Chapter 4
Model Order Reduction: Methods, Concepts and Properties

Athanasios C. Antoulas, Roxana Ionutiu, Nelson Martins, E. Jan W. ter Maten, Kasra Mohaghegh, Roland Pulch, Joost Rommes, Maryam Saadvandi, and Michael Striebel

Abstract This chapter offers an introduction to Model Order Reduction (MOR). It gives an overview on the methods that are mostly used. It also describes the main

A.C. Antoulas (✉)
Department of Electrical and Computer Engineering, William Marsh Rice University, MS-380, PO Box 1892, Houston, TX 77251-1892, USA

School of Engineering and Science, Jacobs University Bremen, 28725 Bremen, Germany
e-mail: A.C.Antoulas@jacobs-university.de; ACA@rice.edu

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

N. Martins
CEPEL, 21941-911, Rio de Janeiro, RJ, Brazil
e-mail: Nelson@cepel.br

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

K. Mohaghegh
Multiscale in Mechanical and Biological Engineering (M2BE), Aragón Institute of Engineering Research (I3A), University of Zaragoza, María de Luna, 3, E-50018 Zaragoza, Spain
Kasra@unizar.es

R. Pulch
Institut für Mathematik und Informatik, Ernst Moritz Arndt Universität Greifswald, Walther-Rathenau-Straße 47, D-17487 Greifswald, Germany
e-mail: PulchR@uni-greifswald.de

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. Günther (ed.), *Coupled Multiscale Simulation and Optimization in Nanoelectronics*, Mathematics in Industry 21,
DOI 10.1007/978-3-662-46672-8_4

concepts behind the methods and the properties that are aimed to be preserved. The sections are in a prefered order for reading, but can be read independentlty. Section 4.1, written by Michael Striebel, E. Jan W. ter Maten, Kasra Mohaghegh and Roland Pulch, overviews the basic material for MOR and its use in circuit simulation. Issues like Stability, Passivity, Structure preservation, Realizability are discussed. Projection based MOR methods include Krylov-space methods (like PRIMA and SPRIM) and POD-methods. Truncation based MOR includes Balanced Truncation, Poor Man's TBR and Modal Truncation.

Section 4.2, written by Joost Rommes and Nelson Martins, focuses on Modal Truncation. Here eigenvalues are the starting point. The eigenvalue problems related to large-scale dynamical systems are usually too large to be solved completely. The algorithms described in this section are efficient and effective methods for the computation of a few specific dominant eigenvalues of these large-scale systems. It is shown how these algorithms can be used for computing reduced-order models with modal approximation and Krylov-based methods.

Section 4.3, written by Maryam Saadvandi and Joost Rommes, concerns passivity preserving model order reduction using the spectral zero method. It detailedly discusses two algorithms, one by Antoulas and one by Sorenson. These two approaches are based on a projection method by selecting spectral zeros of the original transfer function to produce a reduced transfer function that has the specified roots as its spectral zeros. The reduced model preserves passivity.

Section 4.4, written by Roxana Ionutiu, Joost Rommes and Athanasios C. Antoulas, refines the spectral zero MOR method to dominant spectral zeros. The new model reduction method for circuit simulation preserves passivity by interpolating dominant spectral zeros. These are computed as poles of an associated Hamiltonian system, using an iterative solver: the subspace accelerated dominant pole algorithm (SADPA). Based on a dominance criterion, SADPA finds relevant spectral zeros and the associated invariant subspaces, which are used to construct the passivity preserving projection. RLC netlist equivalents for the reduced models are provided.

Section 4.5, written by Roxana Ionutiu and Joost Rommes, deals with synthesis of a reduced model: reformulate it as a netlist for a circuit. A framework for model reduction and synthesis is presented, which greatly enlarges the options for the re-use of reduced order models in circuit simulation by simulators of choice. Especially when model reduction exploits structure preservation, we show that using the model as a current-driven element is possible, and allows for synthesis without controlled

M. Saadvandi
Numerical Approximation and Linear Algebra Group, Department of Computer Science, KU Leuven, Celestijnenlaan 200A, 3001 Heverlee, Belgium Maryam.Saadvandi@cs.kuleuven.be

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

sources. Two synthesis techniques are considered: (1) by means of realizing the reduced transfer function into a netlist and (2) by unstamping the reduced system matrices into a circuit representation. The presented framework serves as a basis for reduction of large parasitic R/RC/RCL networks.

Co-operations Between the Various Co-authors

The subactivity on Model Order Reduction (MOR) of the COMSON project[1] was greatly influenced by interaction with additional research on MOR, first at Philips Research Laboratories and (from october 2006 on) at NXP Semiconductors (both in Eindhoven). There was direct project work with the TU Eindhoven, with the Bergische Universität Wuppertal and with the Royal Institute of Technology (KTH) in Stockholm:

- R. IONUTIU: *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.
- M. SAADVANDI: *Passivity preserving model reduction and selection of spectral zeros*. MSc. Thesis, Royal Institute of Technology (KTH), Stockholm. Also published as Technical Note NXP-TN-2008/00276, Unclassified Report, NXP Semiconductors, Eindhoven, 2008. [In September 2012, Maryam Saadvandi did complete a Ph.D.-Thesis at KU Leuven, Belgium, on *Nonlinear and parametric model order reduction for second order dynamical systems by the dominant pole algorithm*.]
- M.V. UGRYUMOVA: *Applications of Model Order Reduction for IC Modeling*. Ph.D.-Thesis, TU Eindhoven, 2011, http://alexandria.tue.nl/extra2/711015.pdf.
- A. VERHOEVEN: *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.
- T. VOSS: *Model reduction for nonlinear differential algebraic equations*, MSc. Thesis, University of Wuppertal, 2005. Unclassified Report PR-TN-2005/00919, Philips Research Laboratories, September 2005.
  [Afterwards, Thomas Voß did complete a Ph.D.-Thesis at the Rijksuniversiteit Groningen, the Netherlands, on *Port-Hamiltonian modeling and control of piezoelectric beams and plates: application to inflatable space structures*, 2010, http://catalogus.rug.nl/DB=1/SET=1/TTL=4/REL?PPN=326-918639.]

[1] *Coupled Multiscale Simulation and Optimization in Nano-electronics, COMSON* – EU-FP6 MCA-RTN Research and Training Network Project, 2006–2010, http://www.comson.eu/.

Here Roxana Ionutiu was partially funded by the COMSON project. Apart from TU Eindhoven she also worked with Thanos Antoulas at the Jacobs University in Bremen. Roxana Ionutiu appears several times as co-author in this chapter and in the following ones. Also Maryam Saadvandi appears as co-author of a section. Work by the others is found in the reference lists at each section.

Parallel to the COMSON Project research on MOR was done within the O-MOORE-NICE! project.[2] The Marie Curie Fellows, Luciano De Tommasi (University of Antwerp), Davit Harutyunyan (TU Eindhoven), Joost Rommes (NXP Semiconductors), and Michael Striebel (Chemnitz University of Technology), interacted actively with the COMSON PhD-students. They contribute to several sections as co-authors, together with researchers from the staff from NXP Semiconductors (Eindhoven), TU Eindhoven, Bergische Universität Wuppertal and the Politehnica Univ. of Bucharest.

The Politehnica Univ. of Bucharest greatly acknowledges co-operation with Jorge Fernandez Villena and Luis Miguel Silveira of INESC-ID in Lisbon. They appear as co-author in the next chapter. Jorge Fernandez Villena was partially funded by the COMSON project. Work in Bucharest and in Lisbon also did benefit from financial support during earlier years from the following complementary projects: FP6/Chameleon, FP5/Codestar, CEEX/nEDA, UEFISCSU/IDEI 609/16.01.2009 and POSDRU/89/1.5/S/62557.

