extracted networks are usually produced by extraction software of which the algorithms are unknown, and hence the structure of the extracted network is difficult to recover. Standard tools from graph theory, however, can be used to recover at least part of the structure.

Our approach can be summarized as follows:

1. The first step is to compute the strongly connected components [61] of the network. The presence of strongly connected components is very natural in extracted networks: a piece of interconnect connecting two other elements such as diodes or transistors, for instance, results in an extracted network with two terminals, disconnected from the rest of the extracted circuit. By splitting the network into connected components, we have simplified the problem of reduction because we can deal with the connected components one by one.
2. The second step is to selectively eliminate internal nodes in the individual connected components. For resistor networks, this can be done using the Schur complement [67], and no approximation error is made. The key here is that those internal nodes are eliminated that give the least fill-in. First, (Constrained) AMD [62] is used to reorder the unknowns such that the terminal nodes will be among the last to eliminate. To find the optimal reduction, internal nodes are eliminated one-by-one in the order computed by AMD, while keeping track of the reduced system with fewest resistors.

   Since the ordering is chosen to minimize fill-in, the resulting reduced matrix is sparse. Note that all operations are exact, i.e., we do not make any approximations. As a result, the path resistances between external nodes remain equal to the path resistances in the original network.
3. Finally, the reduced conductance matrix can be realized as a reduced resistor network that is equivalent to the original network. This is done easily by unstamping the values in the G matrix intro the corresponding resistor values and their node connections in the netlist [69]. Since the number of resistors (and number of nodes) is smaller than in the original network, also the resulting netlist is smaller in size.

An additional reduction could be obtained by removing relatively large resistors from the resulting reduced network. However, this will introduce an approximation error that might be hard to control a priori, since no sharp upper bounds on the error are available [87]. Another issue that is subject to further research is that the optimal ratio of number of (internal) nodes to resistors (sparsity) may also depend on the ratio of number of external to internal nodes, and on the type of simulation that will be done with the network.

In the following sections we will describe how strongly connected components and fill-in minimizing reorderings can be used for the reduction of RC networks as well.

6.2.2 Reduction of RC Networks

This section presents the developments for RC netlist reduction, first by reviewing an existing approach called PACT (Pole Analysis via Congruence Transformations). Then, graph-based tools are brought in to enhance sparsity preservation with the novel reduction method, SparseMA (Sparse Modal Approximation).

Following the problem description in [73], consider the modified nodal analysis (MNA) description of an input *impedance* type RC circuit, driven by input currents:

$$(\mathbf{G} + s\mathbf{C})\mathbf{x}(s) = \mathbf{B}\mathbf{u}(s), \tag{6.45}$$

where $\mathbf{x}$ denote the node voltages, and $\mathbf{u}$ represent the currents injected into the terminals (also called ports or external nodes). The number of internal nodes is n, and the number of terminals is p, thus $\mathbf{G} \in \mathbb{R}^{(p+n)\times(p+n)}$, $\mathbf{C} \in \mathbb{R}^{(p+n)\times(p+n)}$ and $\mathbf{B} \in \mathbb{R}^{(p+n)\times p}$. A natural choice for the system outputs are the voltage drops at the terminal nodes, i. e., $\mathbf{y}(s) = \mathbf{B}^T\mathbf{x}(s)$. Thus the transfer function of (6.45) is the input impedance:

$$\mathbf{Z}(s) = \frac{\mathbf{y}(s)}{\mathbf{u}(s)} = \mathbf{B}^T(\mathbf{G} + s\mathbf{C})^{-1}\mathbf{B}. \tag{6.46}$$

Modal approximation is a method to reduce (6.45), by preserving its most dominant eigenmodes. The dominant eigenmodes are a subset of the poles of $\mathbf{Z}(s)$ (i. e. of the generalized eigenvalues $\Lambda(-\mathbf{G}, \mathbf{C})$) and can be computed using specialized eigenvalue solvers (SADPA [80] or SAMDP [82, 85]). For the complete discussion on modal approximation and its implementation we refer to [80, 81, 85]. Here, we emphasize that applying modal approximation to reduce (6.45) directly is unsuitable especially if the underlying RC circuit has many terminals (inputs). This is because modal approximation does not preserve the structure of $\mathbf{B}$ and $\mathbf{B}^T$ during reduction (for ease of understanding we denote the input-output structure loss as *non-preservation of terminals*) [69]. Modeling the input-output connectivity of the reduced model would require synthesis via controlled sources at the circuit terminals, and furthermore would connect all terminals with one-another [69]. In this chapter we present several alternatives for reducing RC netlists where not only the terminals are preserved, but also the sparsity of the reduced models.

Grouping the node voltages so that $\mathbf{x}_P \in \mathbb{R}^p$ are the voltages measured at the terminal nodes (ports), and $\mathbf{x}_I \in \mathbb{R}^n$ are the voltages at the internal nodes, we can partition (6.45) as follows:

$$\left(\begin{bmatrix} \mathbf{G}_P & \mathbf{G}_C^T \\ \mathbf{G}_C & \mathbf{G}_I \end{bmatrix} + s \begin{bmatrix} \mathbf{C}_P & \mathbf{C}_C^T \\ \mathbf{C}_C & \mathbf{C}_I \end{bmatrix}\right) \begin{bmatrix} \mathbf{x}_P \\ \mathbf{x}_I \end{bmatrix} = \begin{bmatrix} \mathbf{B}_P \\ \mathbf{0} \end{bmatrix} \mathbf{u}. \tag{6.47}$$

Since no current is injected into internal nodes, the non-zero contribution from the input is $\mathbf{B}_P \in \mathbb{R}^{(p \times p)}$. Eliminating $\mathbf{x}_I$, system (6.47) is equivalent to:

$$\underbrace{[(\mathbf{G}_P + s\mathbf{C}_P)}_{\mathbf{Y}_P(s)} - \underbrace{(\mathbf{G}_C + s\mathbf{C}_C)^T (\mathbf{G}_I + s\mathbf{C}_I)^{-1} (\mathbf{G}_C + s\mathbf{C}_C)]}_{\mathbf{Y}_I(s)} \mathbf{x}_P = \mathbf{B}_P \mathbf{u} \tag{6.48}$$

$$\mathbf{Y}(s) = \mathbf{Y}_P(s) - \mathbf{Y}_I(s) \tag{6.49}$$

In (6.48) the matrix blocks $(\mathbf{G}_P + s\mathbf{C}_P)$ corresponding to the circuit terminals are isolated. Applying modal approximation on $\mathbf{Y}_I(s)$ would reduce the system and preserve the location of the terminals. This would involve for instance computing the dominant eigenmodes of $(-\mathbf{G}_I, \mathbf{C}_I)$ via a variant of SAMDP (called here *frequency dependent SAMDP*, because the input-output matrices $(\mathbf{G}_C + s\mathbf{C}_C)$ depend on the frequency s). We have implemented this approach, but it turns out that a large number of dominant eigenmodes of $(-\mathbf{G}_I, \mathbf{C}_I)$ would be needed to capture the DC and offset of the full system $\mathbf{Y}(s)$. Instead, two alternatives are presented that improve the quality of the approximation: an existing method called *PACT (Pole Analysis via Congruence Transformations)* [73] and a novel graph-based reduction called *SparseMA (Sparse Modal Approximation)*.

6.2.2.1 Existing Method: PACT

In [73] the authors propose to capture the DC and offset of $\mathbf{Y}(s)$ via a congruence transformation which reveals the first two moments of $\mathbf{Y}(s)$ as follows. Since $\mathbf{G}_I$ is symmetric positive definite, the Cholesky factorization $\mathbf{L}\mathbf{L}^T = \mathbf{G}_I$ exists. Using the following congruence transformation:

$$\mathbf{X} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{G}_I^{-1}\mathbf{G}_C & \mathbf{L}^{-T} \end{bmatrix}, \ \mathbf{G}' = \mathbf{X}^T\mathbf{G}\mathbf{X} = \begin{bmatrix} \mathbf{G}'_P & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \ \mathbf{C}' = \mathbf{X}^T\mathbf{C}\mathbf{X} = \begin{bmatrix} \mathbf{C}'_P & \mathbf{C}'^T_C \\ \mathbf{C}'_C & \mathbf{C}'_I \end{bmatrix} \tag{6.50}$$

Eqs. (6.48) and (6.49) are rewritten as:

$$\underbrace{[(\mathbf{G}'_P + s\mathbf{C}'_P)}_{\mathbf{Y}'_P(s)} - \underbrace{s^2\mathbf{C}'^T_C(\mathbf{I} + s\mathbf{C}'_I)^{-1}\mathbf{C}'_C]}_{\mathbf{Y}'_I(s)} \mathbf{x}'_P = \mathbf{B}_P\mathbf{u} \tag{6.51}$$

$$\mathbf{Y}'(s) = \mathbf{Y}'_P(s) - \mathbf{Y}'_I(s), \tag{6.52}$$

where:

$$\mathbf{G}'_P = \mathbf{G}_P - \mathbf{G}_C^T\mathbf{M}, \quad \mathbf{M} = \mathbf{G}_I^{-1}\mathbf{G}_C \tag{6.53}$$

$$\mathbf{C}'_P = \mathbf{C}_P - \mathbf{N}^T\mathbf{M} - \mathbf{M}^T\mathbf{C}_C, \quad \mathbf{N} = \mathbf{C}_C - \mathbf{C}_I\mathbf{M} \tag{6.54}$$

$$\mathbf{C}'_C = \mathbf{L}^{-1}\mathbf{N}, \quad \mathbf{C}'_I = \mathbf{L}^{-1}\mathbf{C}_I\mathbf{L}^{-T}. \tag{6.55}$$

In (6.51), the term $\mathbf{Y}'_P(s)$ captures the first two moments of $\mathbf{Y}'(s)$ and is preserved in the reduced model. The reduction is performed on $\mathbf{Y}'_I(s)$ only. In [73] this is done via modal approximation as described next. Using the symmetric eigendecomposition $\mathbf{C}'_I = \mathbf{U}\Lambda'_I\mathbf{U}^T$, $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, the system matrices (6.50) are block diagonalized as follows:

$$\mathbf{X}' = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{U} \end{bmatrix}, \quad \mathbf{G}'' = \mathbf{X}'^T\mathbf{G}'\mathbf{X}' = \begin{bmatrix} \mathbf{G}'_P & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \mathbf{G}' \tag{6.56}$$

$$\mathbf{C}'' = \mathbf{X}'^T\mathbf{C}'\mathbf{X}' = \begin{bmatrix} \mathbf{C}'_P & \mathbf{C}'^T_C\mathbf{U} \\ \mathbf{U}^T\mathbf{C}'_C & \mathbf{U}^T\mathbf{C}'_I\mathbf{U} \end{bmatrix} = \begin{bmatrix} \mathbf{C}'_P & \mathbf{C}''^T_C \\ \mathbf{C}''_C & \Lambda'_I \end{bmatrix} \tag{6.57}$$

$$\mathbf{Y}''(s) = \mathbf{Y}'_P(s) - s^2[\mathbf{C}''^T_C(\mathbf{I} + s\Lambda'_I)^{-1}\mathbf{C}''_C] \tag{6.58}$$

The reduced model is obtained by selecting only k of the n eigenvalues from Λ'_I:

$$\mathbf{Y}''_k(s) = \mathbf{Y}'_P(s) - s^2\sum_{i=1}^{k}\frac{\mathbf{r}_i^T\mathbf{r}_i}{1 + s\lambda'_i} \ , \ \mathbf{r}_i^T = \mathbf{C}'^T_C\mathbf{U}_{[:,1:k]}, \ \ \lambda'_i = \Lambda'_{I\,[i,i]}. \tag{6.59}$$

In [73], a selection criterion for λ'_i, $i = 1\ldots k$ is proposed, based on a user-specified error and a maximum frequency. These eigenmodes are computed in [73] via the Lanczos algorithm. The criterion proposed in [81, 85] can also be used to compute the dominant eigenmodes λ'_i via SAMDP.

The advantage of the PACT reduction method is the preservation of the first two moments of $\mathbf{Y}(s)$ in $\mathbf{Y}'_P(s)$. This ensures that the DC and offset of the response is approximated well in the reduced model. The main costs of such an approach are: (1) performing a Cholesky factorization of $\mathbf{C}_I$ (which becomes expensive when n is very large, (2) solving an eigenvalue problem from a dense $\mathbf{C}'_I$ matrix and, most importantly, (3) the fill-in in the port block matrices $\mathbf{G}'_P$, $\mathbf{C}'_P$ and in $\mathbf{C}'_C$. It turns out that (2) can be solved more efficiently by keeping $\mathbf{C}'_I$ as a product of sparse matrices during computation, and will be addressed elsewhere. Avoiding problems (1) and (3) however require new strategies to improve sparsity, and are presented in Sect. 6.2.2.2. The fill-in introduced in $\mathbf{G}'_P$, $\mathbf{C}'_P$ becomes especially important for RC netlists with many terminals [$p \sim O(10^3)$]. Compared to the original model where the port blocks $\mathbf{G}_P$ and $\mathbf{C}_P$ were sparse, the dense $\mathbf{G}'_P$, $\mathbf{C}'_P$ will yield many R and C components during synthesis, resulting in a reduced netlist where almost all the nodes are interconnected. Simulating such netlists might require longer time measures than the original circuit simulation, hence sparser reduced models (and netlists) are desired. Next, we present several ideas for improving the sparsity of RC reduced models via a combination of tools including: netlist partitioning, graph-based node reordering strategies, and efficient algorithms for modal approximation.

6.2.2.2 Improved Graph-Based Method: SparseMA

In this section we present an improved model reduction method for RC circuits, which overcomes the disadvantages of PACT: it requires no matrix factorizations prior to reduction, performs all numerical computations on sparse matrices, and most importantly, preserves the sparsity of the matrix blocks corresponding to the external nodes. The method is called *sparse modal approximation (SparseMA)* and uses tools from graph theory to identify a partitioning and reordering of nodes that, when applied prior to the model reduction step, can significantly improve the sparsity of the reduced model.

The idea is to reorder the nodes in the RC netlist so that some of the internal nodes (m) are promoted as external nodes, together with the circuit terminals (p). We will denote as *selected nodes* the collection of $p + m$ terminals and promoted internal nodes. The $n - m$ internal nodes are the *remaining nodes*. Supposing one has already identified such a partitioning of nodes, the following structure is revealed, where without loss of generality we assume the selected nodes appear in the border of the **G** and **C** matrices:

$$\left(\begin{bmatrix} \mathbf{G}_R & \mathbf{G}_K \\ \mathbf{G}_K^T & \mathbf{G}_S \end{bmatrix} + s \begin{bmatrix} \mathbf{C}_R & \mathbf{C}_K \\ \mathbf{C}_K^T & \mathbf{C}_S \end{bmatrix}\right) \begin{bmatrix} \mathbf{x}_R \\ \mathbf{x}_S \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_S \end{bmatrix} \mathbf{u}. \tag{6.60}$$

Note that in $\mathbf{B}_S$ the rows corresponding to the promoted m internal nodes are still zero. Similarly to (6.48), the admittance is expressed as:

$$\underbrace{[(\mathbf{G}_S + s\mathbf{C}_S)}_{\mathbf{Y}_S(s)} - \underbrace{(\mathbf{G}_K + s\mathbf{C}_K)^T (\mathbf{G}_R + s\mathbf{C}_R)^{-1} (\mathbf{G}_K + s\mathbf{C}_K)]}_{\mathbf{Y}_R(s)} \mathbf{x}_S = \mathbf{B}_S \mathbf{u} \tag{6.61}$$

$$\mathbf{Y}(s) = \mathbf{Y}_S(s) - \mathbf{Y}_R(s). \tag{6.62}$$

Recall that reducing $\mathbf{Y}_I(s)$ directly from the simple partitioning (6.47) and (6.48) is not a method of choice, because by preserving $\mathbf{Y}_P(s)$ only, the DC and offset of $\mathbf{Y}(s)$ would not be accurately matched. Using instead the improved partitioning (6.60) and (6.61), one aims at better approximating the DC and offset of $\mathbf{Y}(s)$ by preserving $\mathbf{Y}_S(s)$ (which now encaptures not only the external nodes but also a subset of the internal nodes). Finding the partitioning (6.60) only requires a reordering of nodes, thus no Cholesky factorization or fill-introducing congruence transformation is needed prior to the MOR step. One can reduce $\mathbf{Y}_R(s)$ directly with modal approximation (via frequency dependent SAMDP), and preserve the sparsity of the extended port blocks from $\mathbf{Y}_S(s)$.