The fourth co-author acknowledges the ENIAC JU Project /2010/SP2(Wireless communication)/270683-2 Artemos, *Agile Rf Transceivers and front-Ends for future smart Multi-standard cOmmunications applicationS*, http://.artemos.eu.

The COMSON project did directly lead to four Ph.D.-Theses on MOR-related topics:

- Z. ILIEVSKI: *Model order reduction and sensitivity analysis*. Ph.D.-Thesis, TU Eindhoven, 2010, http://alexandria.tue.nl/extra2/201010770.pdf.
- S. KULA: *Reduced order models of interconnects in high frequency integrated circuits*. Ph.D.-Thesis, Politehnica Univ. of Bucharest, 2009.
- K. MOHAGHEGH: *Linear and nonlinear model order reduction for numerical simulation of electric circuits*. Ph.D.-Thesis, Bergische Universität Wuppertal, Germany. Available at Logos Verlag, Berlin, Germany, 2010.
- A. ŞTEFĂNESCU: *Parametric models for interconnections from analogue high frequency integrated circuits*. Ph.D.-Thesis, Politehnica Univ. of Bucharest, 2009.

[2] *Operational MOdel Order REduction for Nanoscale IC Electronics (O-MOORE-NICE!)* – EU-FP6 MCA-ToK Transfer of Knowledge Project, 2007–2010, http://www.tu-chemnitz.de/mathematik/industrie_technik/projekte/omoorenice/index.php?lang=en

4.1 Circuit Simulation and Model Order Reduction

Speaking of "circuit models", we refer to models of electrical circuits derived by a network approach.[3] In circuit simulation the charge-oriented modified nodal analysis (MNA) is a prominent representative of network approaches used to automatically create mathematical models for a physical electrical circuit. In the following we give a short introduction to circuit modeling with MNA. For a detailed discussion we refer to [22].

In charge-oriented MNA, voltages, currents, electrical charges and magnetic fluxes are the quantities that describe the activity of a circuit. The electrical circuit to be modelled is considered to be an aggregation of basic network elements: the ohmic resistor, capacitor, inductor, voltage source and current source. Real physical circuit elements, especially semiconductor devices, are replaced by idealised network elements or so-called "companion models". The basic network elements correlate the network quantities. Each basic element is associated to a *characteristic equations*:

- Resistor: $I = r(U)$ (linear case: $I = \frac{1}{R} \cdot U$)
- Capacitor: $I = \dot{q}$ with $q = q_C(U)$ (linear case: $I = C \cdot \dot{U}$)
- Inductor: $U = \dot{\phi}$ with $\phi = \phi_L(I)$ (linear case: $U = L \cdot \dot{I}$)
- Voltage source: $U = v(t)$ (controlled source: $U = v(U_{\text{ctrl}}, I_{\text{ctrl}}, t)$)
- Current source: $I = \imath(t)$ (controlled source: $I = \imath(U_{\text{ctrl}}, I_{\text{ctrl}}, t)$)

where U is the voltage drop across the element's terminal, I is the current flowing through the element, q is the electric charge stored in a capacitor and ϕ is the magnetic flux of an inductor. The dot $\dot{}$ on top of a quantity indicates the usual time derivative d/dt on that quantity.

All wires, connecting the circuit elements are considered to be electrically ideal, i.e., no wire possesses any resistance, capacitance or inductance. Thereby, also the volume expansion of the circuit becomes irrelevant, the electrical system is considered being a lumped circuit.The circuit's layout, defined by the interconnects between elements, is thus reduced to its conceptional structure, which is called *network topology*.

The network's topology consists of *branches* and *nodes*. Each network element is regarded as a branch of the circuit and its terminals are the nodes by which it is connected to other elements. Assigning a direction to each branch – the direction of the current traversing the corresponding element – and a serial number to each node, we end up with a *directed graph* representing the network. As any directed graph, the network can be described by an *incidence matrix* A. This matrix has as

[3] Section 4.1 has been written by: Michael Striebel, E. Jan W. ter Maten, Kasra Mohaghegh and Roland Pulch. For additional details we refer to the Ph.D.-Thesis [33] of the third author.

many columns as there are branches, i.e., elements and as many rows as there are nodes in the circuit. Each column of the matrix has one entry $+1$ and one entry -1, displaying the start and end point of the branch. As all other entries are 0, the matrix A is sparse.

Usually, one circuit node is tagged as *ground node*. As a consequence, each branch voltage U between two nodes l an m can be expressed by the two *node voltages* e_l and e_m, which are the voltage differences between each node and the ground node. From this agreement, the node voltage of the ground node is constantly 0 and therefore the information stored in the corresponding row of the incidence matrix becomes redundant and this very row can be removed. Hence, frequently by the term *incidence matrix*, one refers to the reduced matrix $\mathbf{A}$, given by removing the row corresponding to the ground node.

As each branch of the network represents one of the five basic network element resistor (R), capacitor (C), inductor (L), voltage and current source (V and I, respectively), the indicence matrix can be described as an assembly of element related incidence matrices:

$$\mathbf{A} = (\mathbf{A}_C, \mathbf{A}_R, \mathbf{A}_L, \mathbf{A}_V, \mathbf{A}_I)\,,$$

with $A_\Omega \in \{0, +1, -1\}^{n_e \times n_\Omega}$ for $\Omega \in \{C, R, L, V, I\}$. Here, n_e is the number of nodes (without the ground node) and $n_C, \ldots, n_I$ are the cardinalities of the sets of the different basic elements' branches.

The *Kirchhoff's laws*, which relate the branch voltages in a loop and the currents accumulating in a node, namely *Kirchhoff's voltage law* and *Kirchhoff's current law*, respectively, are the final component for setting up the *MNA network equations*:

$$\mathbf{A}_C \frac{d}{dt}\mathbf{q} + \mathbf{A}_R \mathbf{r}(\mathbf{A}_R^T \mathbf{e}) + \mathbf{A}_L \boldsymbol{\imath}_L + \mathbf{A}_V \boldsymbol{\imath}_V + \mathbf{A}_I \imath(t) = \mathbf{0}, \tag{4.1a}$$

$$\frac{d}{dt}\boldsymbol{\phi} - \mathbf{A}_L^T \mathbf{e} = \mathbf{0}, \tag{4.1b}$$

$$\mathbf{v}(t) - \mathbf{A}_V^T \mathbf{e} = \mathbf{0}, \tag{4.1c}$$

$$\mathbf{q} - \mathbf{q}_C(\mathbf{A}_C^T \mathbf{e}) = \mathbf{0}, \tag{4.1d}$$

$$\boldsymbol{\phi} - \boldsymbol{\phi}_L(\boldsymbol{\imath}_L) = \mathbf{0}. \tag{4.1e}$$

It is worthwile to highlight the subequations (4.1a) and (4.1c). The former is the personification of Kirchhoff's current law, stating that for each network node the sum of branch currents meeting is identically zero. The latter reflects the functionality of voltage sources: dictating branch voltages.

The unknowns $\mathbf{q}, \boldsymbol{\phi}, \mathbf{e}, \boldsymbol{\imath}_L, \boldsymbol{\imath}_V$, i.e., the charges, fluxes, node voltages and currents traversing inductors and voltage sources, respectively – each of them functions of time t – are combined to the *state vector* $\mathbf{x}(t) \in \mathbb{R}^n$ of unknowns, of dimension $n = n_C + n_L + n_e + n_L + n_V$. Then, the network equations (4.1) can be stated in a compact form:

$$\frac{d}{dt}\mathbf{q}(\mathbf{x}(t)) + \mathbf{j}(\mathbf{x}(t)) + \mathbf{B}\mathbf{u}(t) = \mathbf{0}, \tag{4.2}$$

where $\mathbf{q}, \mathbf{j} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ describe the contribution of reactive and nonreactive elements, respectively.[4] The excitations defined by the voltage- and current-sources are combined to the vector $\mathbf{u}(t) \in \mathbb{R}^m$ with $m = n_V + n_I$. The excitations are assigned to the corresponding nodes and branches by the matrix $\mathbf{B} \in \mathbb{R}^{n \times m}$.