By interpolating k dominant eigenmodes from the symmetric eigendecoposition $[\Lambda_R, \mathbf{V}] = eig(-\mathbf{G}_R, \mathbf{C}_R)$, the reduced model is obtained:

$$\mathbf{Y}_k(s) = \mathbf{Y}_S(s) - \sum_{i=1}^{k} \frac{\mathbf{q}_i^T \mathbf{q}_i}{1 + s\lambda_i} \;, \;\; \mathbf{q}_i^T = (\mathbf{G}_K + s\mathbf{C}_K)^T \mathbf{V}_{[:,1:k]}, \;\; \lambda_i = \Lambda_{R[i,i]}. \tag{6.63}$$

In matrix terms, the reduced model is easily constructed by re-connecting the preserved selected matrix blocks to the reduced blocks:

$$\left(\begin{bmatrix} \hat{\mathbf{G}}_R & \hat{\mathbf{G}}_K \\ \hat{\mathbf{G}}_K^T & \mathbf{G}_S \end{bmatrix} + s \begin{bmatrix} \hat{\mathbf{C}}_R & \hat{\mathbf{C}}_K \\ \hat{\mathbf{C}}_K^T & \mathbf{C}_S \end{bmatrix} \right) \begin{bmatrix} \hat{\mathbf{x}}_R \\ \mathbf{x}_S \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_S \end{bmatrix} \mathbf{u}, \tag{6.64}$$

where:

$$\hat{\mathbf{G}}_R = \mathbf{V}_{[:,1:k]}^T \mathbf{G}_R \mathbf{V}_{[:,1:k]} \rightarrow \text{diagonal}, \;\; \hat{\mathbf{G}}_K = \mathbf{V}_{[:,1:k]}^T \mathbf{G}_K, \;\; \mathbf{G}_S \rightarrow \text{sparse} \tag{6.65}$$

$$\hat{\mathbf{C}}_R = \mathbf{V}_{[:,1:k]}^T \mathbf{C}_R \mathbf{V}_{[:,1:k]} \rightarrow \text{diagonal}, \;\; \hat{\mathbf{C}}_K = \mathbf{V}_{[:,1:k]}^T \mathbf{C}_K, \;\; \mathbf{C}_S \rightarrow \text{sparse}. \tag{6.66}$$

The remaining problem is how to determine the selected nodes and the partitioning (6.60). Inspired from the results obtained for R networks, we propose to first find the permutation $\mathbf{P}$ which identifies the strongly connected components (sccs) of $\mathbf{G}$. Both $\mathbf{G}$ and $\mathbf{C}$ are reordered according to $\mathbf{P}$, revealing the structure (6.60). With this permutation, the circuit terminals are redistributed according to the sccs of $\mathbf{G}$, and several clusters of nodes can be identified: a large component consisting of internal nodes and very few (or no) terminals, and clusters formed each by internal nodes plus some terminals. We propose to leave all clusters consisting of internal nodes and terminals intact, and denote these nodes as the *selected nodes* mentioned above. If there are still terminals outside these clusters, they are added to these selected nodes and complete the blocks $\mathbf{G}_S$, $\mathbf{C}_S$. The remaining cluster of internal nodes forms $\mathbf{G}_R$ and $\mathbf{C}_R$. The model reduction step is performed on $\mathbf{G}_R$ and $\mathbf{C}_R$ (and implicitly on $\mathbf{G}_K$ and $\mathbf{C}_K$). We also note that matrices $\mathbf{G}_K$ and $\mathbf{C}_k$ resulting from this partitioning usually have many zero columns, thus $\hat{\mathbf{G}}_K$ and $\hat{\mathbf{C}}_K$ will preserve these zero columns.

The procedure is illustrated in Sect. 6.2.3 through a medium-sized example. Larger netlists can be treated via a similar reordering and partitioning strategy, possibly in a recursive manner (for instance when after an initial reordering the number of selected nodes is too large, the same partitioning strategy could be re-applied to $\mathbf{G}_S$ and $\mathbf{C}_S$ and further reduce these blocks). Certainly, other reorderings of **G** and **C** could be exploited, for instance according to a permutation which identifies the sccs of **C** instead of **G**. The choice for either using **G** or **C** to determine the permutation **P** is made according to the structure of the underlying system and may depend on the application. We also emphasize that the reduced models for both PACT and SparseMA are passive [74] and therefore also stable. Passivity is ensured by the fact that all transformations applied throughout are congruence transformations on symmetric positive definite matrices, thus the reduced system matrices remain symmetric positive definite.

6.2.3 Numerical Results

The graph-based reduction procedures were applied on several networks resulting from parasitic extraction. We present results for both R and RC networks.

6.2.3.1 *R* Network Reduction

Table 6.2 shows results for three resistor networks of realistic interconnect layouts. The number of nodes is reduced by a factor > 10 and the number of resistors by a factor > 3. As a result, the computing time for calculating path resistances in the original network (including nonlinear elements such as diodes) is 10 times smaller.

6.2.3.2 *RC* Network Reduction

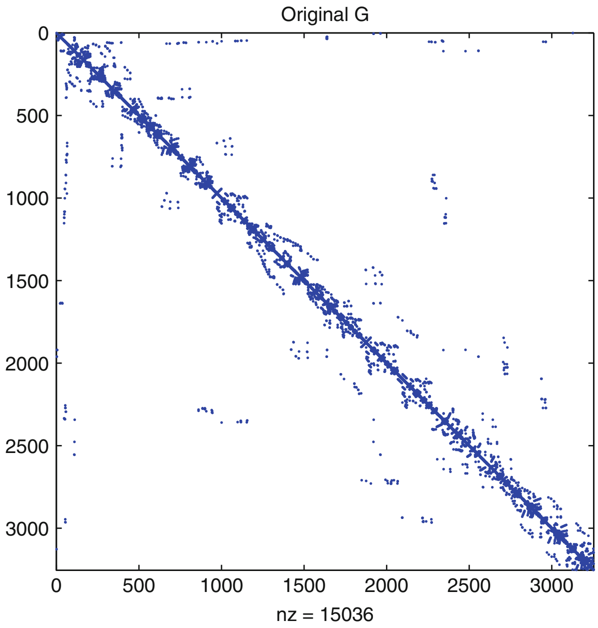
We reduce an RC netlist with $n = 3{,}231$ internal nodes and $p = 22$ terminals (external nodes). The structure of the original **G** and **C** matrices is shown in Figs. 6.7 and 6.8, where the $p = 22$ terminals correspond to their first 22 rows and columns.

The permutation revealing the strongly connected components of **G** reorders the matrices as shown in Figs. 6.9 and 6.10. The reordering is especially visible in the "arrow-form" capacitance matrix. There, the $p = 22$ terminal nodes together with

Table 6.2 Results of reduction algorithm

	Network I		Network II		Network III	
	Original	Reduced	Original	Reduced	Original	Reduced
#external nodes	274		3,399		1,978	
#internal nodes	5,558	516	99,112	6,012	101,571	1,902
#resistors	8,997	1,505	161,183	62,685	164,213	39,011
CPU time	10 s	1 s	67 h	7 h	20 h	2 h
Speed up	10×		9.5×		10×	

Fig. 6.7 Original **G** matrix



$m = 40$ internal nodes are promoted to the border, revealing the 62 selected nodes that will be preserved in the reduced model (i.e. the $\mathbf{G}_S$ and $\mathbf{C}_S$ blocks in (6.60)). The first $n - m = 3{,}191$ nodes are the remaining internal nodes and form the $\mathbf{G}_R$ and $\mathbf{C}_R$ blocks in (6.60). The $\mathbf{G}_K$ block has only 1 non-zero column, and also in $\mathbf{C}_K$ many zero columns can be identified.

The reduced SparseMA model is obtained according to (6.63) and (6.64) and is shown in Figs. 6.11 and 6.12. The internal blocks $\mathbf{G}_R$ and $\mathbf{C}_R$ were reduced from dimension 3,191 to $\hat{\mathbf{G}}_R$ and $\hat{\mathbf{C}}_R$ of dimension $k = 7$, by interpolating the 7 most dominant eigenmodes of $[\Lambda_R, \mathbf{V}] = eig(-\mathbf{G}_R, \mathbf{C}_R)$. Note that $\hat{\mathbf{G}}_R$ and $\hat{\mathbf{C}}_R$ are diagonal. The selected 62 nodes corresponding to the $\mathbf{G}_S$ and $\mathbf{C}_S$ blocks are preserved, evidently preserving sparsity. The only fill-in introduced by the proposed reduction procedure is in the non-zero columns of $\hat{\mathbf{G}}_K$ and $\hat{\mathbf{C}}_K$. It is worth noticing that $\hat{\mathbf{G}}_K$ only has 1 non-zero column, thus remains sparse.

The sparsity structure of the PACT reduced model (6.59) is shown in Figs. 6.13 and 6.14. The blocks corresponding to the first 22 nodes (the preserved external nodes) are full, as are the capacitive connection blocks to the reduced internal part. Only the reduced internal blocks remain sparse (diagonal).

Aside from sparsity preservation, one is interested in the quality of the approximation for the reduced model. In Fig. 6.15, we show that the SparseMA model accurately matches the original response for a wide frequency range (1 Hz → 10 THz). The Pstar [78] simulations of the synthesized model are identical to the Matlab simulations (the synthesized model was obtained via the RLCSYN unstamping procedure [71, 87]). In Fig. 6.16, the relative errors between the

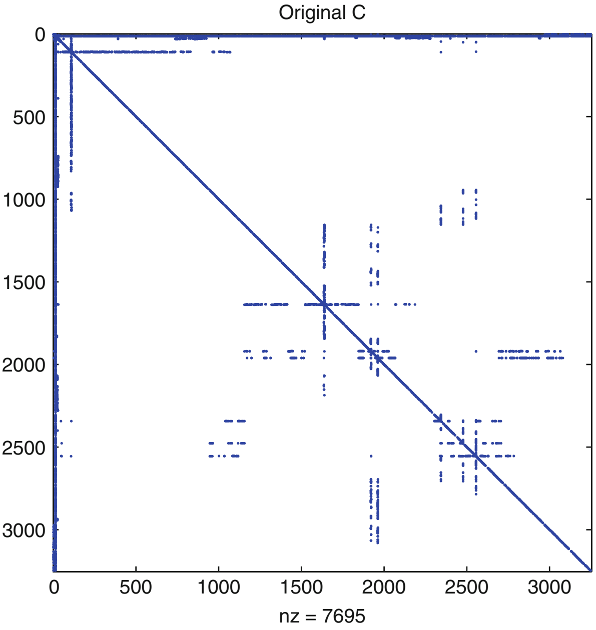


Fig. 6.8 Original **C** matrix

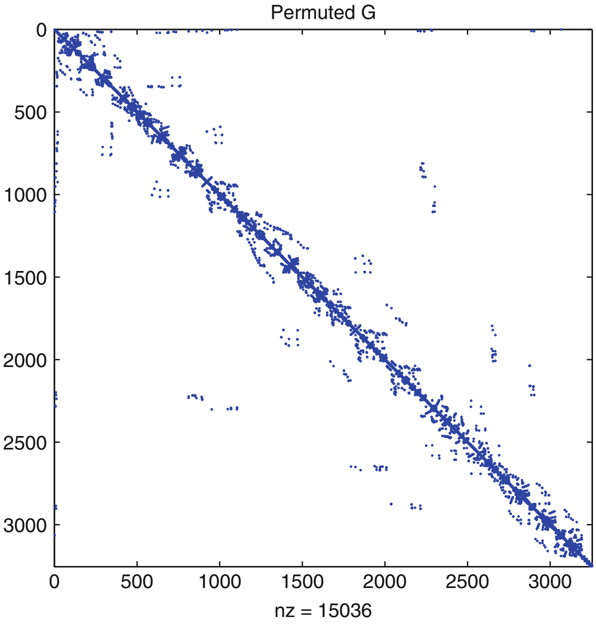


Fig. 6.9 Permuted **G** according to scc(**G**)

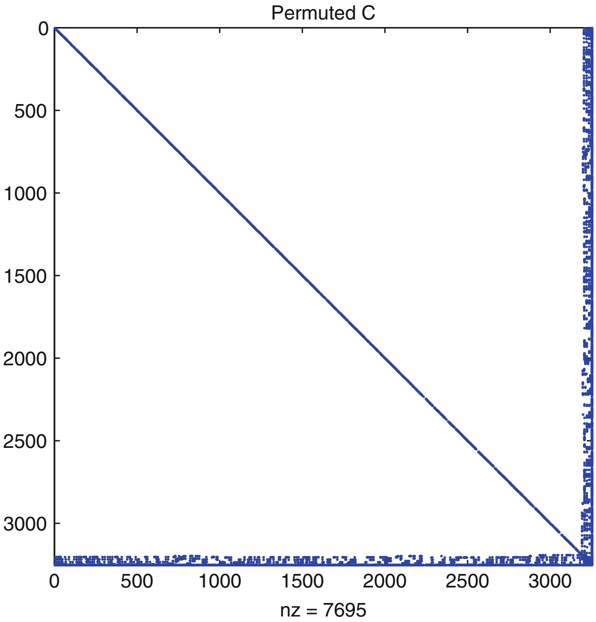


Fig. 6.10 Permuted **C** according to scc(**G**)

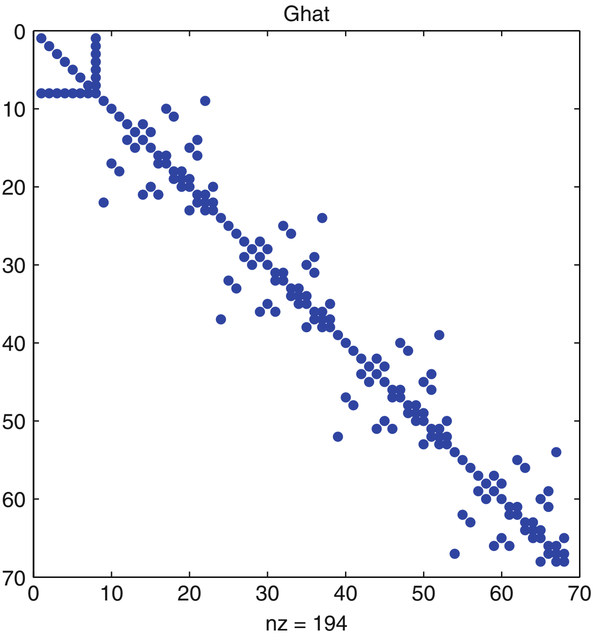


Fig. 6.11 Reduced **G** matrix with Sparse MA

Fig. 6.12 Reduced **C** matrix with Sparse MA

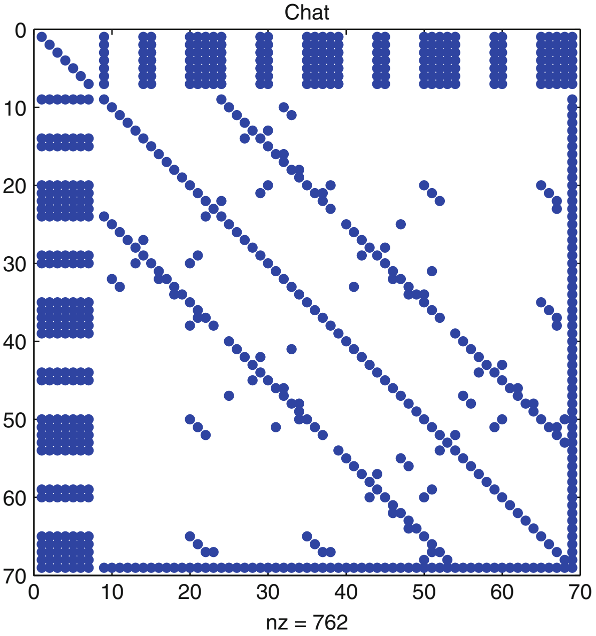
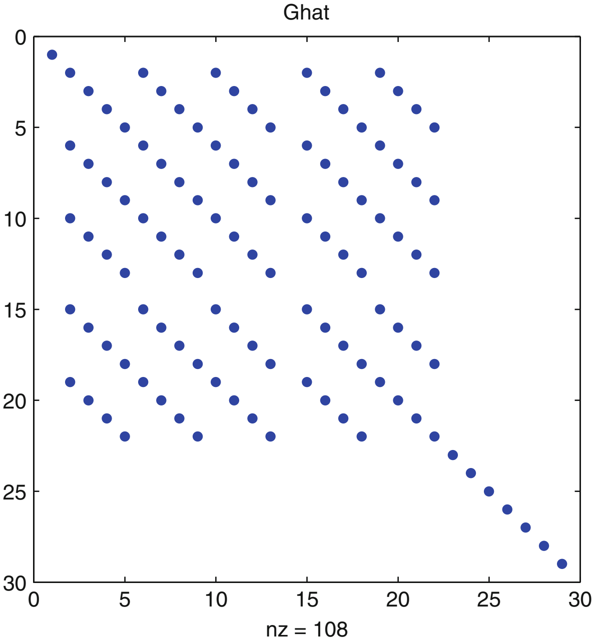


Fig. 6.13 Reduced **G** matrix with PACT



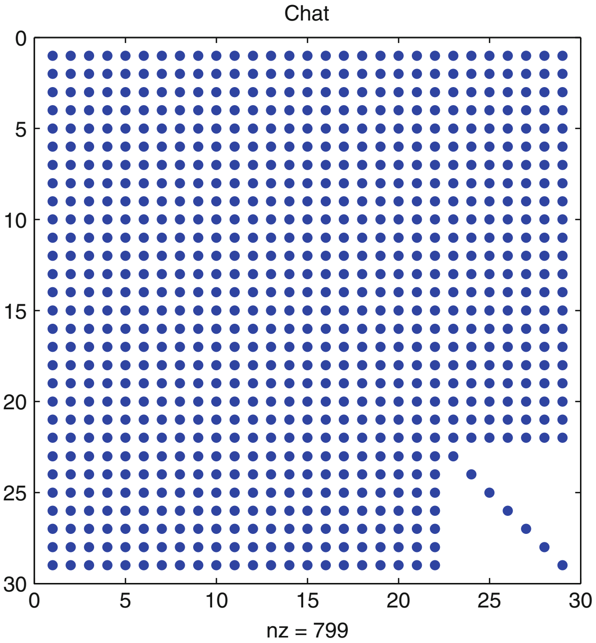


Fig. 6.14 Reduced **C** matrix with PACT

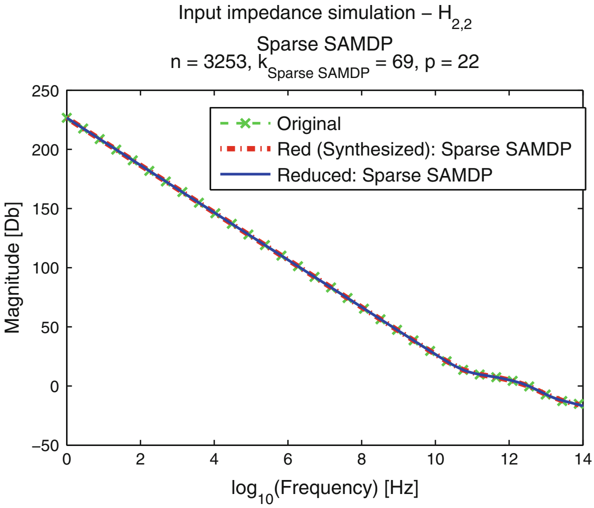


Fig. 6.15 AC simulation 1: original, reduced (Sparse MA) and synthesized model

original model and three reduced models are presented: SparseMA, PACT and the commercial software Jivaro [64]. The SparseMA model is the most accurate for the entire frequency range.