If the circuit under considerations contains only elements with a linear characteristic equation, the network equations can be written as[5]

$$\mathbf{E}\dot{\mathbf{x}}(t) + \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) = \mathbf{0}, \tag{4.3a}$$

with

$$\mathbf{E} = \begin{pmatrix} \mathbf{A}_C \mathscr{C} \mathbf{A}_C^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathscr{L} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} \mathbf{A}_R \mathscr{G} \mathbf{A}_R^T & \mathbf{A}_L & \mathbf{A}_V \\ -\mathbf{A}_L^T & \mathbf{0} & \mathbf{0} \\ -\mathbf{A}_V^T & \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} \mathbf{A}_I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n_V} \end{pmatrix}, \tag{4.3b}$$

where $\mathscr{C}, \mathscr{L}, \mathscr{G}$ are basically diagonal matrices containing the individual capacitors, inductances and conductances (inverse resistances) of the basic network elements. $\mathbf{I}_{n_V}$ is the identity matrix in $\mathbb{R}^{n_V \times n_V}$.

We arrive at this formulation by eliminating the charges and fluxes. Hence the unknown state vector here is $\mathbf{x} = (\mathbf{e}^T, \boldsymbol{\imath}_L^T, \boldsymbol{\imath}_V^T)^T$ and the excitation vector is $\mathbf{u} = (\boldsymbol{\imath}_I^T, \mathbf{v}^T)^T$.

It is straightforward to see that the structure of the matrices $\mathbf{E}, \mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{B} \in \mathbb{R}^{n \times m}$ is determined by the element related incidence matrices $\mathbf{A}_C, \mathbf{A}_R, \mathbf{A}_L, \mathbf{A}_V, \mathbf{A}_I$. As there is usually only a week linkage amongst the network node, i.e., nodes are connected directly to only a few other nodes, these incidence matrices are sparse and so are the system matrices in (4.3a) and the Jacobian matrices $d\mathbf{q}/d\mathbf{x}, d\mathbf{j}/d\mathbf{x} \in \mathbb{R}^{n \times n}$ of the element functions in (4.2), respectively.

[4] Note that the meaning $\mathbf{q}$ in (4.1) and (4.2) is different: in the prior it is an unknown, in the latter it is a mapping.

[5] Note that $\mathbf{A}$ in (4.3a) does not refer to the incidence matrix $\mathbf{A}$. Furthermore the composition of the unknown $\mathbf{x}$ in (4.2) and (4.3a) can be different. In the latter, taking into account the linear characteristics for capacitors and inductors, the time derivatives of the charges and fluxes can be expressed by the time derivative of the node volages $\mathbf{e}$ and the inductor current $\boldsymbol{\imath}_L$ directly. In this case the unknown state vector amounts to $\mathbf{x} = (\mathbf{e}^T, \boldsymbol{\imath}_L^T, \boldsymbol{\imath}_V^T)^T \in \mathbb{R}^n$ with $n = n_e + n_V + n_L$.

In general, real circuit designs contain a large number of transistors. In the course of setting up the network equations such semiconductor devices are replaced by companion models that consist of a larger number of the basic network elements. Here especially resistors with nonlinear characteristics emerge. Hence, the “mathematical image” of an integrated circuit is usually a nonlinear network equation of the form (4.2).

However, also linear network equations of the form (4.3a) are fundamental problems in the design process. As mentioned above, one disregards the volume extension of a circuit and considers wires as electrically ideal. At the end of the design process, however, there will be a physical integrated circuit. Even on the smallest dies there are kilometers of wiring. These wires do have an electric resistance. As the actual devices are getting small and smaller, capacitive effects introduced by neighbouring wires can not be neglected just as little as inductive effects arising from increasing clock rates.

In fact these issues are not neglected. At least at the end of the design process, when the layout of the chip has to be determined these effects are taken into account. In the *parasitic extraction* from the routing on the chip an artificial linear network is extracted which again is assumed to be a lumped and comprise of ideal wires. However, the resistances, capacitances and inductances that are present there describe the effects caused by the wiring on the actual circuit. A characteristic of these artificial networks is their large dimension: here n can easily be in the range of 10^6.

The impact of the effects on the behaviour of the actual circuit are accounted for by coupling the linear parasitic model to the underlying circuit design.

If the electrical circuit comprises reactive elements, i.e., capacitors and inductors, the network equation (4.2) or (4.3a), respectively, forms a dynamical problem for the unknown state vector $\mathbf{x}$. Usually, however, the system matrix $\mathbf{E}$, or the Jacobian $d\mathbf{q}/d\mathbf{x}$, respectively, does not have full rank.[6] Dynamical systems with this property, i.e., systems consisting of differential and algebraic equations are called *differential algebraic equations (DAE)*, or *descriptor systems*. DAEs differ in several senses from purely differential equations, causing problems in various aspects. A requirement for the solvability of the network equation is the regularity of the matrix pencil $\{\mathbf{E}, \mathbf{A}\}$. The matrix pencil is called regular, if the polynomial $\det(\lambda\mathbf{E} + \mathbf{A})$ does not vanish identically. Otherwise $\{\mathbf{E}, \mathbf{A}\}$ is called *singular matrix pencil*. Then a normal initial-value problem for the linear DAE (4.3a) has none or infinitely many solutions. The regularity of the matrix pencil can be checked by examining the element related incidence matrices [15].

[6] This is easy to see from inspecting the first subequation – the node-current relation – of the MNA equation (4.1): a network node for instance, that is not the starting or end point of a capacitor branch causes a row equal to zero in the incidence matrix $\mathbf{A}_C$ and therefore the node-current relation for that node is an algebraic equation only.

In the context of numerical time integration, needed to solve the network problem in time domain, worthwhile stressing that the initial value has to be chosen properly – $\mathbf{x}(0)$ has to satisfy the algebraic constraints – and that numerical perturbations can be amplified dramatically. Hence, numerical methods have to match the requirements posed by the differential-algebraic structure.

For a detailed analysis of DAEs we refer to the textbook [29]. A detailed discussion of solving DAEs can be found in the textbook [24].

4.1.1 Input-Output Systems in Circuit Simulation

We recall that the origin of the network equations in nonlinear or linear form is a real circuit design, ment to be simulated, i.e., tested with respect to its performance under different circumstances. Nowadays, complex integrated circuits are usually not designed from scratch by a single engineer. In fact, large electrical circuits are usually developed in a modular way. In radio frequency applications, for instance, analogue and digital subcircuits are connected to each other. In general several subunits of different functionality, e.g., one providing stable oscillations another one amplifying a signal, are developed separately and glued together. Hence, subunits possess a way of communication with other subunits, the environment they are embedded in.

To allow for a communication with an environment, the network model (4.2) (or (4.3a)) has to be augmented and transfered to a system that can receive and transmit information. Abstractly, the output of a system can be defined as a function of the state and the input:

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) \in \mathbb{R}^p .$$

In circuit simulation, however, usually the output is a linear relation of the form:

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t),$$

with the *output matrix* $\mathbf{C} \in \mathbb{R}^{p\times n}$ and the *feedthrough matrix* $\mathbf{D} \in \mathbb{R}^{p\times m}$.

The excitation, we mentioned above, i.e., the last term $\mathbf{B}\mathbf{u}(t)$ in the network model (4.2) (or (4.3a)) can be understood as information imposed on the system, in the form of branch currents and node voltages. Therefore we call $\mathbf{u}(t)$ the *input* and $\mathbf{B}$ the *input matrix* to the system.

Hence, an *input-output system* in electrical circuit simulation is given in the form

$$\mathbf{0} = \mathbf{E}\dot{\mathbf{x}}(t) + \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \tag{4.4a}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \tag{4.4b}$$

if only linear elements form the system. If also nonlinear elements are present, we arrive at systems of the form:

$$\mathbf{0} = \frac{d}{dt}\mathbf{q}(\mathbf{x}(t)) + \mathbf{j}(\mathbf{x}(t)) + \mathbf{B}\mathbf{u}(t), \tag{4.5a}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t). \tag{4.5b}$$

The input to state mapping (4.4a) and (4.5a), respectively, is a relation defined by a dynamical system. Therefore, the representation of the input-output system (4.4) and (4.5), respectively, is said to be given in *state space formulation*. The dimension n of the state space is referred to as the *order* of the system.

Frequently the state space formulation in circuit design exhibits a special structure.

- Often there is no direct feedthrough of the input to the output, i.e.

$$\mathbf{D} = \mathbf{0} \in \mathbb{R}^{p\times m}\,. \tag{4.6a}$$

- We often observe

$$p \equiv m \quad \text{and} \quad \mathbf{C} = \mathbf{B}^T \in \mathbb{R}^{m\times n}\,. \tag{4.6b}$$

In full system simulation, individual subcircuit models are connected to each other. To allow for an information exchange, done in terms of currents and voltages, each subcircuit possesses a set of terminals – a subset of the unit's pins.

From a subcircuit's point of view incoming information is either a current being added to or a voltage drop being imposed to the terminal nodes. The former corresponds to adding a current source term to (4.1a), the latter corresponds to adding a voltage source to (4.1c). Information returned by the subsystem is the voltage at the terminal node in the former case or the current traversing that artificial voltage source in the latter case. Having a detailed look at the MNA network equations (4.1) and the composition of the state vector $\mathbf{x}(t)$, it is easy to understand that in this setup, assuming that there are no additional sources in the subcircuits, the output matrix is the transpose of the input matrix.

4.1.2 *The Need for Model Order Reduction*

Clearly, mathematical models for a physical circuit are extracted for a purpose. In short, the manufacturing process of an electrical circuit starts with an idea of what the physical system should do and ends with the physical product. In-between there is a, usually iterative, process of conceptual designing the circuit in the form of a circuit schematic, that comprises parameters defining the layout and nominal values of circuit elements and, choosing the parameters, testing the design, adapting the parameter, . . . , etc.

Testing the design means to analyse its behaviour. There are several types of analysis we briefly want to mention in the following. For a more detailed discussion we refer to [22].

- *Static (DC) analysis* searches for the point to which the system settles in an equilibrium or rest condition. This is characterised by $d/dt\,\mathbf{x}(t) = 0$.
- *Transient analysis* computes the response $\mathbf{y}(t)$ to the time varying excitation $\mathbf{u}(t)$ as a function of time.
- *(Periodic) steady-state analysis*, also called *frequency response analysis*, determines the response of the system in the frequency domain to an oscillating, i.e., sinusoidal input signal.
- *Modal analysis* finds the system's natural vibrating frequency modes and their corresponding modal shapes;
- *Sensitivity analysis* determines the changes of the time-domain response and/or the frequency-domain response to variations in the design parameters.

Transient analysis is run in the time domain. Here the challenge is to numerically integrate a very high-dimensional DAE problem.

Both the frequency response and the modal analysis are run in the frequency domain. Hence, a network description in the frequency domain is needed. As this is basically defined only for linear systems[7] we concentrate on linear network problems of the form (4.4). The *Laplace transform* is the tool to get from the time to the frequency domain.

Recall that for a function $f : [0, \infty) \to \mathbb{C}$ with $f(0) = 0$, the Laplace transform $F : \mathbb{C} \to \mathbb{C}$ is defined by

$$F(s) := \mathscr{L}\{f\}(s) = \int_0^\infty f(t)e^{-st}\,dt.$$

For a vector-valued function $\mathbf{f} = (f_1, \dots, f_q)^T$, the Laplace transform is defined component-wise: $\mathbf{F}(s) = (\mathscr{L}\{f_1\}(s), \dots, \mathscr{L}\{f_q\}(s))^T$.

The physically meaningful values of the complex variable s are $s = i\omega$ where $\omega \geq 0$ is referred to as the *(angular) frequency*. Taking the Laplace transform of the time domain representation of the linear network problem (4.4) we obtain the following frequency domain representation:

$$\begin{aligned} \mathbf{0} &= s\mathbf{E}\mathbf{X}(s) + \mathbf{A}\mathbf{X}(s) + \mathbf{B}\mathbf{U}(s), \\ \mathbf{Y}(s) &= \mathbf{C}\mathbf{X}(s) + \mathbf{D}\mathbf{U}(s), \end{aligned} \tag{4.7}$$

[7] Applying those types of analysis to nonlinear problems involves a linearisation about some point of interest $\mathbf{x}$ in the state space.

where $\mathbf{X}(s), \mathbf{U}(s), \mathbf{Y}(s)$ are the Laplace transforms of the states, the input and the output, respectively. Note that we assumed zero initial conditions, i.e., $\mathbf{x}(0) = \mathbf{0}$, $\mathbf{u}(0) = \mathbf{0}$ and $\mathbf{y}(0) = \mathbf{0}$.

Eliminating the variable $\mathbf{X}(s)$ in the frequency domain representation (4.7) we see that the system's response to the input $\mathbf{U}(s)$ in the frequency domain is given by

$$\mathbf{Y}(s) = \mathbf{H}(s)\mathbf{U}(s)$$

with the matrix-valued *transfer function*

$$\mathbf{H}(s) = -\mathbf{C}\left(s\mathbf{E} + \mathbf{A}\right)^{-1}\mathbf{B} + \mathbf{D} \quad \in \mathbb{C}^{p\times m}. \tag{4.8}$$

The evaluation of the transfer function is the key to the frequency domain based analyses, i.e., the steady-state analysis and the modal frequency analysis. The key to the evaluation of the transfer function, in turn, is the solution of a linear system with the system Matrix $(s\mathbf{E} + \mathbf{A}) \in \mathbb{C}^{n\times n}$.[8]

Note that at the very core of any numerical time integration scheme applied in transient simulation we have to solve as well linear equations with system matrices of the form $\alpha\mathbf{E} + \mathbf{A}$ were $\alpha \in \mathbb{R}$ depends on some coefficient characteristic to the method and the stepsize used.

It is the order n of the problem, i.e., the dimension of the state space that determines how much computational work has to be spend to compute the p output quantities. Usually, the order n in circuit simulation is very large, whereas the dimension of the output is rather small.

The idea of *model order reduction (MOR)* is to replace the high dimensional problem by one of reduced order such that the reduced order model produces an output similar to the output of the original problem when excited with the same input.

Before we give an overview of some of the most common MOR techniques we specify the requirement a reduced order model should satisfy. Again, we just briefly describe some concepts. For a more detailed discussion we refer to the textbook [1].

4.1.2.1 Approximation

The output of the ersatz model should approximate the output of the original model for the same input signal. There are various measures for "being an approximation". In fact these different viewpoints form the basis for different reduction strategies.

[8] Note that here we see the necessity of $\{\mathbf{E}, \mathbf{A}\}$ being a regular matrix pencil.

We give first a Theorem (for an ODE) that confirms how an approximation in the frequency domain leads to an accurate result in the time domain. Let $I_\omega \subset \mathbb{R}$ be a closed interval (but may be $I_\omega = \{\omega_0\}$ or $I_\omega = \mathbb{R}$). For convenience we assume single input $u(t)$ and single output $y(t)$, with transfer function $H(s)$ in the frequency domain between the Laplace transforms $U(s)$ and $Y(s)$. Let $\tilde{H}(s)$ be the approximation to $H(s)$ which gives $\tilde{Y}(s) = \tilde{H}(s)\ U(s)$ and $\tilde{y}(t)$ as the output approximation in the time domain.