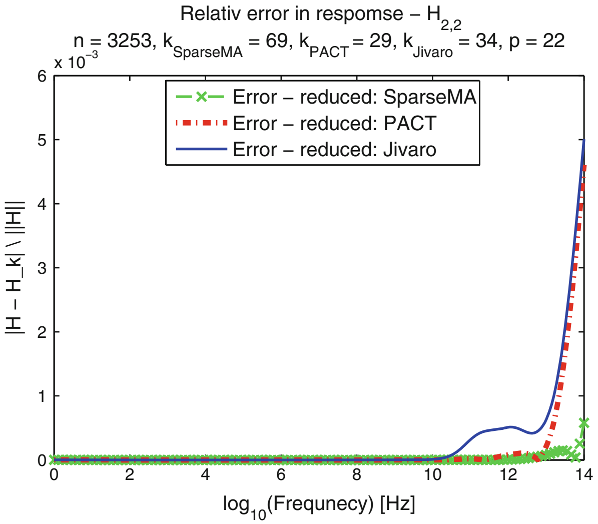


Fig. 6.16 AC simulation 1: relative error between original and reduced models (SparseMA, Pact, Jivaro)

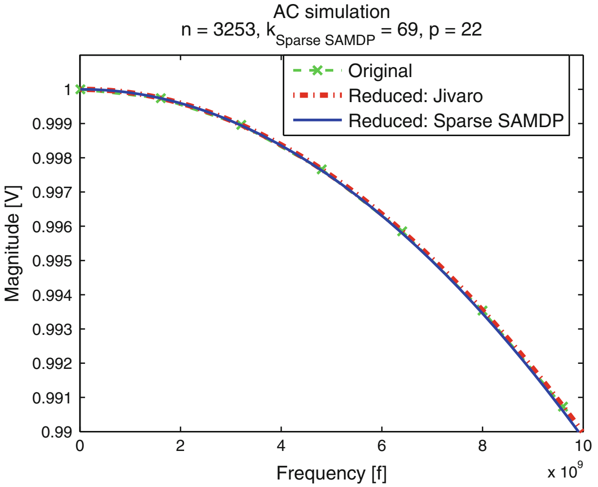


Fig. 6.17 AC simulation 2: original, reduced (Sparse MA) and reduced (Jivaro)

Figure 6.17 shows a different AC circuit simulation, where the SparseMA model performs comparably to the reduced model obtained with the commercial software Jivaro [64]. Finally, the transient simulation in Fig. 6.18 confirms that the SparseMA model is both accurate and stable.

Table 6.3 shows the reduction results for the RC network. For the 3 reduced models: SparseMA, PACT and Jivaro we assess the effect of the reduction by means

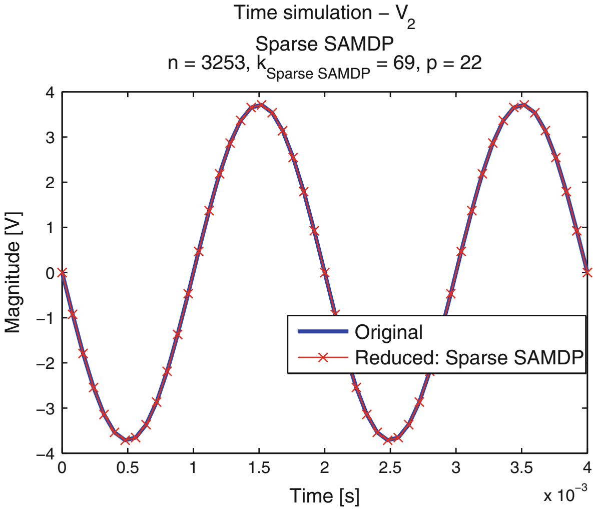


Fig. 6.18 Transient simulation 1: all external nodes grounded and voltage measured at node 2. Original and reduced (Sparse MA – synthesized)

Table 6.3 Results with SparseMA reduction on RC netlist

	Original	Red. SparseMA	Red. PACT	Red. Jivaro
#external nodes	22			
#internal nodes	3,231	47	7	12
#unknowns	3,253	69	29	34
#resistors	7,944	78	68	28
#capacitors	3,466	383	414	97
$\frac{\text{\#elements}}{\text{\#int. nodes}}$	3.53	9.8	68.8	10.4
$\frac{\text{\#elements}}{\text{\#unknowns}}$	3.5	6.7	16.6	3.67
CPU time	6.8 s	0.1 s	0.06	0.02 s
Speed up		68×	113×	340×

of several factors. With all methods, both the number of nodes and the number of circuit elements was reduced significantly, resulting in at least 68x speed-up in AC simulation time. It should be noted that the SparseMA model and the Jivaro model have lower ratios of $\frac{\#elements}{\#unknowns}$ and $\frac{\#elements}{\#int.nodes}$ than the PACT model. Even though the Jivaro and the PACT model are faster to simulate for this network, the SparseMA model gives a good trade-off between approximation quality, sparsity preservation and CPU speed-up. Recall that the matrix blocks corresponding to the circuit terminals become dense with PACT, but remain sparse with SparseMA. As for circuits with more terminals $\sim O(10^3)$ the corresponding matrix blocks become larger, preserving their sparsity via SparseMA is an additional advantage. Hence, the improvement on simulation time could be greater with SparseMA when applied on larger models with many terminals.

6.2.4 Concluding Remarks

New approaches were presented for reducing R and RC circuits with multi-terminals, using tools from graph theory. It was shown how netlist partitioning and node reordering strategies can be combined with existing model reduction techniques, to improve the sparsity of the reduced RC models and implicitly their simulation time. The proposed sparsity preserving method, SparseMA, performs comparably to the commercial tool Jivaro. Future work will investigate how similar strategies can be applied to RC models with many more terminals [$\sim O(10^3)$] and to *RLCk* netlists.

6.3 Simulation of Mutually Coupled Oscillators Using Nonlinear Phase Macromodels and Model Order Reduction Techniques

The design of modern RF (radio frequency) integrated circuits becomes increasingly more complicated due to the fact that more functionality needs to be integrated on a smaller physical area.[5] In the design process floor planning, i.e., determining the locations for the functional blocks, is one of the most challenging tasks. Modern RF chips for mobile devices, for instance, typically have an FM radio, Bluetooth, and GPS on one chip. These functionalities are implemented with Voltage Controlled Oscillators (VCOs), that are designed to oscillate at certain different frequencies. In the ideal case, the oscillators operate independently, i.e., they are not perturbed by each other or any signal other than their input signal. Practically speaking, however, the oscillators are influenced by unintended (parasitic) signals coming from other blocks (such as Power Amplifiers) or from other oscillators, via for instance (unintended) inductive coupling through the substrate. A possibly undesired consequence of the perturbation is that the oscillators lock to a frequency different than designed for, or show pulling, in which case the oscillators are perturbed from their free running orbit without locking.

The locking effect was first observed by the Dutch scientist Christian Huygens in the seventeenth century. He observed that pendulums of two nearby clocks hanging on the same wall after some time moved in unison [120] (in other words they

[5] Section 6.3 has been written by Davit Harutyunyan, Joost Rommes, E. Jan W. ter Maten and Wil H.A. Schilders.

locked to the same frequency). Similar effects occur also for electrical oscillators. When an oscillator is locked to a different frequency, it physically means that the frequency of the oscillator is changed and as a result the oscillator operates at the new frequency. In this case in the spectrum of the oscillator we will observe a single peak corresponding to the new frequency of the oscillator. Contrary to the locking case, frequency pulling occurs when the interfering frequency source is not strong enough to cause frequency locking (e.g. weak substrate coupling). In this case in the spectrum of the pulled oscillator we will observe several sidebands around the carrier frequency of the oscillator. In Sect. 6.3.9 we will discuss several practical examples of locking and pulling effects.

Oscillators appear in many physical systems and interaction between oscillators has been of interest in many applications. Our main motivation comes from the design of RF systems, where oscillators play an important role [95, 100, 107, 120] in, for instance, high-frequency Phase Locked Loops (PLLs). Oscillators are also used in the modeling of circadian rhythm mechanisms, one of the most fundamental physiological processes [91]. Another application area is the simulation of large-scale biochemical processes [114].

Although the use of oscillators is widely spread over several disciplines, their intrinsic nonlinear behavior is similar, and, moreover, the need for fast and accurate simulation of their dynamics is universal. These dynamics include changes in the frequency spectrum of the oscillator due to small noise signals (an effect known as jitter [100]), which may lead to pulling or locking of the oscillator to a different frequency and may cause the oscillator to malfunction. The main difficulty in simulating these effects is that both phase and amplitude dynamics are strongly nonlinear and spread over separated time scales [113]. Hence, accurate simulation requires very small time steps during time integration, resulting in unacceptable simulation times that block the design flow. Even if computationally feasible, transient simulation only gives limited understanding of the causes and mechanisms of the pulling and locking effects.

To some extent one can describe the relation between the locking range of an oscillator and the amplitude of the injected signal (these terms will be explained in more detail in Sect. 6.3.1). Adler [90] shows that this relation is linear, but it is now well known that this is only the case for small injection levels and that the modeling fails for higher injection levels [111]. Also other linearized modeling techniques [120] suffer, despite their simplicity, from the fact that they cannot model nonlinear effects such as injection locking [111, 127].

In this section we use the nonlinear phase macromodel introduced in [100] and further developed and analyzed in [104–106, 111, 113, 115, 116, 127]. Contrary to linear macromodels, the nonlinear phase macromodel is able to capture nonlinear effects such as injection locking. Moreover, since the macromodel replaces the original oscillator system by a single scalar equation, simulation times are decreased while the nonlinear oscillator effects can still be studied without loss of accuracy. One of the contributions of this paper is that we show how such macromodels can be used in industrial practice to predict the behavior of inductively coupled oscillators.

Returning to our motivation, during floor planning, it is of crucial importance that the blocks are located in such a way that the effects of any perturbing signals are minimized. A practical difficulty here is that transient simulation of the full system is very expensive and usually unfeasible during the early design stages. One way to get insight in the effects of inductive coupling and injected perturbation signals is to apply the phase shift analysis [100]. In this section we will explain how this technique can be used to estimate the effects for perturbed individual and coupled oscillators, and how this can be of help during floor planning. We will consider perturbations caused by oscillators and by other components such as balanced/unbalanced transformers (baluns).

In some applications to reduce clockskew (clocksignals becoming out of phase), for instance, oscillators can be coupled via transmission lines [102]. Since accurate models for transmission lines can be large, this may lead to increased simulation times. We show how model order reduction techniques [94, 96, 97, 124] can be used to decrease simulation times without unacceptable loss of accuracy.

The section is organized as follows. In Sect. 6.3.1 we summarize the phase noise theory. A practical oscillator model and an example application are described in Sect. 6.3.2. Inductively coupled oscillators are discussed in detail in Sect. 6.3.3. In Sect. 6.3.4 we give an overview of existing methods to model injection locking of individual and resistively/capacitively coupled oscillators. In Sect. 6.3.5 we consider small parameter variations for mutually coupled oscillators. In Sects. 6.3.6 and 6.3.7 we show how the phase noise theory can be used to analyze oscillator-balun coupling and oscillator-transmission line coupling, respectively. In Sect. 6.3.8 we give a brief introduction to model order reduction and present a Matlab script used in our implementations. Numerical results are presented in Sect. 6.3.9 and the conclusions are drawn in Sect. 6.3.10.

6.3.1 Phase Noise Analysis of Oscillator

A general free-running oscillator can be expressed as an autonomous system of differential (algebraic) equations:

$$\frac{\mathrm{d}\mathbf{q}(\mathbf{x})}{\mathrm{d}t} + \mathbf{j}(\mathbf{x}) = 0, \tag{6.67a}$$

$$\mathbf{x}(0) = \mathbf{x}(T), \tag{6.67b}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ are the state variables, T is the period of the free running oscillator, which is in general unknown, $\mathbf{q}, \mathbf{j} : \mathbb{R}^n \to \mathbb{R}^n$ are (nonlinear) functions describing the oscillator's behavior and n is the system size. The solution of (6.67) is called Periodic Steady State (PSS) and is denoted by $\mathbf{x}_{pss}$. Although finding the PSS solution can be an challenging task in itself, we will not discuss this in the present paper and refer the interested reader to, for example, [105, 108–110, 122, 123, 126].

A general oscillator under perturbation can be expressed as a system of differential equations

$$\frac{\mathrm{d}\mathbf{q}(\mathbf{x})}{\mathrm{d}t} + \mathbf{j}(\mathbf{x}) = \mathbf{b}(t), \tag{6.68}$$

where $\mathbf{b}(t) \in \mathbb{R}^n$ are perturbations to the free running oscillator. For small perturbations $\mathbf{b}(t)$ it can be shown [100] that the solution of (6.68) can be approximated by

$$\mathbf{x}_p(t) = \mathbf{x}_{pss}(t + \alpha(t)) + \mathbf{y}(t), \tag{6.69}$$

where $\mathbf{y}(t)$ is the orbital deviation and $\alpha(t) \in \mathbb{R}$ is the phase shift, which satisfies the following scalar nonlinear differential equation:

$$\dot{\alpha}(t) = \mathbf{V}^T(t + \alpha(t)) \cdot \mathbf{b}(t), \tag{6.70a}$$

$$\alpha(0) = 0, \tag{6.70b}$$

where $\mathbf{V}(t) \in \mathbb{R}^n$ is called Perturbation Projection Vector (PPV) of (6.68). It is a special projection vector of the perturbations and is computed based on Floquet theory [99, 100, 115]. The PPV is a periodic function with the same period as the oscillator and can efficiently be computed directly from the PPS solution, see for example [101]. Using this simple and numerically cheap method one can do many kinds of analysis for oscillators, e.g. injection locking, pulling, a priori estimate of the locking range [100, 111].

For small perturbations the orbital deviation $\mathbf{y}(t)$ can be ignored [100] and the response of the perturbed oscillator is computed by

$$\mathbf{x}_p(t) = \mathbf{x}_{pss}(t + \alpha(t)). \tag{6.71}$$

6.3.2 LC Oscillator

For many applications oscillators can be modeled as an LC tank with a nonlinear resistor as shown in Fig. 6.19. This circuit is governed by the following differential equations for the unknowns (v, i):

$$C\frac{\mathrm{d}v(t)}{\mathrm{d}t} + \frac{v(t)}{R} + i(t) + S\tanh(\frac{G_n}{S}v(t)) = b(t), \tag{6.72a}$$

$$L\frac{\mathrm{d}i(t)}{\mathrm{d}t} - v(t) = 0, \tag{6.72b}$$

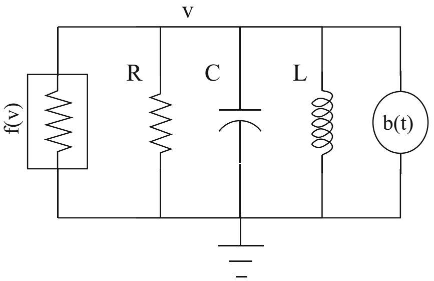


Fig. 6.19 Voltage controlled oscillator: current of the nonlinear resistor is given by $f(v) = S\tanh(\frac{G_n}{S}v(t))$

where C, L and R are the capacitance, inductance and resistance, respectively. The nodal voltage is denoted by v and the branch current of the inductor is denoted by i. The voltage controlled nonlinear resistor is defined by S and G_n parameters, where S has influence on the oscillation amplitude and G_n is the gain [111].

A lot of work [111, 120] has been done for the simulation of this type of oscillators. Here we will give an example that can be of practical use for designers. During the design process, early insight in the behavior of system components is of crucial importance. In particular, for perturbed oscillators it is very convenient to have a direct relationship between the injection amplitude and the side band level.

For the given RLC circuit with the following parameters $L = 930 \cdot 10^{-12}$ H, $C = 1.145 \cdot 10^{-12}$ F, $R = 1{,}000\,\Omega$, $S = 1/R$, $G_n = -1.1/R$ and injected signal $b(t) = A_{\mathrm{inj}}\sin(2\pi f)$, we plot the side band level of the voltage response versus the amplitude A_{inj} of the injected signal for different offset frequencies, see Fig. 6.20. The results in Fig. 6.20 can be seen as a simplified representation of Arnol'd tongues [98], that is helpful in engineering practice. We see, for instance, that the oscillator locks to a perturbation signal with an offset of 10 MHz if the corresponding amplitude is larger than $\sim\ 10^{-4}$ A (when the signal is locked the sideband level becomes 0 dB). This information is useful when designing the floor plan of a chip, since it may put additional requirements on the placement (and shielding) of components that generate, or are sensitive to, perturbing signals.

As an example, consider the floor plan in Fig. 6.21. The analysis described above and in Fig. 6.20 first helped to identify and quantify the unintended pulling and locking effects due to the coupling of the inductors (note that the potential causes (inductors) of pulling and locking effects first have to be identified; in practice, designers usually have an idea of potential coupling issues, for instance when there are multiple oscillators in a design). The outcome of this analysis indicated that there were unintended pulling effects in the original floorplan and hence some components were relocated (and shielded) to reduce unintended pulling effects. Finally, the same macromodels, but with different coupling factors due to the relocation of components, were used to verify the improved floorplan.