Theorem 4.1 *Let $||u(t)||_{L^2([0,\infty))} < \infty$ and $U(i\omega) = 0$ for $\omega \notin I_\omega$. If the system (4.4a) consists of ODEs, then we have the estimate*

$$\max_{t>0} |y(t) - \tilde{y}(t)| \leq (\frac{1}{2\pi} \int_{I_\omega} |H(i\omega) - \tilde{H}(i\omega)|^2 d\omega)^{\frac{1}{2}} (\int_0^\infty |u(t)|^2 dt)^{\frac{1}{2}}. \quad (4.9)$$

Proof We obtain by using the Cauchy-Schwarz inequality in $L^2(I_\omega)$

$$\begin{aligned}
\max_{t>0} |y(t) - \tilde{y}(t)| &\leq \max_{t>0} |\frac{1}{2\pi} \int_{\mathbb{R}} (Y(i\omega) - \tilde{Y}(i\omega)) e^{i\omega t} d\omega| \\
&\leq \max_{t>0} \frac{1}{2\pi} \int_{\mathbb{R}} |Y(i\omega) - \tilde{Y}(i\omega)| \cdot |e^{i\omega t}|\ d\omega \\
&= \frac{1}{2\pi} \int_{\mathbb{R}} |H(i\omega) - \tilde{H}(i\omega)| \cdot |U(i\omega)|\ d\omega \\
&= \frac{1}{2\pi} \int_{I_\omega} |H(i\omega) - \tilde{H}(i\omega)| \cdot |U(i\omega)|\ d\omega \\
&\leq \frac{1}{2\pi} (\int_{I_\omega} |H(i\omega) - \tilde{H}(i\omega)|^2 d\omega)^{\frac{1}{2}} (\int_{I_\omega} |U(i\omega)|^2\ d\omega)^{\frac{1}{2}} \\
&\leq (\frac{1}{2\pi} \int_{I_\omega} |H(i\omega) - \tilde{H}(i\omega)|^2 d\omega)^{\frac{1}{2}} (\int_0^\infty |u(t)|^2 dt)^{\frac{1}{2}}.
\end{aligned}$$

This completes the proof. □

We note that for $I_\omega = \mathbb{R}$ the above error estimate is already found in [23], also for parameterized problems. In [42] the more general case I_ω is considered and applied to Uncertainty Quantification for parameterized problems. In MOR the error estimate becomes often small in an interval I_ω sufficiently close to the used expansion point.

Besides producing similar outputs, the reduced order model should behave similar to the original model in various aspects, which we discuss next.

4.1.2.2 Stability

One of the principal concepts of analyzing dynamical systems is its stability.

An autonomous dynamical system, i.e., a system without input is called *stable* if the solution trajectories are bounded in the time domain. For a linear autonomous system the system matrices determine whether it is stable or unstable. Considering for instance the network equation (4.3a) with $\mathbf{B} \equiv \mathbf{0}$ we have to calculate the generalized eigenvalues[9] $\{\lambda_i(\mathbf{A}, -\mathbf{E}), i = 1, \ldots, n\}$ of the matrix pair $(\mathbf{A}, -\mathbf{E})$ to decide whether or not the system is stable. The system is stable if, and only if, all generalized eigenvalues have non-positive real parts and all generalized eigenvalues with $\mathrm{Re}(\lambda_i(\mathbf{A}, -\mathbf{E})) = 0$ are simple.

4.1.2.3 Passivity

For input-output systems of the form (4.4), stability is not strong enough. If nonlinear components are connected to a stable system it can become unstable.

For square systems, i.e., system where the number of inputs is equal to the number of outputs, $p = m$, a property called *passivity* can be defined. This property is much stronger than stability: it means that a system is unable to generate energy.

Here, an inspection of the system's transfer function yields evidence if the system is passive or not. A necessary and sufficient conditions for a square system to be passive is that the transfer function is *positive real*. This means that

- $\mathbf{H}(s)$ is analytic for $\mathrm{Re}(s) > 0$;
- $\mathbf{H}(\bar{s}) = \overline{\mathbf{H}(s)}$, for all $s \in \mathbb{C}$;
- The Hermitian part of $\mathbf{H}(s)$ is symmetric positive, i.e.: $\mathbf{H}^H(s) + \mathbf{H}(s) \geq 0$, for all s with $\mathrm{Re}(s) > 0$ [50]. Here ^{H} means the transposed conjugate complex: $\mathbf{A}^H = \bar{\mathbf{A}}^T$.

The second condition is satisfied for real systems and the third condition implies the existence of a rational function with a stable inverse. Any congruence transformation applied to the system matrices satisfies the previous conditions if the original system satisfies them, and so preserves passivity of the system if the following conditions are true:

- The system matrices are positive definite, $\mathbf{E}$, $\mathbf{A} \geq 0$.
- $\mathbf{B} = \mathbf{C}^T, \mathbf{D} = 0$.

These conditions are sufficient, but not necessary. They are usually satisfied in the case of electrical circuits, which makes congruence-based projection methods very popular in circuit simulation.

[9] For a matrix pair $(\mathbf{A}, \mathbf{B})$ λ is a generalized eigenvalue with a generalized eigenvector $\mathbf{v}$, if $\mathbf{Av} = \lambda\mathbf{Bv}$.

4.1.2.4 Structure Preservation

For the case of having a circuit made up of linear elements only we have seen in (4.3b) that the system matrices exhibit a block structured form. Furthermore we recognized that the system matrices are sparse. In fact, the same properties hold for the linear case (4.3a) also.

As a consequence, the matrices of the form $(\xi \mathbf{E} + \mathbf{A})$ that have to be decomposed during the different modes of analysis exhibit already a form that can be exploited when solving the corresponding linear systems.

If the full system (4.4) is replaced by a small dimensional system, it would be most desirable if that ersatz system again has a structure similar to the structure of the full problem. Namely, a block structure should be preserved and the system matrix arising from the reduced order model should be sparse as well, as it can be more expensive to decompose a small dense matrix then a larger sparse one.

4.1.2.5 Realizability

Preserving the block structure, as just mentioned, is crucial for *realizing* a reduced order model again as an RLC-circuit again. Another prerequisit for a reduced order model to be synthesizable is *reciprocity*.[10] This is a special form of symmetry of the transfer function **H**. We will not give details here but refer to [44] for a precise definition and MOR techniques and to [6] for other reciprocity preserving MOR techniques.

There is an ongoing discussion if it is necessary to execute this realization (also referred to as *un-stamping*). It is worthwhile mentioning two benefits of that

- An industrial circuit simulator does in fact never create the MNA equations. Actually, a circuit is given in the form of a netlist, i.e., a table where each line correspond to one element. Each time a system has to be solved, the simulator runs through that list, evaluates each element and stamps the corresponding value in the correct places of the system matrix and the corresponding right-hand side. If a reduced order model is available in the form of such a table as well, the simulator can treat that ersatz model like any other subcircuit and does not have to change to a different mode of including the contribution of the subsystem to the overall system.
- A synthezised reduced order model can provide more insight to the engineers and designers than the reduced order model in mathematical form [52].

[10] A two-terminal element is said to be reciprocal, if a variation of the values of one terminal immediately has the reverse effect on the other terminal's value. Linear characteristics obviously have this property.

4.1.3 MOR Methods

We recall the idea of model order reduction (MOR):

Replace a high dimensional problem, say of order n by one of reduced order $r \ll n$ such that the two input-output systems produce a similar output when excited with the same input. Furthermore the reduced order problem should conserve the characteristics of the full model it was derived from.

In fact there is a need for MOR techniques in various fields of applications and for different kind of problem structures. Although a lot of effort is being spent on deriving reliable MOR methods for, e.g., nonlinear problems of the form (4.5) and for linear time varying (LTV) problems – these are problems of the form (4.4) where the system matrices $\mathbf{E}, \mathbf{A}, \ldots$ depend on time t – MOR approaches for linear time systems, or, more precisely, for linear time invariant (LTI) systems, are best understood and are technically mature.