Although the LC tank model is relatively simple, it can be of high value especially in the early stages of the design process (schematic level), since it can be used to estimate the effects of perturbation and (unintended) coupling on the behavior of oscillators. As explained before, this may be of help during floor

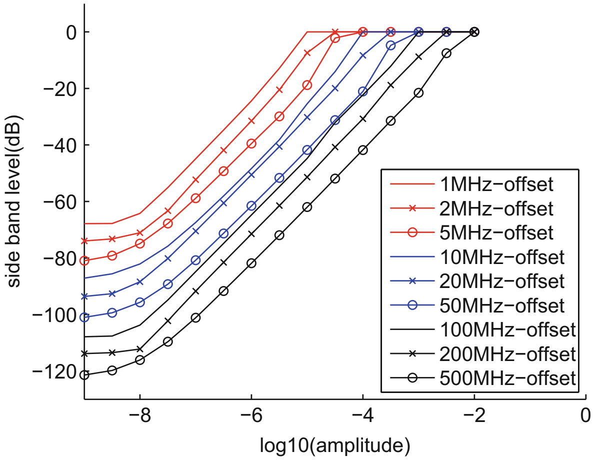


Fig. 6.20 Side band level of the voltage response versus the injected current amplitude for different offset frequencies

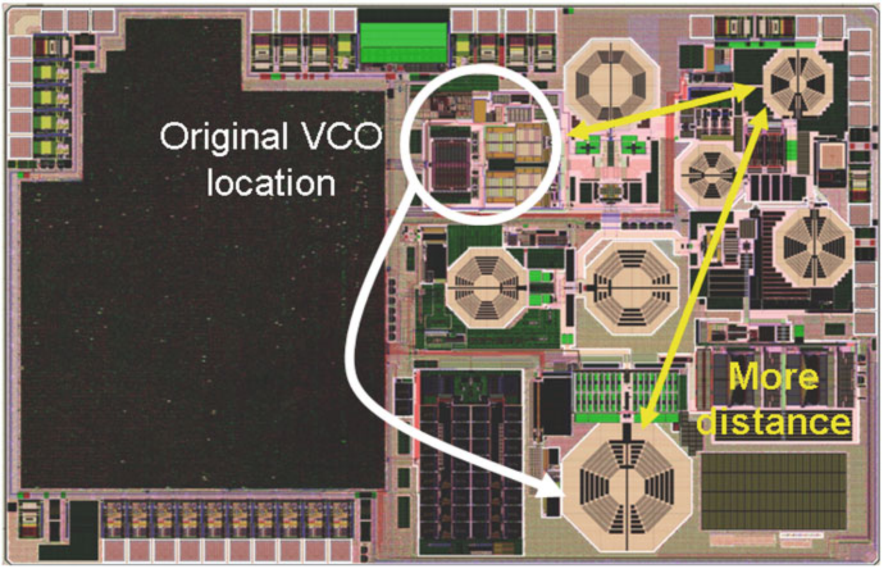


Fig. 6.21 Floor plan with relocation option that was considered after nonlinear phase noise analysis showed an intolerable pulling due to unintended coupling. Additionally, shielding was used to limit coupling effects even further

planning. In later stages, one typically validates the design via layout simulations, which can be much more complex due to the inclusion of parasitic elements. In general one has to deal with larger dynamical systems when parasitics are included, but the phase noise theory still applies. Therefore, in this paper we do not consider

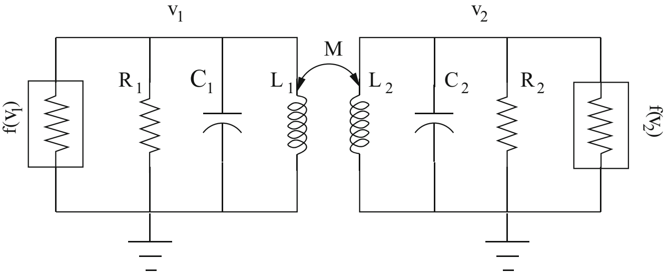


Fig. 6.22 Two inductively coupled LC oscillators

extracted parasitics. However, the values for L, C, R and coupling factors are typically based on measurement data and layout simulations of real designs.

6.3.3 *Mutual Inductive Coupling*

Next we consider the two mutually coupled LC oscillators shown in Fig. 6.22. The inductive coupling between these two oscillators can be modeled as

$$L_1 \frac{\mathrm{d}i_1(t)}{\mathrm{d}t} + M \frac{\mathrm{d}i_2(t)}{\mathrm{d}t} = v_1(t), \tag{6.73a}$$

$$L_2 \frac{\mathrm{d}i_2(t)}{\mathrm{d}t} + M \frac{\mathrm{d}i_1(t)}{\mathrm{d}t} = v_2(t), \tag{6.73b}$$

where $M = k\sqrt{L_1 L_2}$ is the mutual inductance and $|k| < 1$ is the coupling factor. This makes the matrix

$$\begin{pmatrix} L_1 & M \\ M & L_2 \end{pmatrix}$$

positive definite, which ensures that the problem is well posed. In this section all the parameters with a subindex refer to the parameters of the oscillator with the same subindex. If we combine the mathematical model (6.72) of each oscillator with (6.73), then the two inductively coupled oscillators can be described by the following differential equations

$$C_1 \frac{\mathrm{d}v_1(t)}{\mathrm{d}t} + \frac{v_1(t)}{R_1} + i_1(t) + S\tanh(\frac{G_n}{S} v_1(t)) = 0, \tag{6.74a}$$

$$L_1 \frac{\mathrm{d}i_1(t)}{\mathrm{d}t} - v_1(t) = -M \frac{\mathrm{d}i_2(t)}{\mathrm{d}t}, \tag{6.74b}$$

$$C_2\frac{\mathrm{d}v_2(t)}{\mathrm{d}t} + \frac{v_2(t)}{R_2} + i_2(t) + S\tanh(\frac{G_n}{S}v_2(t)) = 0, \tag{6.74c}$$

$$L_2\frac{\mathrm{d}i_2(t)}{\mathrm{d}t} - v_2(t) = -M\frac{\mathrm{d}i_1(t)}{\mathrm{d}t}. \tag{6.74d}$$

For small values of the coupling factor k the right-hand side of (6.74b) and (6.74d) can be considered as a small perturbation to the corresponding oscillator and we can apply the phase shift theory described in Sect. 6.3.1. Then we obtain the following simple nonlinear equations for the phase shift of each oscillator:

$$\dot{\alpha_1}(t) = \mathbf{V}_1^T(t+\alpha_1(t))\cdot\begin{pmatrix}0\\ -M\dfrac{\mathrm{d}i_2(t)}{\mathrm{d}t}\end{pmatrix}, \tag{6.75a}$$

$$\dot{\alpha_2}(t) = \mathbf{V}_2^T(t+\alpha_2(t))\cdot\begin{pmatrix}0\\ -M\dfrac{\mathrm{d}i_1(t)}{\mathrm{d}t}\end{pmatrix}, \tag{6.75b}$$

where the currents and voltages are evaluated by using (6.71):

$$[v_1(t),\, i_1(t)]^T = \mathbf{x}_{pss}^1(t+\alpha_1(t)), \tag{6.75c}$$

$$[v_2(t),\, i_2(t)]^T = \mathbf{x}_{pss}^2(t+\alpha_2(t)). \tag{6.75d}$$

Small parameter variations have also been studied in the literature by Volterra analysis, see e.g. [92, 93].

6.3.3.1 Time Discretization

The system (6.75) is solved by using implicit backward Euler for the time discretization and the Newton method is applied for the solution of the resulting two dimensional nonlinear equations (6.76a) and (6.76b), i.e.

$$\alpha_1^{m+1} = \alpha_1^m + \tau\mathbf{V}_1^T(t^{m+1}+\alpha_1^{m+1})\cdot \begin{pmatrix}0\\ -M\dfrac{i_2(t^{m+1})-i_2(t^m)}{\tau}\end{pmatrix}, \tag{6.76a}$$

$$\alpha_2^{m+1} = \alpha_2^m + \tau\mathbf{V}_2^T(t^{m+1}+\alpha_2^{m+1})\cdot \begin{pmatrix}0\\ -M\dfrac{i_1(t^{m+1})-i_1(t^m)}{\tau}\end{pmatrix}, \tag{6.76b}$$

$$[v_1(t^{m+1}),\, i_1(t^{m+1})]^T = \mathbf{x}_{pss}^1(t^{m+1}+\alpha_1^{m+1}), \tag{6.76c}$$

$$[v_2(t^{m+1}),\, i_2(t^{m+1})]^T = \mathbf{x}^2_{pss}(t^{m+1} + \alpha_2^{m+1}), \tag{6.76d}$$

$$\alpha_1^1 = 0,\ \alpha_2^1 = 0,\ m = 1, \dots,$$

where $\tau = t^{m+1} - t^m$ denotes the time step. For the Newton iterations in (6.76a) and (6.76b) we take (α_1^m, α_2^m) as initial guess on the time level $(m+1)$. This provides very fast convergence (in our applications within around four Newton iterations). See [123] and references therein for more details on time integration of electric circuits.

6.3.4 *Resistive and Capacitive Coupling*

For completeness in this section we describe how the phase noise theory applies to two oscillators coupled by a resistor or a capacitor.

6.3.4.1 Resistive Coupling

Resistive coupling is modeled by connecting two oscillators by a single resistor, see Fig. 6.23. The current i_{R_0} flowing through the resistor R_0 satisfies the following relation

$$i_{R_0} = \frac{v_1 - v_2}{R_0}, \tag{6.77}$$

where R_0 is the coupling resistance. Then the phase macromodel is given by

$$\dot{\alpha_1}(t) = \mathbf{V}_1^T(t + \alpha_1(t)) \cdot \begin{pmatrix} (v_1 - v_2)/R_0 \\ 0 \end{pmatrix}, \tag{6.78a}$$

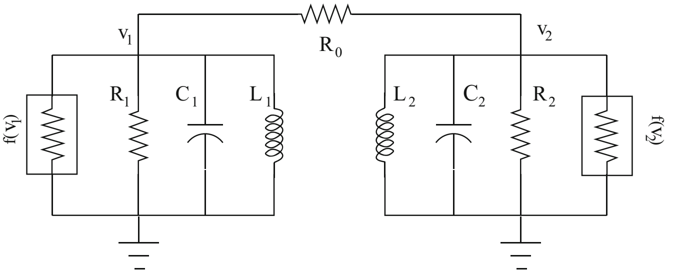


Fig. 6.23 Two resistively coupled LC oscillators

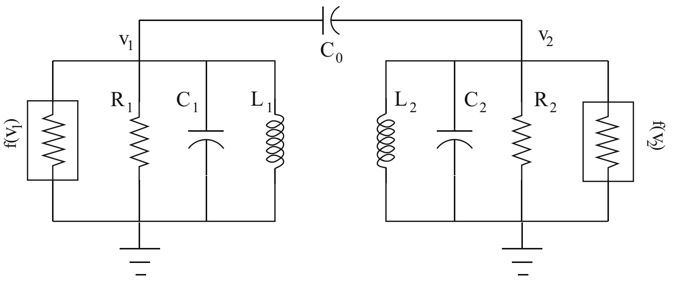


Fig. 6.24 Two capacitively coupled LC oscillators

$$\dot{\alpha_2}(t) = \mathbf{V}_2^T(t + \alpha_2(t)) \cdot \begin{pmatrix} -(v_1 - v_2)/R_0 \\ 0 \end{pmatrix}, \tag{6.78b}$$

where the voltages are updated by using (6.71). More details on resistively coupled oscillators can be found in [113].

6.3.4.2 Capacitive Coupling

When two oscillators are coupled via a single capacitor with a capacitance C_0 (see Fig. 6.24), then the current i_{C_0} through the capacitor C_0 satisfies

$$i_{C_0} = C_0 \frac{d(v_1 - v_2)}{dt}. \tag{6.79}$$

In this case the phase macromodel is given by

$$\dot{\alpha_1}(t) = \mathbf{V}_1^T(t + \alpha_1(t)) \cdot \begin{pmatrix} C_0 \dfrac{d(v_1 - v_2)}{dt} \\ 0 \end{pmatrix}, \tag{6.80a}$$

$$\dot{\alpha_2}(t) = \mathbf{V}_2^T(t + \alpha_2(t)) \cdot \begin{pmatrix} -C_0 \dfrac{d(v_1 - v_2)}{dt} \\ 0 \end{pmatrix}, \tag{6.80b}$$

where the voltages are updated by using (6.71).

Time discretization of (6.78) and (6.80) is done according to (6.76).

6.3.5 Small Parameter Variation Model for Oscillators

For many applications performing simulations with nominal design parameters is no longer sufficient and it is necessary to do simulations around the nominal parameters. In practice designers use Monte-Carlo type simulation techniques to get insight about the device performance for small parameter variations. However these methods can be very time consuming and not applicable for large problems. For analyzing small parameter variations one can use polynomial chaos approach described in [119]. But in this paper we apply the technique described in [128] to mutually coupled oscillators. Here we briefly sketch the ideas of the method and for details we refer to [128].

Consider an oscillator under a perturbation $\mathbf{b}(t)$ described by a set of ODE's:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} + f(\mathbf{x}, p) = \mathbf{b}(t), \tag{6.81}$$

where f describes the nonlinearity in the oscillator and it is a function of the state variables $\mathbf{x}$ and the parameter $\mathbf{p}$. Let us consider a parameter variation

$$\mathbf{p} = \mathbf{p}_0 + \Delta\mathbf{p}, \tag{6.82}$$

where $\mathbf{p}_0$ is the nominal parameter and $\Delta\mathbf{p}$ is the parameter deviation from $\mathbf{p}_0$. Then for small parameter deviations the phase shift equation for (6.81) reads

$$\dot{\alpha}(t) = \mathbf{V}^T(t + \alpha(t)) \cdot (\mathbf{b}(t) - F_P(t + \alpha(t))\Delta\mathbf{p}), \tag{6.83a}$$

$$\alpha(0) = 0, \tag{6.83b}$$

where $\mathbf{V}(t)$ is the perturbation projection vector of the oscillator with nominal parameters and

$$F_P(t + \alpha(t)) = \left.\frac{\partial f}{\partial \mathbf{p}}\right|_{\mathbf{x}_{\text{pss}}(t+\alpha(t)),\mathbf{p}_0}, \tag{6.84}$$

where $\mathbf{x}_{\text{pss}}$ is the PSS of (6.81) with nominal parameters.

In Sect. 6.3.9.1 we show numerical experiments of two inductively coupled oscillators using small parameter variations.

6.3.6 Oscillator Coupling with Balun

In this section we analyze inductive coupling effects between an oscillator and a balun. A balun is an electrical transformer that can transform balanced signals to unbalanced signals and vice versa, and they are typically used to change impedance (applications in (RF) radio). The (unintended) coupling between an oscillator and a balun typically occurs on chips that integrate several oscillators for, for instance, FM radio, Bluethooth and GPS, and hence it is important to understand possible

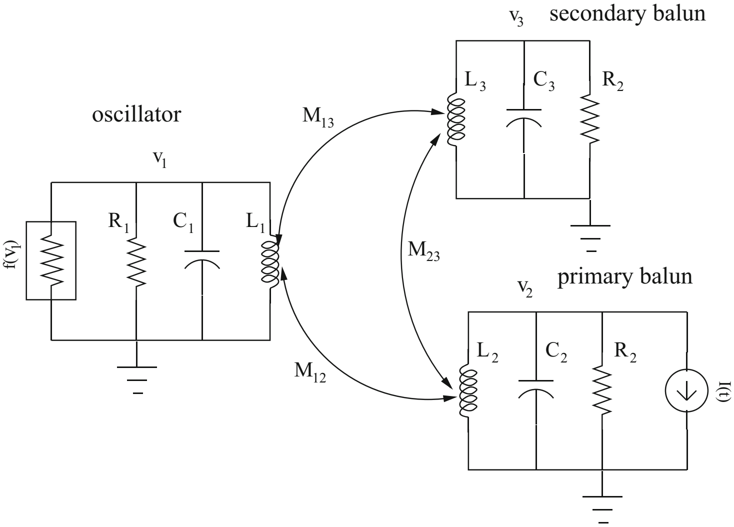


Fig. 6.25 Oscillator coupled with a balun

coupling effects during the design. In Fig. 6.25 a schematic view is given of an oscillator which is coupled with a balun via mutual inductors.

The following mathematical model is used for oscillator and balun coupling (see Fig. 6.25):

$$C_1\frac{dv_1(t)}{dt}+\frac{v_1(t)}{R_1}+i_1(t)+S\tanh(\frac{Gn}{S}v_1(t))=0, \tag{6.85a}$$

$$L_1\frac{di_1(t)}{dt}+M_{12}\frac{di_2(t)}{dt}+M_{13}\frac{di_3(t)}{dt}-v_1(t)=0, \tag{6.85b}$$

$$C_2\frac{dv_2(t)}{dt}+\frac{v_2(t)}{R_2}+i_2(t)+I(t)=0, \tag{6.85c}$$

$$L_2\frac{di_2(t)}{dt}+M_{12}\frac{di_1(t)}{dt}+M_{23}\frac{di_3(t)}{dt}-v_2(t)=0, \tag{6.85d}$$

$$C_3\frac{dv_3(t)}{dt}+\frac{v_3(t)}{R_3}+i_3(t)=0, \tag{6.85e}$$

$$L_3\frac{di_3(t)}{dt}+M_{13}\frac{di_1(t)}{dt}+M_{23}\frac{di_2(t)}{dt}-v_3(t)=0, \tag{6.85f}$$

where $M_{ij}=k_{ij}\sqrt{L_iL_j}$, $i,j=1,2,3$, $i<j$ is the mutual inductance and k_{ij} is the coupling factor. The parameters of the nonlinear resistor are $S=1/R_1$ and $G_n=-1.1/R_1$ and the current injection in the primary balun is denoted by $I(t)$.