The outcome of MOR applied to the linear state space problem (4.4) is an ersatz system of the form

$$\mathbf{0} = \hat{\mathbf{E}}\dot{\mathbf{z}}(t) + \hat{\mathbf{A}}\mathbf{z}(t) + \hat{\mathbf{B}}\mathbf{u}(t), \tag{4.10a}$$

$$\tilde{\mathbf{y}}(t) = \hat{\mathbf{C}}\mathbf{z}(t) + \hat{\mathbf{D}}\mathbf{u}(t), \tag{4.10b}$$

with state variable $\mathbf{z}(t) \in \mathbb{R}^r$, output $\tilde{\mathbf{y}}(t) \in \mathbb{R}^p$ and system matrices $\hat{\mathbf{E}}, \hat{\mathbf{A}} \in \mathbb{R}^{r\times r}$, $\hat{\mathbf{B}} \in \mathbb{R}^{r\times m}$, $\hat{\mathbf{C}} \in \mathbb{R}^{p\times r}$ and $\hat{\mathbf{D}} \in \mathbb{R}^{p\times m}$. The order r of this system is much smaller than the order n of the original system (4.4).

There are many ways to derive such a reduced order model and there are several possibilities for classifying these approaches. It is beyond the scope of this introductory chapter to give a detailed description of all the techniques – for this we refer to [3] and to the textbooks [1, 7, 51] and the papers cited therein.

We classify MOR approaches in projection and truncation based techniques. For each of the two classes we reflect two methods that can be seen as the basis for current developments. Note, that actually it is not possible to draw a sharp line. In fact all MOR techniques aim at keeping major information and removing the less important one. It is in how they measurure importance that the methods differ. In fact several current developments can be regarded as a hybridization of different techniques.

4.1.4 Projection Based MOR

The concept of all projection based MOR techniques is to approximate the high dimensional state space vector $\mathbf{x}(t) \in \mathbb{R}^n$ with the help of a vector $\mathbf{z}(t) \in \mathbb{R}^r$ of reduced dimension $r \ll n$, within the meaning of

$$\mathbf{x}(t) \approx \tilde{\mathbf{x}}(t) := \mathbf{V}\mathbf{z}(t) \quad \text{with } \mathbf{V} \in \mathbb{R}^{n\times r}.$$

Note that the first approximation may be interpreted as a wish. We will only aim for $\mathbf{y}(t) \approx \tilde{\mathbf{y}}(t) = \mathbf{C}\mathbf{V}\mathbf{z}(t) + \hat{\mathbf{D}}\mathbf{u}(t)$. The columns of the matrix $\mathbf{V}$ are a basis of a subspace $\tilde{\mathscr{M}} \subseteq \mathbb{R}^n$, i.e., the state space $\mathscr{M}$, the solution $\mathbf{x}(t)$ of the network equation (4.4a) resides in, is projected on $\tilde{\mathscr{M}}$. A reduced order model, representing the full problem (4.4) results from deriving a state space equation that determines the reduced state vector $\mathbf{z}(t)$ such that $\tilde{\mathbf{x}}(t)$ is a reasonable approximation to $\mathbf{x}(t)$.

If we insert $\tilde{\mathbf{x}}(t)$ on the right-hand side of the dynamic part of the input-output problem (4.4a), it will not vanish identically. Instead we get a residual:

$$\mathbf{r}(t) := \mathbf{E}\mathbf{V}\dot{\mathbf{z}}(t) + \mathbf{A}\mathbf{V}\mathbf{z}(t) + \mathbf{B}\mathbf{u}(t) \quad \in \mathbb{R}^n .$$

We can not demand $\mathbf{r}(t) \equiv \mathbf{0}$ in general as this would state an overdetermined system for $\mathbf{z}(t)$. Instead we apply the Petrov-Galerkin technique, i.e., we demand the residual to be orthogonal to some testspace $\mathscr{W}$. Assuming that the columns of a matrix $\mathbf{W} \in \mathbb{R}^{n\times r}$ span this testspace, the mathematical formulation of this orthogonality becomes

$$\mathbf{0} = \mathbf{W}^T\mathbf{r}(t) = \mathbf{W}^T \left(\mathbf{E}\mathbf{V}\dot{\mathbf{z}}(t) + \mathbf{A}\mathbf{V}\mathbf{z}(t) + \mathbf{B}\mathbf{u}(t)\right) \quad \in \mathbb{R}^r,$$

which states a differential equation for the reduced state $\mathbf{z}(t)$.

Defining

$$\begin{aligned} &\hat{\mathbf{E}} := \mathbf{W}^T\mathbf{E}\mathbf{V} \in \mathbb{R}^{r\times r}, \quad && \hat{\mathbf{A}} := \mathbf{W}^T\mathbf{A}\mathbf{V} \in \mathbb{R}^{r\times r}, \\ &\hat{\mathbf{B}} := \mathbf{W}^T\mathbf{B} \in \mathbb{R}^{r\times m}, && \hat{\mathbf{C}} := \mathbf{C}\mathbf{V} \in \mathbb{R}^{p\times r}, \\ &\hat{\mathbf{D}} := \mathbf{D} \in \mathbb{R}^{p\times m}, && \end{aligned} \tag{4.11}$$

we arrive at the reduced order model (4.10).

To relate $\mathbf{V}$ and $\mathbf{W}$ we demand biorthogonality of the spaces $\mathscr{V}$ and $\mathscr{W}$ spanned by the columns of the two matrices, respectively, i.e. $\mathbf{W}^T\mathbf{V} = \mathbf{I}_r$. With this, the reduced problem (4.10) is the projection of the full problem (4.4) onto $\mathscr{V}$ along $\mathscr{W}$. If an orthonormal $\mathbf{V}$ and $\mathbf{W} = \mathbf{V}$ is chosen, we speak of an orthogonal projection on the space $\mathscr{V}$ (and we come down to a Galerkin method).

Now, MOR projection methods are characterised by the way of how to construct the matrices $\mathbf{V}$ and $\mathbf{W}$ that define the projection. In the following we find a short introduction of *Krylov methods* and *POD* approaches. The former starts from the frequency domain representation, the latter from the time domain formulation of the input-output problem.

4.1.4.1 Krylov Method

Krylov-based methods to MOR are based on a series expansion of the transfer function $\mathbf{H}$. The idea is to construct a reduced order model such that the series expansions of the transfer function $\hat{\mathbf{H}}$ of the reduced model and the full problem's transfer function agree up to a certain index of summation.

In the following we will assume that the system under consideration does not have a direct feedthrough, i.e., (4.6a) is satisfied. Furthermore we restrict to SISO systems, i.e., single input single output systems. In this case we have $p = m = 1$, i.e., $\mathbf{B} = \mathbf{b}$ and $\mathbf{C} = \mathbf{c}^H$ where $\mathbf{b}, \mathbf{c} \in \mathbb{R}^n$, and the (scalar) transfer function becomes:

$$\mathbf{H}(s) = -\mathbf{c}^H \left(s\mathbf{E} + \mathbf{A}\right)^{-1} \mathbf{b} \quad \in \mathbb{C},$$

As $\{\mathbf{E}, \mathbf{A}\}$ is a regular matrix pencil we can find some frequency s_0 such that $s_0\mathbf{E} + \mathbf{A}$ is regular (for a good discussion on how to choose such "expansion points" s_0, see [17]). Then the transfer function can be reformulated as

$$\mathbf{H}(s) = \mathbf{l}\left(\mathbf{I}_n - (s - s_0)\mathbf{F}\right)^{-1} \mathbf{r}, \tag{4.12}$$

with $\mathbf{l} := -\mathbf{c}^H$, $\mathbf{r} := -(s_0\mathbf{E} + \mathbf{A})^{-1}\mathbf{b}$ and $\mathbf{F} := (s_0\mathbf{E} + \mathbf{A})^{-1}\mathbf{A}$.