For small coupling factors we can consider $M_{12}\frac{di_2(t)}{dt}+M_{13}\frac{di_3(t)}{dt}$ in (6.85b) as a small perturbation to the oscillator. Then similar to (6.75), we can apply the phase shift macromodel to (6.85a)–(6.85b). The reduced model corresponding to (6.85a)–(6.85b) is

$$\frac{\mathrm{d}\alpha(t)}{\mathrm{d}t} = \mathbf{V}^T(t+\alpha(t)) \cdot \begin{pmatrix} 0 \\ -M_{12}\frac{di_2(t)}{dt} - M_{13}\frac{di_3(t)}{dt} \end{pmatrix}. \tag{6.86}$$

The balun is described by a linear circuit (6.85c)–(6.85f) which can be written in a more compact form:

$$E\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} + A\mathbf{x}(t) + B\frac{\mathrm{d}i_1(t)}{\mathrm{d}t} + C = 0, \tag{6.87}$$

where

$$E = \begin{pmatrix} C_2 & 0 & 0 & 0 \\ 0 & L_2 & 0 & M_{23} \\ 0 & 0 & C_3 & 0 \\ 0 & M_{23} & 0 & L_3 \end{pmatrix}, \tag{6.88a}$$

$$A = \begin{pmatrix} 1/R_2 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1/R_3 & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix}, \tag{6.88b}$$

$$B^T = \begin{pmatrix} 0 & M_{12} & 0 & M_{13} \end{pmatrix}, \tag{6.88c}$$

$$C^T = \begin{pmatrix} I(t) & 0 & 0 & 0 \end{pmatrix}, \tag{6.88d}$$

$$\mathbf{x}^T = \begin{pmatrix} v_2(t) & i_2(t) & v_3(t) & i_3(t) \end{pmatrix}. \tag{6.88e}$$

With these notations (6.86) and (6.87) can be written in the following form

$$\frac{\mathrm{d}\alpha(t)}{\mathrm{d}t} = \mathbf{V}^T(t+\alpha(t)) \cdot \begin{pmatrix} 0 \\ -B^T\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} \end{pmatrix}, \tag{6.89}$$

$$E\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} + A\mathbf{x}(t) + B\frac{\mathrm{d}i_1(t)}{\mathrm{d}t} + C = 0, \tag{6.90}$$

where $i_1(t)$ is computed by using (6.71). This system can be solved by using a finite difference method.

6.3.7 Oscillator Coupling to a Transmission Line

In some applications oscillators are coupled via transmission lines. By coupling oscillators via transmission lines, for instance, one can reduce the clock skew in clock distribution networks [102]. Accurate models for transmission lines may contain up to thousands or millions of RLC components [129]. Furthermore, the oscillators or the components that perturb (couple to) the oscillators can consists of many RLC components, for instance when ones takes into account parasitic effects. Since simulation times usually increase with the number of elements, one would like to limit the number of (parasitic) components as much as possible, without losing accuracy.

The schematic view of an oscillator coupled to a transmission line is given in Fig. 6.26. Using phase macromodel for oscillator and by applying Kirchhoff's current law to the transmission line circuit, we obtain the following set of differential equations:

$$\frac{\mathrm{d}\alpha(t)}{\mathrm{d}t} = \mathbf{V}^T(t+\alpha(t)) \cdot \begin{pmatrix} \frac{y(t)-v(t)}{R_1} \\ 0 \end{pmatrix} \tag{6.91a}$$

$$E\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = A\mathbf{x}(t) + B\mathbf{u}(t), \tag{6.91b}$$

$$y(t) = \mathcal{C}^T\mathbf{x}, \tag{6.91c}$$

where

$$E = \mathrm{diag}(C_1, C_2, \dots, C_n),\ A = \mathrm{tridiag}(\frac{1}{R_i}, -\frac{1}{R_i} - \frac{1}{R_{i+1}}, \frac{1}{R_{i+1}}), \tag{6.92a}$$

$$B = \begin{pmatrix} \frac{1}{R_1} & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 1 \end{pmatrix},\ \mathbf{x} = \begin{pmatrix} v_1(t) \\ v_2(t) \\ \vdots \\ v_n(t) \end{pmatrix},\ \mathbf{u}(t) = \begin{pmatrix} v(t) \\ I(t) \end{pmatrix},\ \mathcal{C} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{6.92b}$$

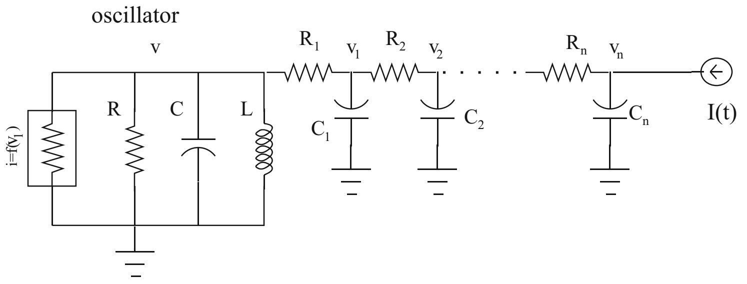


Fig. 6.26 Oscillator coupled to a transmission line

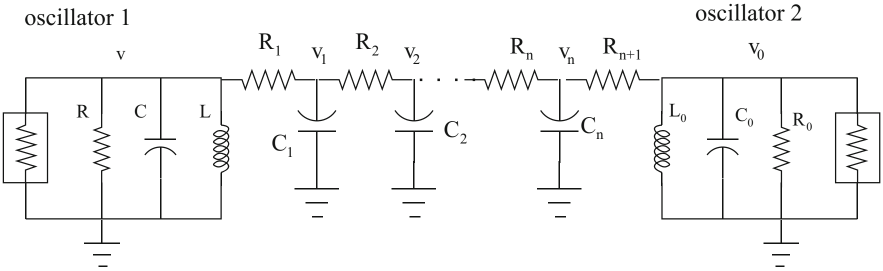


Fig. 6.27 Two oscillators coupled via a transmission line

In a similar way the phase macromodel of two oscillators coupled via a transmission line, see Fig. 6.27, is given by the following equations:

$$\frac{\mathrm{d}\alpha_1(t)}{\mathrm{d}t} = \mathbf{V}_1^T(t + \alpha_1(t)) \cdot \begin{pmatrix} \dfrac{v_1(t) - v(t)}{R_1} \\ 0 \end{pmatrix} \tag{6.93a}$$

$$E\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = A\mathbf{x}(t) + B\mathbf{u}(t), \tag{6.93b}$$

$$\frac{\mathrm{d}\alpha_2(t)}{\mathrm{d}t} = \mathbf{V}_2^T(t + \alpha_2(t)) \cdot \begin{pmatrix} \dfrac{v_n(t) - v_0(t)}{R_{n+1}} \\ 0 \end{pmatrix}, \tag{6.93c}$$

where $\alpha_1(t)$ and $\alpha_2(t)$ ($\mathbf{V}_1$ and $\mathbf{V}_2$) are phase shifts (PPV's) of the corresponding oscillator. The matrices E, A and $\mathbf{x}$ are given by (6.92) and

$$B = \begin{pmatrix} \frac{1}{R_1} & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & \frac{1}{R_{n+1}} \end{pmatrix}, \; \mathbf{u}(t) = \begin{pmatrix} v(t) \\ v_0(t) \end{pmatrix}. \tag{6.94}$$

6.3.8 Model Order Reduction

Model order reduction (MOR) techniques [94, 96, 97, 124] can be used to reduce the number of elements significantly. Here we show how model order reduction can be used for the analysis of oscillator perturbation effects as well. Since the main focus is to show how MOR techniques can be used (and not which technique is the most suitable), we limit the discussion here to Balanced Truncation [118]. For other methods, see, e.g., [94, 96, 97, 124].

Given a dynamical system (A, B, C) (assume $E = I$), balanced truncation [118] consists of first computing a balancing transformation $V \in \mathbb{R}^{n \times n}$. The balanced system $(V^T A V, V^T B, V^T C)$ has the nice property that the Hankel Singular Values[6] are easily available. A reduced order model can be constructed by selecting the columns of V that correspond to the $k < n$ largest Hankel Singular Values. With $V_k \in \mathbb{R}^{n \times k}$ having as columns these k columns, the reduced order model (of order k) becomes $(V_k^A V_k, V_k^T B, V_k^T C)$. If $E \neq I$ is nonsingular, balanced truncation can be applied to $(E^{-1}A, E^{-1}B, C)$. For more details on balanced truncation, see [96, 97, 118, 124].

In this section we apply model order reduction to linear circuits that are coupled to oscillators, and the relevant equations for each problem describing linear circuits have the form of (6.89b)–(6.89c). For each problem the corresponding matrices A, E, B, and C can be identified readily, see (6.88), (6.92), (6.94) and note $C \equiv \mathcal{C}$. We use Matlab [117] implementation for balanced truncation to obtain reduced order models:

```
sys = ss( -E\A, -E\B, C', 0 ) ;
[hsv, baldata] = hsvd(sys); % Hankel singular values
mor_dim = nnz((hsv>1e-10)); % choose largest singular
% values where mor_dim is the dimension
% of the reduced system
rsys= balred(sys,mor_dim,'Elimination','Truncate',...
'Balancing', baldata) ; %truncate
```

Note that we can apply balanced truncation because E is nonsingular. It is well known that in many cases in circuit simulation the system is a descriptor system and hence E is singular. Although generalizations of balanced truncation to descriptor systems exist [124, 125], other MOR techniques such as Krylov subspace methods and modal approximation might be more appropriate. We refer the reader to [94, 96, 97, 124] for a good introduction to such techniques and MOR in general.

6.3.9 Numerical Experiments

It is known that a perturbed oscillator either locks to the injected signal or is pulled, in which case side band frequencies all fall on one side of the injected signal, see, e.g., [111]. We will see that contrary to the single oscillator case, where side band frequencies all fall on one side of the injected signal, for (weakly) coupled oscillators a double-sided spectrum is formed.

In Sects. 6.3.9.1–6.3.9.3 we consider two LC oscillators with different kinds of coupling and injection. The inductance and resistance in both oscillators are

[6] Similar to singular values of matrices, the Hankel singular values and corresponding vectors can be used to identify the dominant subspaces of the system's statespace: the larger the Hankel singular value, the more dominant.

$L_1 = L_2 = 0.64\,\text{nH}$ and $R_1 = R_2 = 50\,\Omega$, respectively. The first oscillator is designed to have a free running frequency $f_1 = 4.8$ GHz with capacitance $C_1 = 1/(4L_1\pi^2 f_1^2) = 1.7178$ pF. Then the inductor current in the first oscillator is $A_1 = 0.0303$ A and the capacitor voltage is $V_1 = 0.5844$ V. In a similar way the second oscillator is designed to have a free running frequency $f_2 = 4.6$ GHz with the inductor current $A_2 = 0.0316$ A and the capacitor voltage $V_2 = 0.5844$ V. For both oscillators we choose $S_i = 1/R_i$, $G_n = -1.1/R_i$ with $i = 1, 2$. In Sect. 6.3.9.4 we describe experiments for an oscillator coupled to a balun.

The values for L, C, R and (mutual) coupling factors are based on measurement data and layout simulations of real designs.

In all the numerical experiments the simulations are run until $T_{\text{final}} = 6 \cdot 10^{-7}$ s with the fixed time step $\tau = 10^{-11}$. Simulation results with the phase shift macromodel are compared with simulations of the full circuit using the CHORAL[103, 121] one-step time integration algorithm, hereafter referred to as full simulation. All experiments have been carried out in Matlab 7.3. We would like to remark that in all experiments simulations with the macromodels were typically ten times faster than the full circuit simulations.

In all experiments, for a given oscillator or balun we use the response of the nodal voltage to plot the spectrum (spectrum composed of discrete harmonics) of the signal.

6.3.9.1 Inductively Coupled Oscillators

Numerical simulation results of two inductively coupled oscillators, see Fig. 6.22, for different coupling factors k are shown in Fig. 6.28, where the frequency is plotted versus the Power Spectral Density (PSD[7]). In Fig. 6.28 we present results for the first oscillator. Similar results are obtained for the second oscillator around its own carrier frequency. For small values of the coupling factor we observe a very good approximation with the full simulation results. As the coupling factor grows, small deviations in the frequency occur, see Fig. 6.28d. Because of the mutual pulling effects between the two oscillators a double sided spectrum is formed around each oscillator carrier frequency. The additional sidebands are equally spaced by the frequency difference of the two oscillators.

The phase shift $\alpha_1(t)$ of the first oscillator for a certain time interval is given in Fig. 6.29. We note that it has a sinusoidal behavior. For a single oscillator under perturbation a completely different behavior is observed: in locked condition the phase shift changes linearly, whereas in the unlocked case the phase shift has a nonlinear behavior different than a sinusoidal, see for example [112].

[7] Matlab code for plotting the PSD is given in [107].

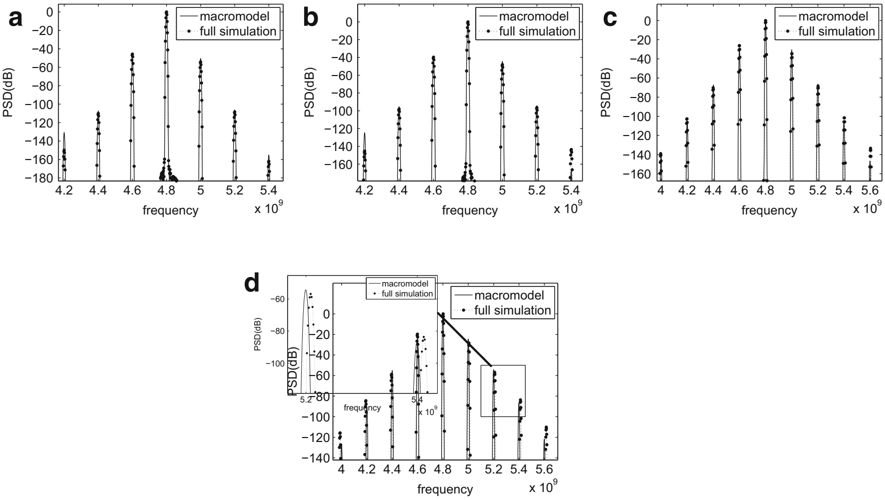


Fig. 6.28 Inductive coupling. Comparison of the output spectrum of the first oscillator obtained by the phase macromodel and by the full simulation for a different coupling factor k. (**a**) $k = 0.0005$. (**b**) $k = 0.001$. (**c**) $k = 0.005$. (**d**) $k = 0.01$

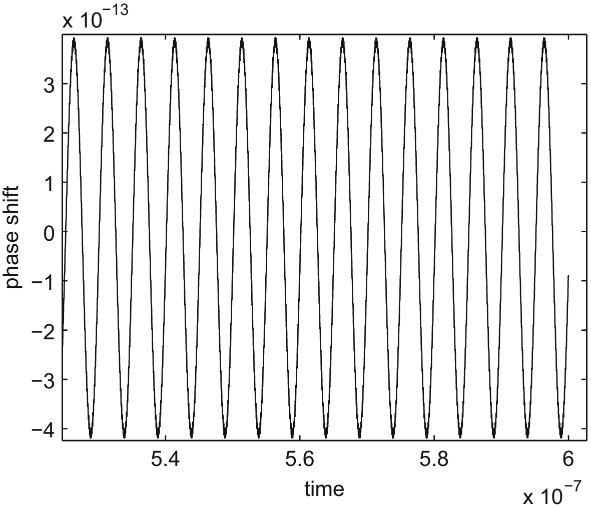


Fig. 6.29 Inductive coupling. Phase shift $\alpha_1(t)$ of the first oscillator with $k = 0.001$

Parameter Variation in Two Inductively Coupled Oscillators

Let us consider two inductively coupled oscillators with the nominal parameters given in Sect. 6.3.9 and a small parameter $\varDelta L$ variation in the inductance of the second oscillator. Then the corresponding model is:

$$\frac{\mathrm{d}v_1(t)}{\mathrm{d}t} + \frac{v_1(t)}{C_1 R_1} + i_1(t) + \frac{S}{C_1}\tanh(\frac{G_n}{S}v_1(t)) = 0, \tag{6.95a}$$

$$\frac{\mathrm{d}i_1(t)}{\mathrm{d}t} - \frac{v_1(t)}{L_1} = -\frac{M}{L_1}\frac{\mathrm{d}i_2(t)}{\mathrm{d}t}, \tag{6.95b}$$

$$\frac{\mathrm{d}v_2(t)}{\mathrm{d}t} + \frac{v_2(t)}{C_2 R_2} + i_2(t) + \frac{S}{C_2}\tanh(\frac{G_n}{S}v_2(t)) = 0, \tag{6.95c}$$

$$\frac{\mathrm{d}i_2(t)}{\mathrm{d}t} - \frac{v_2(t)}{L_2 + \varDelta L} = -\frac{M}{L_2 + \varDelta L}\frac{\mathrm{d}i_1(t)}{\mathrm{d}t}. \tag{6.95d}$$

By using the small parameter variation model given in Sect. 6.3.5 we obtain the corresponding phase shift macromodel for (6.95):

$$\dot{\alpha_1}(t) = \mathbf{V}_1^T(t + \alpha_1(t)) \cdot \begin{pmatrix} 0 \\ -\frac{M}{L_1}\frac{\mathrm{d}i_2(t)}{\mathrm{d}t} \end{pmatrix}, \tag{6.96a}$$

$$\dot{\alpha_2}(t) = \mathbf{V}_2^T(t + \alpha_2(t)) \cdot \begin{pmatrix} 0 \\ -\frac{M}{L_2 + \varDelta L}\frac{\mathrm{d}i_1(t)}{\mathrm{d}t} - \frac{v_2(t)}{L_2^2}\varDelta L \end{pmatrix}, \tag{6.96b}$$

where the currents and voltages are evaluated by using (6.75c)–(6.75d).

For this numerical experiments we consider the coupling factor to be equal to $k = 0.0005$. Furthermore, let us denote by $f_2^{\mathrm{full},\varDelta L}$ and $f_2^{\mathrm{phase},\varDelta L}$ the new frequency of the second oscillator obtained by full simulation and phase macromodel for the given parameter variation $\varDelta L$. Then we define

$$\varDelta f = f_2^{\mathrm{full},\varDelta L} - f_2^{\mathrm{phase},\varDelta L}.$$

In Fig. 6.30 we show the relative frequency difference $\varDelta f$ versus parameter variation $\varDelta L$. We note that for small parameter variations ($\varDelta L/L_2 \leq 0.01$) the phase macromodel provides a good approximation to the full simulation results.

In Fig. 6.31 we show the output spectrum of the second oscillator for several values of the parameter $\varDelta L$.

6.3.9.2 Capacitively Coupled Oscillators

The coupling capacitance in Fig. 6.24 is chosen to be $C_0 = k \cdot C_{\mathrm{mean}}$, where $C_{\mathrm{mean}} = (C_1 + C_2)/2 = 1.794 \cdot 10^{-12}$ and we call k the capacitive coupling factor. Simulation results for the first oscillator for different capacitive coupling factors k are given in Fig. 6.32 (similar results are obtained for the second oscillator around its own carrier frequency).

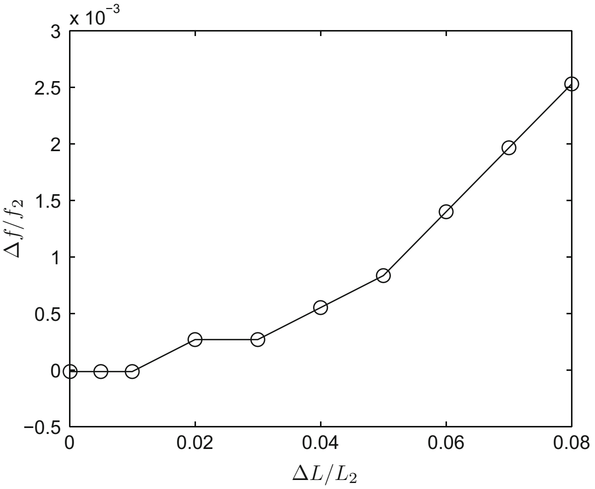


Fig. 6.30 Frequency difference versus parameter variation

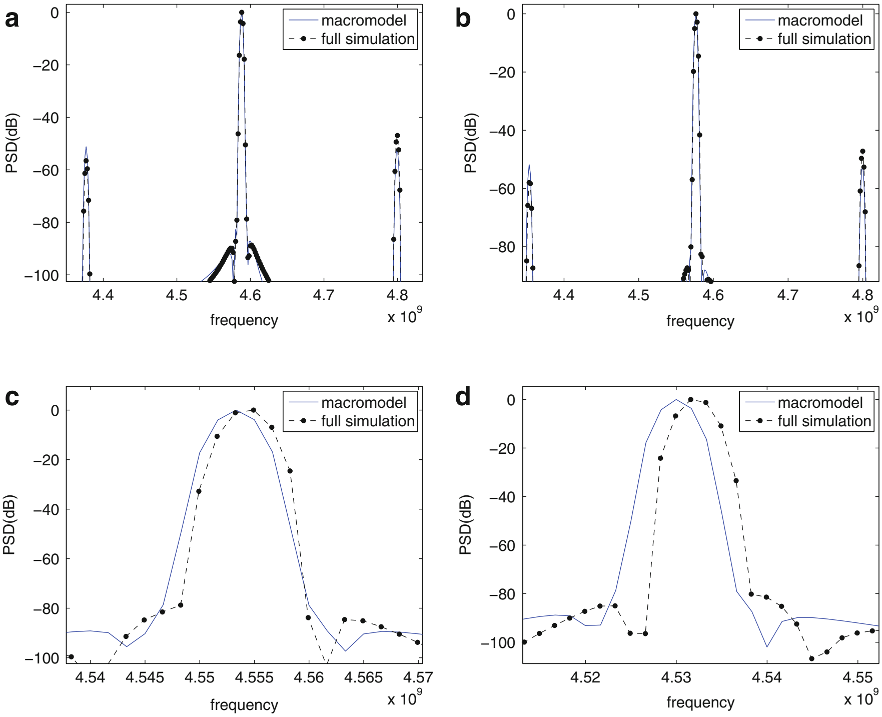


Fig. 6.31 Output spectrum of the second oscillator for several parameter variations $\varDelta L$. (**a**) $\varDelta L/L_2 = 0.005$. (**b**) $\varDelta L/L_2 = 0.01$. (**c**) $\varDelta L/L_2 = 0.02$. (**d**) $\varDelta L/L_2 = 0.03$

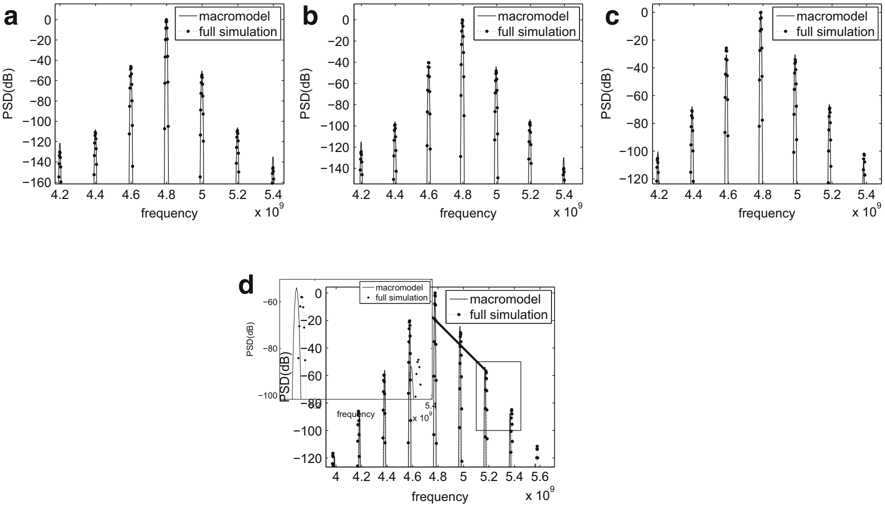


Fig. 6.32 Capacitive coupling. Comparison of the output spectrum of the first oscillator obtained by the phase macromodel and by the full simulation for a different coupling factor k. (**a**) $k = 0.0005$. (**b**) $k = 0.001$. (**c**) $k = 0.005$. (**d**) $k = 0.01$

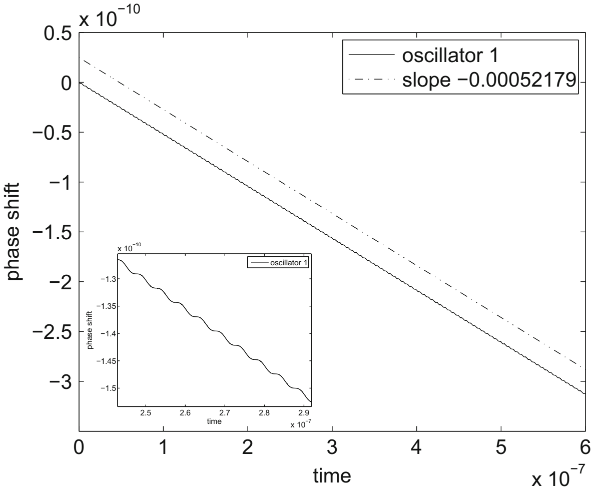


Fig. 6.33 Capacitive coupling. Phase shift of the first oscillator with $k = 0.001$

For a larger coupling factor $k = 0.01$ the phase shift macromodel shows small deviations from the full simulation results Fig. 6.32d.

The phase shift $\alpha_1(t)$ of the first oscillator and a zoomed section for some interval are given in Fig. 6.33. In a long run the phase shift seems to change linearly with a slope of $a = -0.00052179$. The linear change in the phase shift is a clear indication that the frequency of the first oscillator is changed and is locked to a new frequency, which is equal to $(1+a)f_1$. The change of the frequency can be explained as follows: as noted in [114], capacitive coupling may change the free running frequency because this kind of coupling changes the equivalent tank capacitance. From a mathematical point of view it can be explained in the following way. For the capacitively coupled oscillators the governing equations can be written as:

$$(C_1 + C_0)\frac{dv_1(t)}{dt} + \frac{v_1(t)}{R} \tag{6.97a}$$

$$+ i_1(t) + S\tanh(\frac{G_n}{S}v_1(t)) = C_0\frac{dv_2(t)}{dt},$$

$$L_1\frac{di_1(t)}{dt} - v_1(t) = 0, \tag{6.97b}$$

$$(C_2 + C_0)\frac{dv_2(t)}{dt} + \frac{v_2(t)}{R} \tag{6.97c}$$

$$+ i_2(t) + S\tanh(\frac{G_n}{S}v_2(t)) = C_0\frac{dv_1(t)}{dt},$$

$$L_2\frac{di_2(t)}{dt} - v_2(t) = 0. \tag{6.97d}$$

This shows that the capacitance in each oscillator is changed by C_0 and that the new frequency of each oscillator is

$$\tilde{f}_i = \frac{1}{2\pi\sqrt{L_1(C_i + C_0)}},\ i = 1, 2.$$

In the zoomed figure within Fig. 6.33 we note that the phase shift is not exactly linear but that there are small wiggles. By numerical experiments it can be shown that these small wiggles are caused by a small sinusoidal contribution to the linear part of the phase shift. As in case of mutually coupled inductors, the small sinusoidal contributions are caused by mutual pulling of the oscillators (right-hand side terms in (6.97a) and (6.97c)).

6.3.9.3 Inductively Coupled Oscillators Under Injection

As a next example, let us consider two inductively coupled oscillators where in one of the oscillators an injected current is applied. Let us consider the case where a sinusoidal current of the form

$$I(t) = A_{\text{inj}}\sin(2\pi(f_1 - f_{\text{off}})t) \tag{6.98}$$

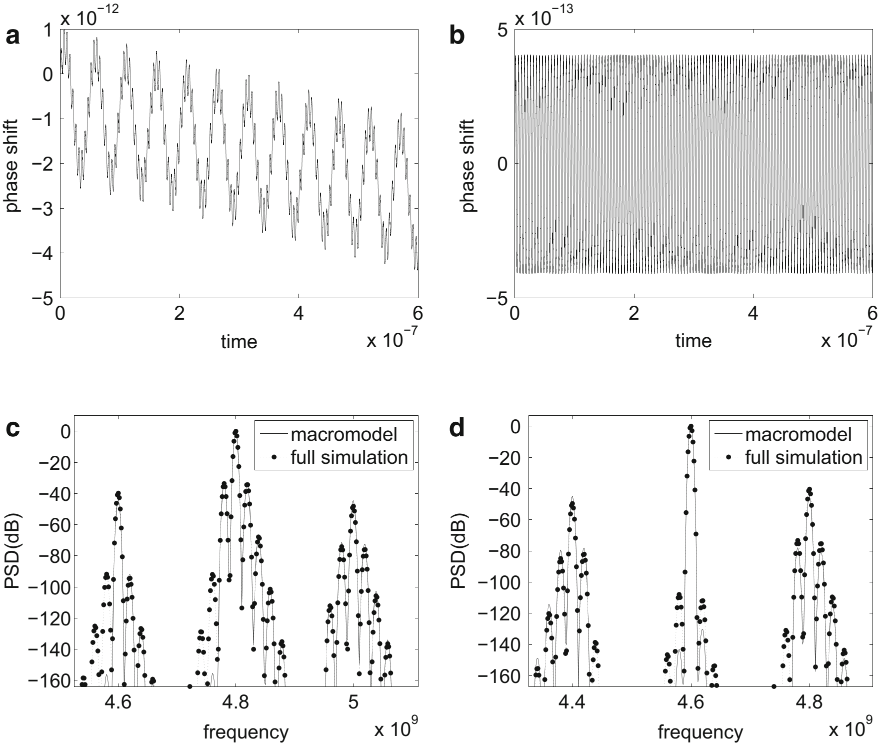


Fig. 6.34 Inductive coupling with injection and $k = 0.001$. *Top*: phase shift. *Bottom*: comparison of the output spectrum obtained by the phase macromodel and by the full simulation with a small current injection. (**a**) Oscillator 1. (**b**) Oscillator 2. (**c**) Oscillator 1. (**d**) Oscillator 2

is injected in the first oscillator. Then (6.75a) is modified to

$$\dot{\alpha_1}(t) = \mathbf{V}_1^T(t + \alpha_1(t)) \cdot \begin{pmatrix} -I(t) \\ -M\dfrac{di_2(t)}{dt} \end{pmatrix}. \tag{6.99}$$

For a small current injection with $A_{\mathrm{inj}} = 10\,\mu\mathrm{A}$ and an offset frequency $f_{\mathrm{off}} =$ 20 MHz the spectra of both oscillators and the phase shift with coupling factor $k = 0.001$ are given in Fig. 6.34. It is clear from Figs. 6.34a, b that the phase shift of both oscillators does not change linearly, which implies that the oscillators are not in the steady state. As a result in Figs. 6.34c, d we observe spectral widening in the spectra of both oscillators. We note that the phase macromodel simulations are good approximations of the full simulation results.

6.3.9.4 Oscillator Coupled to a Balun

Finally, consider an oscillator coupled to a balun as shown in Fig. 6.25 with the following parameters values:

Oscillator	Primary balun	Secondary balun
$L_1 = 0.64\,\text{nH}$ $C_1 = 1.71\,\text{pF}$ $R_1 = 50\,\Omega$	$L_2 = 1.10\,\text{nH}$ $C_2 = 4.00\,\text{pF}$ $R_2 = 40\,\Omega$	$L_3 = 3.60\,\text{nH}$ $C_3 = 1.22\,\text{pF}$ $R_2 = 60\,\Omega$

The coupling factors in (6.85) are chosen to be

$$k_{12} = 10^{-3},\ k_{13} = 5.96 * 10^{-3},\ k_{23} = 9.33 * 10^{-3}. \tag{6.100}$$

The injected current in the primary balun is of the form

$$I(t) = A_{\text{inj}} \sin(2\pi(f_0 - f_{\text{off}})t), \tag{6.101}$$

where $f_0 = 4.8\,\text{GHz}$ is the oscillator's free running frequency and $f_{\text{off}} = 20$ MHz is the offset frequency.

Results of numerical experiments done with the phase macromodel and the full simulations are shown in Fig. 6.35. We note that for a small current injection ($A_{\text{inj}} = 10^{-4} – 10^{-2}$ A) both the oscillator and the balun are pulled by each other. When the injected current is not strong ($A_{\text{inj}} = 10^{-4}$ A) the oscillator is pulled slightly and in the spectrum of the oscillator (Fig. 6.35a) we observe a spectral widening with two spikes around-60 dB (weak "disturbance" of the oscillator). By gradually increasing the injected current, the oscillator becomes more disturbed and in the spectrum we observe widening with higher side band levels, cf. Fig. 6.35c–f. When the injected current is strong enough (with $A_{\text{inj}} = 10^{-1}$ A) to lock the oscillator to the frequency of the injected signal, we observe a single spike at the new frequency. Similar results are also obtained for the secondary balun.

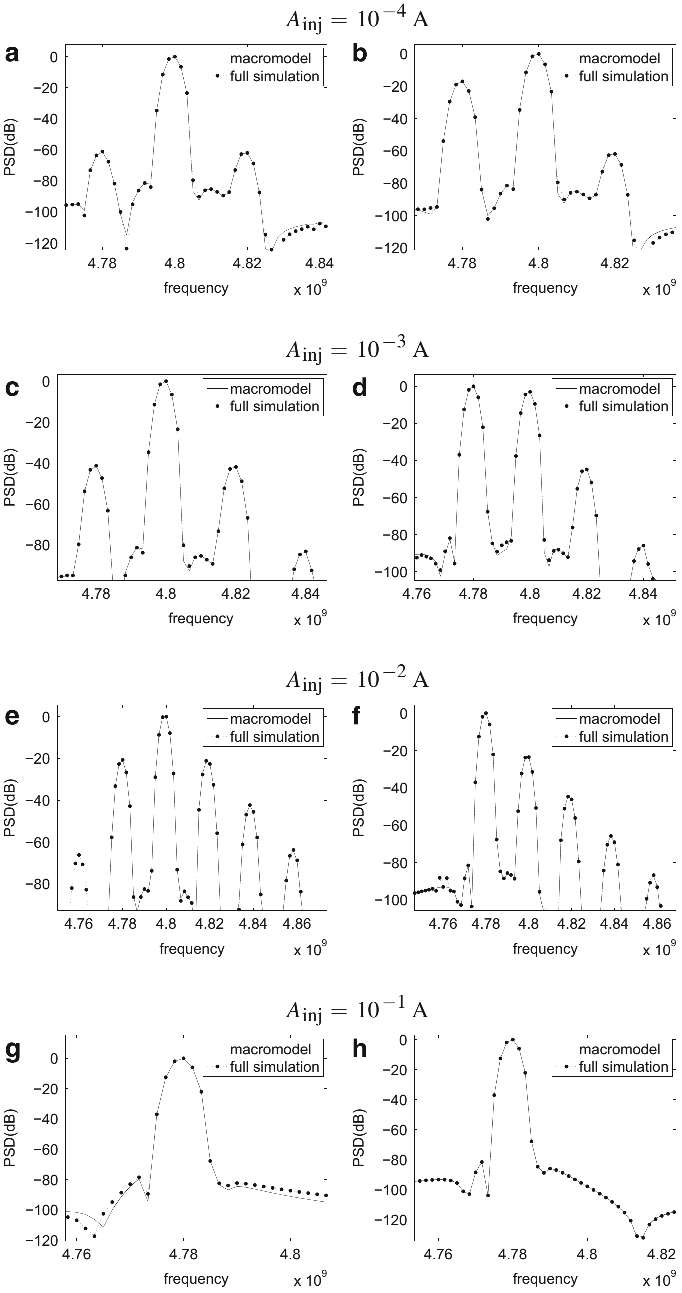
Oscillator Coupled to a Balun

Consider an oscillator coupled to a balun as shown in Fig. 6.25 with the following parameters values:

Oscillator	Primary balun	Secondary balun
$L_1 = 0.64 \cdot 10^{-9}$ $C_1 = 1.71 \cdot 10^{-12}$ $R_1 = 50$	$L_2 = 1.10 \cdot 10^{-9}$ $C_2 = 4.00 \cdot 10^{-12}$ $R_2 = 40$	$L_3 = 3.60 \cdot 10^{-9}$ $C_3 = 1.22 \cdot 10^{-12}$ $R_2 = 60$

The coefficients of the mutual inductive couplings are $k_{12} = 10^{-3}$, $k_{13} = 5.96 * 10^{-3}$, $k_{23} = 9.33 * 10^{-3}$. The injected current in the primary balun is of the form

$$I(t) = A_{\text{inj}} \sin(2\pi(f_0 - f_{\text{off}})t), \tag{6.102}$$



$A_{\text{inj}} = 10^{-4}$ A

$A_{\text{inj}} = 10^{-3}$ A

$A_{\text{inj}} = 10^{-2}$ A

$A_{\text{inj}} = 10^{-1}$ A

Fig. 6.35 Comparison of the output spectrum of the oscillator coupled to a balun obtained by the phase macromodel and by the full simulation for an increasing injected current amplitude A_{inj} and an offset frequency $f_{\text{off}} = 20$ MHz. (**a**) oscillator. (**b**) primary balun. (**c**) oscillator. (**d**) primary balun. (**e**) oscillator. (**f**) primary balun. (**g**) oscillator. (**h**) primary balun