In a neighbourhood of s_0 one can replace the matrix inverse in (4.12) by the corresponding Neumann series. Hence, a series expansion of the transfer function is

$$\mathbf{H}(s) = \sum_{k=0}^{\infty} m_k (s - s_0)^k \quad \text{with} \quad m_k := \mathbf{l}\mathbf{F}^k\mathbf{r} \quad \in \mathbb{C}. \tag{4.13}$$

The quantities m_k for $k = 0, 1, \ldots$ are called *moments* of the transfer function.

A different model, of lower dimension, can now be considered to be an approximation to the full problem, if the moments $\hat{m}_k$ of the new model's transfer function $\hat{\mathbf{H}}(s)$ agree with the moments m_k defined above, for $k = 1, \ldots, q$ for some $q \in \mathbb{N}$.

AWE [38], the *Asymptotic Waveform Evaluation*, was the first MOR method that was based on this idea. However, the explicit computation of the moments m_k, which is the key to AWE, is numerically unstable. This method can, thus, only be used for small numbers q of moments to be matched.

Expressions like $\mathbf{F}^k\,\mathbf{r}$ or $\mathbf{l}\,\mathbf{F}^k$ arise also in methods, like Krylov-subspace-methods, which are used for the iterative solution of large algebraic equations. Here the Lanczos- and the Arnoldi-method are algorithms that compute biorthogonal bases $\mathbf{W}, \mathbf{V}$ or a orthonormal basis $\mathbf{V}$ of the μth left and/or right Krylov subspaces

$$\mathscr{K}_l(\mathbf{F}^T, \mathbf{l}^T, \mu) := \operatorname{span}\left(\mathbf{l}^T, \mathbf{F}^T\,\mathbf{l}^T, \ldots, \left(\mathbf{F}^T\right)^{\mu-1}\,\mathbf{l}^T\right),$$
$$\mathscr{K}_r(\mathbf{F}, \mathbf{r}, \mu) := \operatorname{span}\left(\mathbf{r}, \mathbf{F}\,\mathbf{r}, \ldots, \mathbf{F}^{\mu-1}\,\mathbf{r}\right),$$

for $\mu \in \mathbb{N}$, respectively in a numerically robust way.

The Krylov subspaces, thus "contain" the moments m_k of the transfer function and it can be shown, e.g., [2, 12], that from applying Krylov-subspace methods, reduced order models can be created. These reduced order models, however, did not arise from a projection approach. In fact, the Lanczos- and the Arnold-algorithm produces besides the matrices $\mathbf{W}$ and/or $\mathbf{V}$ whose columns span the Krylov subspaces $\mathscr{K}_l$ and/or $\mathscr{K}_r$, respectively, a tridiagonal or an upper Hessenbergmatrix $\mathscr{T}$, respectively. This matrix is then used to postulate a dynamical system whose transfer function has the desired matching property.

Concerning the moment matching property there is a difference for reduced order models created from a Lanczos- and those created from an Arnoldi-based process.

For a fixed q, the Lanczos-process constructs the qth left and the qth right Krylov-subspace, hence biorthogonal matrices $\mathbf{W}, \mathbf{V} \in \mathbb{R}^{n \times q}$. A reduced order model of order q, arising from this procedure possesses a transfer function $\hat{\mathbf{H}}(s)$ whose first $2q$ moments coincide with the first $2q$ moments of the original problem's transfer function $\mathbf{H}(s)$, i.e. $\hat{m}_k = m_k$ for $k = 0, \ldots, 2q - 1$. Hence, the Lanczos MOR model yields a Padé approximation.

The Arnoldi method on the other hand is a one sided Krylov subspace method. For a fixed q only the qth right Krylov subspace is constructed. As a consequence, here only the first q moments of the original system's and the reduced system's transfer function match.

Owing to their robustness and low computational cost, Krylov subspace algorithms proved suitable for the reduction of large-scale systems, and gained considerable popularity, especially in electrical engineering. A number of Krylov-based MOR algorithms have been developed, including techniques based on the Lanczos method [9, 19] and the Arnoldi algorithm [36, 56]. Note that the moment matching, mentioned above, can only be valid locally, i.e., for a certain frequency range around the expansion point s_0. However, also Krylov MOR schemes based on a multipoint expansion in the frequency range have been constructed [21].

The main drawbacks of these methods are, in general, lack of provable error bounds for the extracted reduced models, and no guarantee for preserving stability and passivity. There are techniques to turn reduced systems to passive reduced systems. However, this introduced some post-processing of the model [18].

4.1.4.2 Passivity Preservation

Odabasioglu et al. [36] turned the Krylov based MOR schemes into a real projection method. In addition, the developed scheme, *PRIMA* (Passive Reduced-Order Interconnect Macromodeling Algorithm), is able to preserve passivity.

This MOR technique can be applied to electrical circuits that contain only passive linear resistors, capacitors and inductors and which accepts only currents as input at the terminals. One says that the RLC-circuit is in impedance form, i.e., the inputs $\mathbf{u}(t)$ are currents and the outputs $\mathbf{y}$ are voltages.

In this case, the system matrices $\mathbf{E}, \mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ have a special structure (cp. (4.3b)), namely:

$$\mathbf{E} = \begin{pmatrix} \mathbf{E}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{E}_2 \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ -\mathbf{A}_2^T & \mathbf{0} \end{pmatrix}, \quad \mathbf{B} = \mathbf{C}^T = \begin{pmatrix} \mathbf{B}_1 \\ \mathbf{0} \end{pmatrix}, \tag{4.14}$$

where $\mathbf{E}_1, \mathbf{A}_1 \in \mathbb{R}^{n_e \times n_e}$ and $\mathbf{E}_2 \in \mathbb{R}^{n_L \times n_L}$ and are symmetric non-negative definit matrices.

In *PRIMA*, first the Arnoldi method is applied to create the projection matrix $\mathbf{V}$. Then, choosing $\mathbf{W} = \mathbf{V}$, the system matrices are reduced according to (4.11). For several implementational details, covering Block-Arnoldi as well as deflation, see [55]. The reduced order model arising in this way can be shown to be passive [36]. The key to these findings is the above special structure of linear RLC-circuits in (4.14).

It is, however, not necessary, to use the Arnoldi method to construct the matrix $\mathbf{V}$. Furthermore, it is also possible to apply the technique to systems in admittance form, i.e., where the inputs are voltages and the outputs are currents. For more details we refer to [27] in this book.

4.1.4.3 Structure Preservation

As we have seen *PRIMA* takes advantage of the special block structure (4.14) of linear RLC circuits to create passive reduced order models. The structure, however, is not preserved during the reduction. This makes it hard to synthesise the model, i.e., realize the reduced model as an RLC circuit again.

Freund [12–16] developed a Krylov-based method where passivity, the structure and reciprocity are preserved. *SPRIM* (Structure-Preserving Reduced-Order Interconnect Macromodell) is similar to *PRIMA* as first the Arnoldi-method is run to create a matrix $\mathbf{V} \in \mathbb{R}^{n \times r}$. This, however, is not taken as the projection matrix directly. Instead, the matrix $\mathbf{V}$ is partitioned to

$$\mathbf{V} = \begin{pmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{pmatrix} \quad \text{with} \quad \mathbf{V}_1 \in \mathbb{R}^{n_e \times r}, \mathbf{V}_2 \in \mathbb{R}^{n_L \times r},$$

corresponding to the block structure of the system matrices $\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C}$.