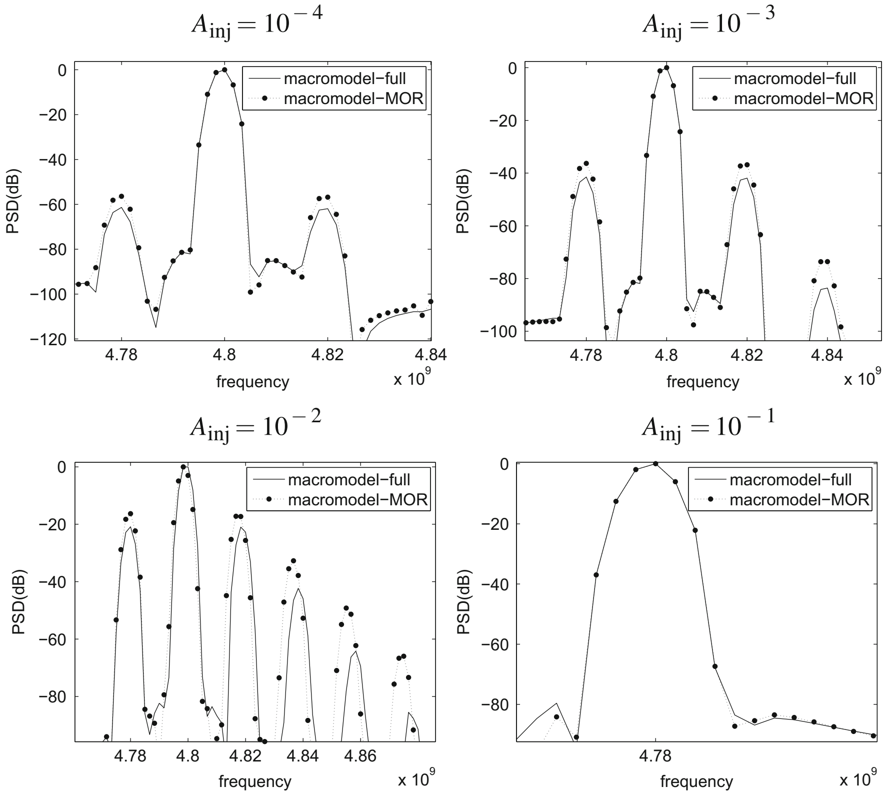


Fig. 6.36 Comparison of the output spectrum of the oscillator coupled to a balun obtained by the macromodel-full and the macromodel-MOR simulations for an increasing injected current amplitude A_{inj} and an offset frequency $f_{\text{off}} = 20$ MHz

where $f_0 = 4.8$ GHz is the oscillator's free running frequency, f_{off} is the offset frequency and A_{inj} is the current amplitude.

Results of the numerical experiments are shown in Fig. 6.36, where the results obtained by the macromodel-MOR technique with mor_dim = 2 provide a good approximation to the full-simulation results. We note that for the injected current with $A_{\text{inj}} = 10^{-1}$ A the oscillator is locked to the injected signal. Similar results are also obtained for the balun.

6.3.9.5 Oscillators Coupled with Transmission Lines

In this section we consider two academic examples, where transmission lines are modeled with RC components.

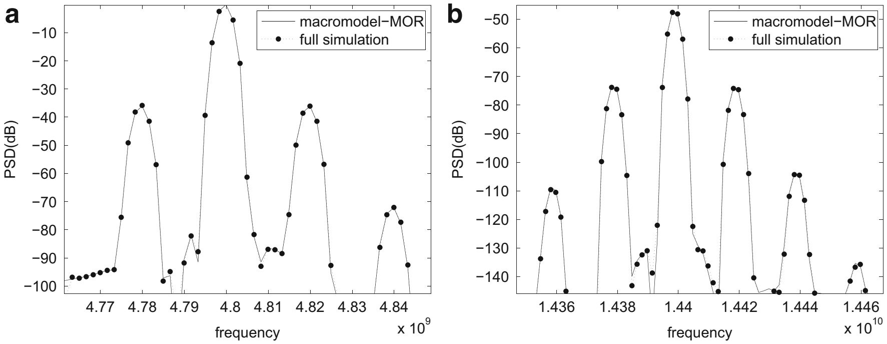


Fig. 6.37 Comparison of the output spectrum around the first and third harmonics of the oscillator coupled to a transmission line, cf. Fig. 6.26. (**a**) first harmonic. (**b**) third harmonic

Single Oscillator Coupled to a Transmission Line

Let us consider the same oscillator as given in the previous section, now coupled to a transmission line, see Fig. 6.26. The size of the transmission line is $n = 100$ with the following parameters: $C_1 = \ldots = C_n = 10^{-2}\,\text{pF}, R_1 = 40\,\text{k}\Omega, R_2 = \ldots = R_n = 1\,\Omega$. The injected current has the form (6.102) with $A_{\text{inj}} = 10^{-2}$ A and $f_{\text{off}} = 20\,\text{MHz}$. Dimension of the reduced system is mor_dim = 18. Simulation results around the first and third harmonics (this oscillator does not have a second harmonic) are shown in Fig. 6.37. The macromodel-MOR method, using techniques described in Sect. 6.3.8, gives a good approximation to the full simulation results.

Two LC Oscillators Coupled via a Transmission Line

For this experiment we consider two LC oscillators coupled via a transmission line with the mathematical model given by (6.93). The first oscillator has a free running frequency $f_1 = 4.8\,\text{GHz}$ and is described in Sect. 6.3.9.4. The second LC oscillator has the following parameter values: $R_0 = 50\,\Omega$, $L_0 = 0.64\,\text{nH}$, $C_0 = 1.87\,\text{pF}$ and a free running frequency $f_2 = 4.6\,\text{GHz}$. The size of the transmission line is $n = 100$ with the following parameters: $C_1 = \ldots = C_n = 10^{-2}\,\text{pF}, R_1 = R_{n+1} = 4\,\text{k}\Omega, R_2 = \ldots = R_n = 0.001\,\Omega$. Dimension of the reduced system is mor_dim = 16. Numerical simulation results are given in Fig. 6.38. We note that macromodel-MOR approach gives a very good approximation to the full-simulation results.

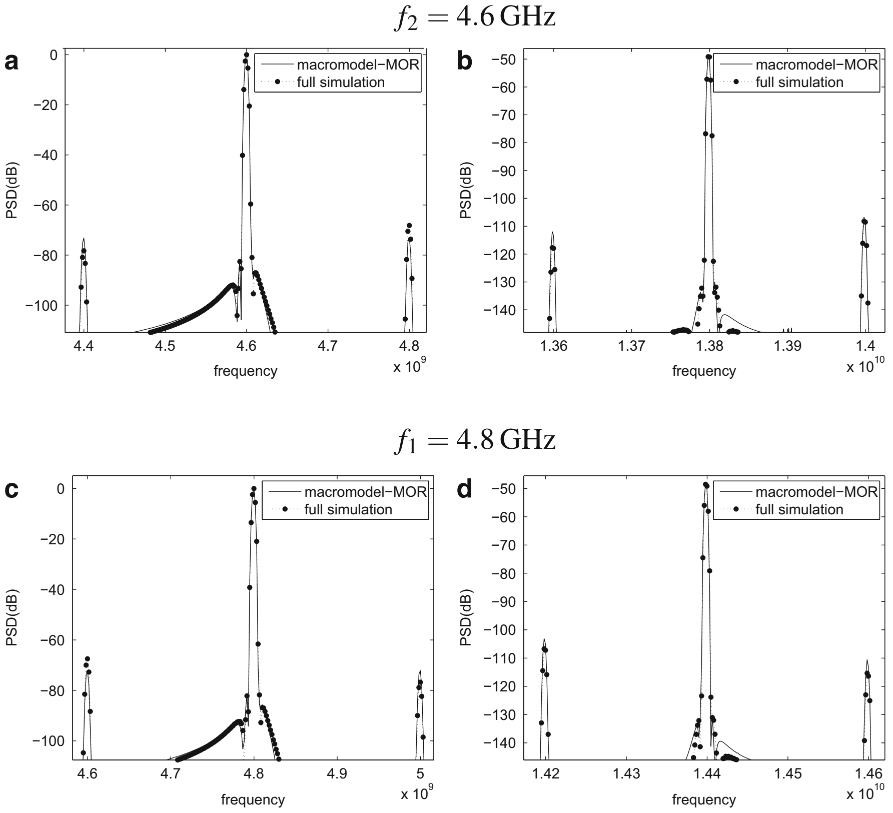


Fig. 6.38 Comparison of the output spectrum around the first and third harmonics of two oscillators coupled via a transmission line. (**a**) first harmonic. (**b**) third harmonic. (**c**) first harmonic. (**d**) third harmonic

6.3.10 Conclusion

In this section we have shown how nonlinear phase macromodels can be used to accurately predict the behavior of individual or mutually coupled voltage controlled oscillators under perturbation, and how they can be used during the design process. Several types of coupling (resistive, capacitive, and inductive) have been described and for small perturbations, the nonlinear phase macromodels produce results with accuracy comparable to full circuit simulations, but at much lower computational costs. Furthermore, we have studied the (unintended) coupling between an oscillator and a balun, a case which typically arises during design and floor planning of RF circuits. For the coupling of oscillators with transmission lines we showed how the phase macromodel can be used with model order reduction techniques to provide an accurate and efficient method.

References

References for Section 6.1

1. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems. SIAM, Philadelphia (2005)
2. Astrid, P.: Reduction of process simulation models: a proper orthogonal decomposition approach. Ph.D.-thesis, Technische Universiteit Eindhoven (2004)
3. Astrid, P., Verhoeven, A.: Application of least squares mpe technique in the reduced order modeling of electrical circuits. In: Proceedings of the 17th International Symposium on MTNS, Kyoto, pp. 1980–1986 (2006)
4. Astrid, P., Weiland, S., Willcox, K., Backx, T.: Missing point estimation in models described by proper orthogonal decomposition. IEEE Trans. Autom. Control **53**(10), 2237–2251 (2008)
5. Bai, Z., Skoogh, D.: Krylov subspace techniques for reduced-order modeling of nonlinear dynamical systems. Appl. Numer. Math. **43**, 9–44 (2002)
6. Bai, Z., Skoogh, D.: A projection method for model reduction of bilinear dynamical systems. Linear Algebra Appl. 415(2–3), 406–425 (2006)
7. Bechtold, T., Striebel, M., Mohaghegh, K., ter Maten, E.J.W.: Nonlinear model order reduction in nanoelectronics: combination of POD and TPWL. PAMM Proc. Appl. Maths Mech. **8**(1), 10057–10060 (2009). doi:10.1002/pamm.200810057. (Special Issue on Proceedings GAMM Annual Meeting 2008)
8. Benner, P., Breiten, T.: Krylov-subspace based model reduction of nonlinear circuit models using bilinear and quadratic-linear approximations. In: Günther, M., Bartel, A., Brunk, M., Schöps, S., Striebel, M. (eds.) Progress in Industrial Mathematics at ECMI 2010. Mathematics in Industry, vol. 17, pp. 153–159. Springer, Berlin/New York (2012)
9. Benner, P., Damm, T.: Lyapunov equations, energy functionals and model order reduction of bilinear and stochastic systems. SIAM J. Control Optim. **49**(2), 686–711 (2011)
10. Bittner, K., Urban, K.: Adaptive wavelet methods using semiorthogonal spline wavelets: sparse evaluation of nonlinear functions. Appl. Comput. Harmon. Anal. **24**, 91–119 (2008)
11. Cai, W., Wang, J.: Adaptive multiresolution collocation methods for initial-boundary value problems of nonlinear PDEs. SIAM J. Numer. Anal. **33**(3), 937–970 (1996)
12. Chaturantabut, C., Sorensen, D.C.: Nonlinear model reduction via discrete empirical interpolation. SIAM J. Sci. Comput. **32**(5), 2737–2764 (2010)
13. Chaturantabut, C., Sorensen, D.C.: A state space error estimate for POD-DEIM nonlinear model reduction. SIAM J. Numer. Anal. **50**(1), 46–63 (2012)
14. Condon, M., Ivanov, R.: Nonlinear systems – algebraic gramians and model reduction. COMPEL Int. J. Comput. Math. Electr. Electron. Eng. **24**(1), 202–219 (2005)
15. Condon, M., Ivanov, R.: Krylov subspaces from bilinear representations of nonlinear systems. COMPEL Int. J. Comput. Math. Electr. Electron. Eng. **26**(2), 399–406 (2007)
16. Dautbegović, E., Condon, M., Brennan, C.: An efficient nonlinear circuit simulation technique. IEEE. Trans. Microw. Theory Tech. **53**(2), 548–555 (2005)
17. Dautbegović, E.: Wavelets in ciruit simulation. In: Roos, J., Costa, L.R.J. (eds.) Scientific Computing in Electrical Engineering SCEE 2008. Mathematics in Industry, vol. 14, pp. 131–141. Springer, Berlin/Heidelberg (2010)
18. Dong, N., Roychowdhury, J.: General-purpose nonlinear model-order reduction using piecewise-polynomial representations. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **27**(2), 249–264 (2008)
19. Freund, R.W.: Krylov-subspace methods for reduced-order modeling in circuit simulation. J. Comput. Appl. Math. **123**(1–2), 395–421 (2000)
20. Fujimoto, K., Scherpen, J.M.A.: Singular value analysis and balanced realizations for nonlinear systems. In: Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.) Model Order Reduction: Theory, Research Aspects and Applications, pp. 251–272. Springer, Berlin/Heidelberg (2008)

21. Gad, E., Nakhla, M.: Efficient model reduction of linear periodically time-varying systems via compressed transient system function. IEEE Trans. Circuit Syst. 1 **52**(6), 1188–1204 (2005)
22. Gu, C.: Model order reduction of nonlinear dynamical systems. Ph.D.-thesis, University of California, Berkeley (2012). http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-217.pdf
23. Günther, M.: Simulating digital circuits numerically – a charge-oriented ROW approach. Numer. Math. **79**, 203–212 (1998)
24. Holmes, P., Lumley, J., Berkooz, G.: Turbulence, Coherent Structures, Dynamical Systems and Symmetry. Cambrige University Press, Cambrige (1996)
25. Ionescu, T.C.: Balanced truncation for dissipative and symmetric nonlinear systems. Ph.D.-thesis, Rijksuniversiteit Groningen (2009)
26. Ionescu, T.C., Scherpen, J.M.A.: Nonlinear cross gramians. In: Korytowski, A., Malanowski, K., Mitkowski, W., Szymkat, M. (eds.) System Modeling and Optimization. Series: IFIP Advances in Information and Communication Technology, vol. 312, pp. 293–306. Springer, Berlin/Heidelberg (2009)
27. Loève, M.: Probability Theory. Van Nostrand, New York (1955)
28. Martinez, J.A.: Model order reduction of nonlinear dynamic systems using multiple projection bases and optimized state-space sampling. Ph.D.-thesis, University of Pittsburgh (2009). http://d-scholarship.pitt.edu/6685/
29. Mohaghegh, K., Striebel, M., ter Maten, J., Pulch, R.: Nonlinear model order reduction based on trajectory piecewise linear approach: comparing different linear cores. In: Roos, J., Costa, L.R.G. (eds.) Scientific Computing in Electrical Engineering SCEE 2008. Mathematics in Industry, vol. 14, pp. 563–570. Springer, Berlin/Heidelberg (2010)
30. Moore, B.: Principal component analysis in linear systems: controllability, observability, and model reduction. IEEE Trans. Autom. Control **26**(1), 17–32 (1981)
31. Pinnau, R.: Model reduction via proper orthogonal decomposition. In: Schilders, W., van der Vorst, H., Rommes, J. (eds.) Model Order Reduction: Theory, Applications, and Research Aspects, pp. 95–109. Springer, Berlin (2008)
32. Rathinam, M., Petzold, L.R.: A new look at proper orthogonal decomposition. SIAM J. Numer. Anal. **41**(5), 1893–1925 (2003)
33. Rewieński, M.J., White, J.: A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. IEEE Trans. CAD Int. Circuit. Syst. **22**(2), 155–170 (2003)
34. Rewieński, M.J.: A trajectory piecewise-linear approach to model order reduction of nonlinear dynamical systems. Ph.D.-thesis, Massachusetts Institute of Technology (2003)
35. Striebel, M., Rommes, J.: Model order reduction of nonlinear systems in circuit simulation: status, open issues, and applications. CSC Preprint 08-07, Chemnitz University of Technology (2008). http://www.tu-chemnitz.de/mathematik/csc/index.php
36. Striebel, M., Rommes, J.: Model order reduction of nonlinear systems in circuit simulation: status and applications. In: Benner, P., Hinze, M., ter Maten, E.J.W. (eds.) Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74, pp. 289–301. Springer, Dordrecht (2011)
37. Striebel, M., Rommes, J.: Model order reduction of parameterized nonlinear systems by interpolating input-output behavior. In: Michielsen, B., Poirier, J.-R. (eds.) Scientific Computing in Electrical Engineering SCEE 2008. Mathematics in Industry, vol. 16, pp. 405–413. Springer, Heidelberg (2012)
38. Striebel, M., Rommes, J.: Model order reduction of nonlinear systems by interpolating input-output behavior. In: Günther, M., Bartel, A., Brunk, M., Schöps, S., Striebel, M. (eds.) Progress in Industrial Mathematics at ECMI 2010. Mathematics in Industry, vol. 17, pp. 145–151. Springer, Berlin/New York (2012)
39. Tiwary, S.K., Rutenbar, R.A.: Scalable trajectory methods for on-demand analog macromodel extraction. In: DAC '05: Proceedings of the 42nd Annual Conference on Design Automation, San Diego, pp. 403–408. ACM, New York (2005)

40. Vasilyev, D., Rewieński, M., White, J.: A TBR-based trajectory piecewise-linear algorithm for generating accurate low-order models for nonlinear analog circuits and MEMS. In: Proceedings of the Design Automation Conference, Anaheim, pp. 490–495 (2003)
41. Verhoeven, A.: Redundancy reduction of IC models by multirate time-integration and model order reduction. Ph.D.-thesis, TU Eindhoven (2008). http://alexandria.tue.nl/extra2/200712281.pdf
42. Verhoeven, A., Voss, T., Astrid, P., ter Maten, E.J.W., Bechtold, T.: Model order reduction for nonlinear problems in circuit simulation. PAMM Proc. Appl. Math. Mech. **7**(1), 1021603–1021604 (2008). doi:10.1002/pamm.200700537. (Special Issue Proceedings ICIAM-2007)
43. Verhoeven, A., ter Maten, J., Striebel, M., Mattheij, R.: Model order reduction for nonlinear IC models. In: Korytowski, A., Malanowski, K., Mitkowski, W., Szymkat, M. (eds.) System Modeling and Optimization. Series: IFIP Advances in Information and Communication Technology, vol. 312, pp. 476–491. Springer, Berlin/Heidelberg (2009)
44. Verhoeven, A., Striebel, M., ter Maten, E.J.W.: Model order reduction for nonlinear IC models with POD. In: Roos, J., Costa, L.R.J. (eds.) Scientific Computing in Electrical Engineering SCEE 2008. Mathematics in Industry, vol. 14, pp. 571–578. Springer, Berlin/Heidelberg (2010)
45. Verhoeven, A., Striebel, M., Rommes, J., ter Maten, E.J.W., Bechtold, T.: Proper orthogonal decomposition model order reduction of nonlinear IC models. In: Fitt, A.D., Norbury, J., Ockendon, H., Wilson, E. (eds.) Progress in Industrial Mathematics at ECMI 2008, Mathematics in Industry, vol. 15, pp. 441–446. Springer, Berlin/Heidelberg (2010)
46. Verriest, E.I.: Time variant balancing and nonlinear balanced realizations. In: Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.) Model Order Reduction: Theory, Research Aspects and Applications, pp.213–250. Springer, Berlin/Heidelberg (2008)
47. Volkwein, S.: Model reduction using proper orthogonal decomposition . TU Graz (2008). http://www.uni-graz.at/imawww/volkwein/POD.pdf
48. Voß, T.: Model reduction for nonlinear differential algebraic equations. MSc.-thesis, Bergische Universität Wuppertal. Also published as Nat.Lab. Unclassified Report PR-TN-2005/00919, Philips Research (2005)
49. Voß, T., Pulch, R., ter Maten, J., El Guennouni, A.: Trajector piecewise linear aproach for nonlinear differential-algebraic equations in circuit simulation. In: Ciuprina, G., Ioan, D. (eds.) Scientific Computing in Electrical Engineering. Mathematics in Industry, vol. 11, pp. 167–173. Springer, Berlin/New York (2007)
50. Willcox, K., Peraire, J.: Balanced model reduction via the proper orthogonal decomposition. AIAA J. **40**(11), 2323–2330 (2002)
51. Zhou, D., Cai, W.: A fast wavelet collocation method for highspeed circuit simulation. IEEE Trans. Circuits Syst. I Fundam. Theory Appl. **46**(8), 920–930 (1999)
52. Zhou. D., Cai, W., Zhang, W.: An adaptive wavelet method for nonlinear circuit simulation. IEEE Trans. Circuits Syst. I Fundam. Theory Appl. **46**(8), 930–938 (1999)
53. Zong, K., Yang, F., Zeng, X.: A wavelet-collocation-based trajectory piecewise-linear algorithm for time-domain model-order reduction of nonlinear circuits. IEEE. Trans. Circuits Syst. I Regular Papers **57**(11), 2981–2990 (2010)

References for Section 6.2

54. Amestoy, P.R., Davis, T.A., Duff, I.S.: An approximate minimum degree ordering algorithm. SIAM J. Matrix Anal. Appl. **17**(4), 886–905 (1996)
55. Benner, P., Schneider, A.: Model order and terminal reduction approaches via matrix decomposition and low rank approximation. In: Roos, J., Costa, L.R.J. (eds.) Scientific Computing in Electrical Engineering SCEE 2008. Mathematics in Industry, vol. 14, pp. 523–530. Springer, Berlin/Heidelberg (2010)

56. Benner, P., Schneider, A.: On stability, passivity and reciprocity preservation of ESVDMOR. In: [58], pp. 277–287 (2011)
57. Benner, P., Schneider, A.: Model reduction for linear descriptor systems with many ports. In: Günther, M., Bartel, A., Brunk, M., Schöps, S., Striebel, M. (eds.) Progress in Industrial Mathematics at ECMI 2010. Mathematics in Industry, vol. 17, pp. 137–144. Springer, Berlin/New York (2012)
58. Benner, P., Hinze, M., ter Maten, E.J.W. (eds.) Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74. Springer, Dordrecht (2011)
59. Cadence: AssuraRCX. http://www.cadence.com
60. Chua, L.O., Lin, P.: Computer Aided Analysis of Electric Circuits: Algorithms and Computational Techniques, 1st edn. Prentice Hall, Englewood Cliffs (1975)
61. Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introduction to Algorithms, 1st edn. MIT, Cambridge (1990)
62. Davis, T.A.: Suite sparse: a suite of sparse matrix packages. http://www.cise.ufl.edu/research/sparse/SuiteSparse/
63. Duff, I.S., Erisman, A.M., Reid, J.K.: Direct Methods for Sparse Matrices. Clarendon Press, Oxford (1986)
64. Jivaro: available from EdXact SA, Voiron. http://www.edxact.com
65. Electrostatic discharge association. http://www.esda.org
66. Freund, R.W.: SPRIM: Structure-preserving reduced-order interconnect macromodeling. In: Technical Digest of the 2004 IEEE/ACM International Conference on CAD, Los Alamitos, pp. 80–87 (2004)
67. Golub, G.H., van Loan, C.F.: Matrix Computations, 3rd edn. John Hopkins University Press, Baltimore (1996)
68. Ionutiu, R.: Model order reduction for multi-terminal systems – with applications to circuit simulation. Ph.D.-thesis, TU Eindhoven (2011). http://alexandria.tue.nl/extra2/716352.pdf
69. Ionutiu, R., Rommes, J.: Circuit synthesis of reduced order models. Technical report 2008/00316, NXP Semiconductors (2009)
70. Ionutiu, R., Rommes, J.: A framework for synthesis of reduced order models. In: Coupled Multiscale Simulation and Optimization in Nanoelectronics, Ch. 4, Section 4.5, this volume (2015)
71. Ionutiu, R., Rommes, J., Antoulas, A.C.: Passivity Preserving Model Reduction using the Dominant Spectral Zero Method. In: Coupled Multiscale Simulation and Optimization in Nanoelectronics, Ch. 4, Section 4.4, this volume (2015)
72. Karypis, G., Kumar, V.: METIS, A software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices. http://glaros.dtc.umn.edu/gkhome/metis/
73. Kerns, K.J., Yang, A.T.: Stable and efficient reduction of large, multiport networks by pole analysis via congruence transformations. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. (TCAD) **16**(7), 734–744 (1997)
74. Kerns, K.J., Yang, A.T.: Preservation of passivity during RLC network reduction via split congruence transformations. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. (TCAD) **17**(7), 582–591 (1998)
75. Kolyer, J.M., Watson, D.: ESD: From A to Z. Springer, Boston (1996)
76. McCalla, W.J.: Fundamentals of Computer Aided Circuit Simulation, 1st edn. Kluwer Academic, Boston (1988)
77. Miettinen, P., Honkala, M., Roos, J.: Using METIS and hMETIS Algorithms in Circuit Partitioning. Circuit Theory Laboratory Report Series CT-49. Helsinki University of Technology, Espoo (2006)
78. Pstar: In-house industrial circuit simulator. NXP Semiconductors, Eindhoven. http://www.nxp.com
79. Phillips, J.R., Silveira, L.M.: Poor man's tbr: a simple model reduction scheme. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. (TCAD) **24**(1), 283–288 (2005)

80. Rommes, J., Martins, N.: Efficient computation of transfer function dominant poles using subspace acceleration IEEE Trans. Power Syst. **21**(3), 1218–1226 (2006)
81. Rommes, J., Martins, N.: Efficient computation of multivariable transfer function dominant poles using subspace acceleration. IEEE Trans. Power Syst. **21**(4), 1471–1483 (2006)
82. Rommes, J.: Methods for eigenvalue problems with applications in model order reduction. Ph.D.-thesis, Utrecht University (2007). http://igitur-archive.library.uu.nl/dissertations/2007-0626-202553/index.htm
83. Rommes, J., Lenaers, P., Schilders, W.H.A.: Reduction of large resistor networks. In: Roos, J., Costa, L.R.J. (eds.) Scientific Computing in Electrical Engineering 2008. Series Mathematics in Industry, vol. 14, pp. 555–562. Springer, Berlin/Heidelberg (2010)
84. Rommes, J., Schilders, W.H.A.: Efficient methods for large resistor networks. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. (TCAD) **29**(1), 28–39 (2010)
85. Rommes, J., Martins, N.: Model reduction using modal approximation. This COMSON Handbook (2013)
86. Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.): Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 13. Springer, Berlin (2008)
87. Yang, F., Zeng, X., Su, Y., Zhou, D.: RLC equivalent circuit synthesis method for structure-preserved reduced-order model of interconnect in VLSI. Commun. Comput. Phys. **3**(2), 376–396 (2008)
88. Zečević, A.I., Šiljak, D.D.: Balanced decompositions of sparse systems for multilevel prallel processing. IEEE Trans. Circuit Syst. I Fund. Theory Appl. **41**(3), 220–233 (1994)
89. Zhou, Q., Sun, K., Mohanram, K., Sorensen, D.C.: Large power grid analysis using domain decomposition. In: Proceedings of the Design Automation and Test in Europe (DATE), Munich, pp. 27–32 (2006)

References for Section 6.3

90. Adler, R.: A study of locking phenomena in oscillators. Proc. I.R.E. Waves Electron. **34**, 351–357 (1946)
91. Agarwal, S., Roychowdhury, J.: Efficient multiscale simulation of circadian rhythms using automated phase macromodelling techniques. In: Proceedingsof the Pacific Symposium on Biocomputing, Kohala Coast, vol. 13, pp. 402–413 (2008)
92. Aikio, J.P.: Frequency domain model fitting and Volterra analysis implemented on top of harmonic balance simulation. Ph.D.-thesis, Faculty of Technology of the University of Oulu (2007). http://jultika.oulu.fi/Record/isbn978-951-42-8420-5
93. Aikio, J.P., Makitalo, M., Rahkonen, T.: Harmonic load-pull technique based on Volterra analysis. In: European Microwave Conference, Rome, pp. 1696–1699 (2009)
94. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems. SIAM, Philadelphia (2005)
95. Banai, A., Farzaneh, F.: Locked and unlocked behaviour of mutually coupled microwave oscillators. IEE Proc. Antennas Propag. **147**, 13–18 (2000)
96. Benner, P., Mehrmann, V., Sorensen, D. (eds.): Dimension Reduction of Large-Scale Systems. Lecture Notes in Computational Science and Engineering, vol. 45. Springer, Berlin/New York (2005)
97. Benner, P., Hinze, M., ter Maten, E.J.W. (eds.): Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74. Springer, Dordrecht (2011)
98. Boyland, P.L.: Bifurcations of circle maps: Arnol'd tongues, bistability and rotation intervals. Commun. Math. Phys. **106**(3), pp. 353–381 (1986)
99. Brachtendorf, H.G.: Theorie und Analyse von autonomen und quasiperiodisch angeregten elektrischen Netzwerken. Habilitationsschrift, Universität Bremen (2001)

100. Demir, A., Mehrotra, A., Roychowdhury, J.: Phase noise in oscillators: a unifying theory and numerical methods for characterization. IEEE Trans. Circuit Syst. I **47**(5), 655–674 (2000)
101. Demir, A., Long, D., Roychowdhury, J.: Computing phase noise eigenfunctions directly from steady-state jacobian matrices. IEEE/ACM International Conference on Computer Aided Design, ICCAD-2000, San Jose, pp. 283–288 (2000)
102. Galton, I., Towne, D.A., Rosenberg, J.J., Jensen, H.T.: Clock distribution using coupled oscillators. In: IEEE International Symposium on Circuits and Systems, Atlanta, vol. 3, pp. 217–220 (1996)
103. Günther, M.: Simulating digital circuits numerically – a charge-oriented ROW approach. Numer. Math. **79**, 203–212 (1998)
104. Harutyunyan, D., Rommes, J., ter Maten, J., Schilders, W.: Simulation of mutually coupled oscillators using nonlinear phase macromodels. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. (TCAD) **28**(10), 1456–1466 (2009)
105. Günther, M., Feldmann, U., ter Maten, J.: Modelling and discretization of circuit problems. In: [123], pp. 523–659 (2005)
106. Harutyunyan, D., Rommes, J.: Simulation of coupled oscillators using nonlinear phase macromodels and model order reduction. In: [97], pp. 163–175 (2011)
107. Heidari, M.E., Abidi, A.A.: Behavioral models of frequency pulling in oscillators. In: IEEE International Behavioral Modeling and Simulation Workshop, San Jose, pp. 100–104 (2007)
108. Houben, S.H.J.M.: Circuits in motion: the numerical simulation of electrical oscillators. Ph.D.-thesis, Technische Universiteit Eindhoven (2003). http://alexandria.tue.nl/extra2/200310849.pdf
109. Kevenaar, T.A.M.: Periodic steady state analysis using shooting and wave-form-Newton. Int. J. Circuit Theory Appl. **22**(1), 51–60 (1994)
110. Kundert, K., White, J., Sangiovanni-Vincentelli, A.: An envelope-following method for the efficient transient simulation of switching power and filter circuits. In: IEEE International Conference on Computer-Aided Design, ICCAD-88. Digest of Technical Papers, Santa Clara, pp. 446–449 (1988)
111. Lai, X., Roychowdhury, J.: Capturing oscillator injection locking via nonlinear phase-domain macromodels. IEEE Trans. Micro. Theory Tech. **52**(9), 2251–2261 (2004)
112. Lai, X., Roychowdhury, J.: Automated oscillator macromodelling techniques for capturing amplitude variations and injection locking. In: IEEE/ACM International Conference onComputer Aided Design, ICCAD-2004, San Jose, pp. 687–694 (2004)
113. Lai, X., Roychowdhury, J.: Fast and accurate simulation of coupled oscillators using nonlinear phase macromodels. In: 2005 IEEE MTT-S International Microwave Symposium Digest, Long Beach, pp. 871–874 (2005)
114. Lai, X., Roychowdhury, J.: Fast simulation of large networks of nanotechnological and biochemical oscillators for investigating self-organization phenomena. In: Proceedings of the IEEE Asia South-Pacific Design Automation Conference, Yokohama, pp. 273–278 (2006)
115. Maffezzoni, P.: Unified computation of parameter sensitivity and signal-injection sensitivity in nonlinear oscillators. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **27**(5), 781–790 (2008)
116. Maffezzoni, P., D'Amore, D.: Evaluating pulling effects in oscillators due to small-signal injection. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **28**(1), 22–31 (2009)
117. Mathworks: Matlab 7 (2009). http://www.mathworks.com/
118. Moore, B.C.: Principal component analysis in linear systems: controllability, observability and model reduction. IEEE Trans. Autom. Control **26**(1), pp. 17–32 (1981)
119. Pulch, R.: Polynomial chaos expansions for analysing oscillators with uncertainties. In: Troch, I., Breitenecker, F. (eds.) Proceedings MATHMOD 09 Vienna, ARGESIM Report 35 (Full Papers), TU Vienna (2009)
120. Razavi, B.: A study of injection locking and pulling in oscillators. IEEE J. Solid-State Circuit **39**(9), 1415–1424 (2004)
121. Rentrop, P., Günther, M., Hoschek, M., Feldmann, U.: CHORAL—a charge-oriented algorithm for the numerical integration of electrical circuits. In: Jäger, W., Krebs, H.-J. (eds.) Mathematics—Key Technology for the Future, pp. 429–438. Springer, Berlin (2003)

122. Semlyen, A., Medina, A.: Computation of the periodic steady state in systems with nonlinear components using a hybrid time and frequency domain method. IEEE Trans. Power Syst. **10**(3), 1498–1504 (1995)
123. Schilders, W.H.A., ter Maten, E.J.W. (eds.) Numerical Methods in Electromagnetics. Handbook of Numerical Analysis, vol. 13. Elsevier, Amsterdam/Oxford (2005)
124. Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.): Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 13. Springer, Berlin (2008)
125. Stykel, T.: Gramian based model reduction for descriptor systems. Math. Control Signals Syst. **16**, 297–319 (2004)
126. ter Maten, E.J.W., Fijnvandraat, J.G., Lin, C., Peters, J.M.F.: Periodic AC and periodic noise in RF simulation for electronic circuit design. In: Antreich, K., Bulirsch, R., Gilg, A., Rentrop, P. (eds.) Modeling, Simulation and Optimization of Integrated Circuits. International Series of Numerical Mathematics, vol. 146, pp. 121–134. Birkhäuser Verlag, Basel (2003)
127. Wan, Y., Lai, X., Roychowdhury, J.: Understanding injection locking in negative resistance lc oscillators intuitively using nonlinear feedback analysis. In: Proceedings of the IEEE Custom Integrated Circuits Conference, San Jose, pp. 729–732 (2005)
128. Wang, Z., Lai, X., Roychowdhury, J.: PV-PPV: parameter variability aware, automatically extracted, nonlinear time-shifted oscillator macromodels. In: Proceedings of the Design Automation Conference DAC'07, Yokohama, pp. 142–147 (2007)
129. Yanzhu, Z., Dingyu, X.: Modeling and simulating transmission lines using fractional calculus. In: International Conference on Wireless Communications, Networking and Mobile Computing (WiCom 2007), Shanghai, pp. 3115–3118 (2007). doi:10.1109/WICOM.2007.773