Finally, after re-orthogonalization, the blocks $\mathbf{V}_1, \mathbf{V}_2$ are rearranged to the matrix

$$\hat{\mathbf{V}} = \begin{pmatrix} \mathbf{V}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_2 \end{pmatrix} \in \mathbb{R}^{n \times (2r)}, \tag{4.15}$$

which is then used to transform the system to a reduced order model, according to the transformations given in (4.11) (with $\mathbf{V} = \mathbf{W} = \hat{\mathbf{V}}$).

It can be shown, that the *SPRIM*-model preserves twice as many moments as the *PRIMA*, if the same Arnoldi-method is applied. Note, however, that the dimension also increases by a factor 2.

4.1.4.4 Multi-input Multi-output

For the general case, where p and m are larger than one, i.e., when we have multiple inputs and multiple outputs, the procedure carried out by the Krylov MOR methods is in principle the same. In this case however, Krylov subspaces for multiple starting vectors have to be computed and one has to take care, when a "breakdown" or a "near-breakdown" occurs, that is, when the basis vectors constructed for differing starting vectors, $\mathbf{r}_1$ and $\mathbf{r}_2$ become linearly dependent. In this case the progress for the Krylov subspace becoming linear dependent has to be stopped. The Krylov subspace methods arising from that considerations are called *Block Krylov methods*. For a detailed discussion we refer to the literature given above.

4.1.4.5 POD Method

While the Krylov approaches are based on the matrices, i.e., on the system itself, the method of Proper Orthogonal Decomposition (*POD*) is based on the trajectory $\mathbf{x}(t)$, i.e., the outcome of the system (4.4). One could also say that the Krylov methods are based on the frequency domain, whereas POD is based on the time domain formulation of the input output system to be modelled.

POD first collects data $\{\mathbf{x}_1, \dots, \mathbf{x}_K\}$. The datapoints are snapshots of the state space solution $\mathbf{x}(t)$ of the network equation (4.4a) at different timepoints t or for different input signals $\mathbf{u}(t)$. They are usually constructed by a numerical time simulation, but may also arise from measurements of a real physical system.

From analysing this data, a subspace is created such that the data points as a whole are approximated by corresponding points in the subspace in a optimal least-squares sense. The basis of this approach is also known as *Principal Component Analysis* and *Karhunen–Loève Theorem* from picture and data analysis.

The mathematical formulation of POD [39] is as follows: Given a set of K datapoints $X := \{\mathbf{x}_1, \dots, \mathbf{x}_K\}$ a subspace $\mathscr{S}_r \subset \mathbb{R}^n$ of dimension r is searched for that minimizes

$$\|X - \varrho_r X\|_2^2 := \frac{1}{K} \sum_{k=1}^{K} \|\mathbf{x}_k - \varrho_r \mathbf{x}_k\|_2^2, \tag{4.16}$$

where $\varrho_r : \mathbb{R}^n \to \mathscr{S}_r$ is the orthogonal projection onto $\mathscr{S}_r$.

We will not describe POD in full detail here, as in literature, e.g., [1, 39], this is well explained. However, the key to solving this minimization problem is the computation of the eigenvalues λ_i and eigenvectors φ_i (for $i = 1, \dots, n$) of the correlation matrix $\mathbf{X}\mathbf{X}^T$:

$$\mathbf{X}\mathbf{X}^T \boldsymbol{\varphi}_i = \lambda_i \boldsymbol{\varphi}_i,$$

where the eigenvalues and eigenvectors are sorted such that $\lambda_1 \geq \cdots \geq \lambda_n$. The matrix $\mathbf{X}$ is defined as $\mathbf{X} := (\mathbf{x}_1, \ldots, \mathbf{x}_K) \in \mathbb{R}^{n \times K}$ and is called *snapshot matrix*.

Intuitively the correlation matrix detects the principal directions in the data cloud that is made up of the snapshots $\mathbf{x}_1, \ldots, \mathbf{x}_K$. The eigenvectors and eigenvalues can be thought of as directions and radii of axes of an ellipsoid that incloses the cloud of data. Then, the smaller the radii of one axis is, the less information is lost if that direction is neglected.

The question arises, how many directions r should be kept and how many can be neglected. There is no a-priori error bound for the POD reduction (Rathinam and Petzold [43], though, perform a precise analysis of the POD accuracy). However, the eigenvalues are a measure for the relevance of the dimensions of the state space. Hence, it seems reasonable to choose the dimension r of the reduced order model in such a way, that the relative information content of the reduced model with respect to the full system is high. The measure for this content, used in the literature cited above is

$$\mathscr{I}(r) = \frac{\lambda_1 + \cdots \lambda_r}{\lambda_1 + \cdots \lambda_r + \lambda_{r+1} + \cdots \lambda_n}.$$

Clearly, a high relative information content means to have $\mathscr{I}(r) \approx 1$. Typically r is chosen such that this measure is around 0.99 or 0.995.

If the eigenvalues and eigenvectors are available and a dimension r has been chosen, the projection matrices $\mathbf{V}$ and $\mathbf{W}$ in (4.11) are taken as

$$\mathbf{V} := \mathbf{W} := (\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r) \in \mathbb{R}^{n \times r} .$$

leading to an orthogonal projection $\varrho_r = \mathbf{V}\mathbf{V}^T$ on the space $\mathscr{S}_r$ spanned by $\varphi_1, \ldots, \varphi_r$.

The procedure described so far relies on the eigenvalue decomposition of the $n \times n$ matrix $\mathbf{X}\mathbf{X}^T$. This direct approach is feasible only for problems of moderate size. For high dimensional problems, i.e., for dimensions $n \gg 1$, the eigenvalues and eigenvectors are derived form the Singular Value Decomposition (SVD) of the snapshot matrix $\mathbf{X} \in \mathbb{R}^{n \times K}$.

The SVD provides three matrices:

$$\begin{aligned}
\boldsymbol{\Phi} &= (\varphi_1, \cdots, \varphi_n) \in \mathbb{R}^{n \times n} \quad \text{orthogonal,} \\
\boldsymbol{\Psi} &= (\psi_1, \cdots, \psi_K) \in \mathbb{R}^{K \times K} \quad \text{orthogonal,} \\
\boldsymbol{\Sigma} &= \operatorname{diag}(\sigma_1, \ldots, \sigma_\nu) \in \mathbb{R}^{\nu \times \nu} \quad \text{with } \sigma_1 \geq \cdots \geq \sigma_\nu > \sigma_{\nu+1} = \ldots = \sigma_K = 0,
\end{aligned}$$

such that

$$\mathbf{X} = \boldsymbol{\Phi} \begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \boldsymbol{\Psi}^T, \tag{4.17}$$

where the columns of $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ are the left and right singular eigenvectors, respectively, and $\sigma_1, \dots, \sigma_\nu$ are the singular values of $\mathbf{X}$ (σ_ν being the smallest non-zero singular value; this also defines the index ν). It follows that $\varphi_1, \dots, \varphi_n$ are eigenvectors of the correlation matrix $\mathbf{X}\mathbf{X}^T$ with the n eigenvalues $\sigma_1^2, \dots, \sigma_\nu^2, 0, \dots, 0$.

4.1.5 Truncation Based MOR

The MOR approaches we reviewed so far rely on the approximation of the high-dimensional state space, the solution of (4.4) resides in, by an appropriate space of lower dimension. An equation for the correspondent $\mathbf{z}(t)$ of $\mathbf{x}(t)$ is derived by constructing a projection onto that lower-dimensional space.

Although the approaches we are about to describe in the following can also be considered as projection methods in a certain sense, we decided to present them separately. What makes them different, is that these techniques base on preserving key characteristics of the system rather than reproducing the solution. We will get aquainted with an ansatz based upon energy considerations and an approach meant to preserve poles and zeros of the transfer function.

4.1.5.1 Balanced Truncation

The technique of Balanced Truncation, introduced by Moore [35], is based on control theory, where one essentially investigates how a system can be steered and how its reaction can be observed. In this regard, the basic idea of *Balanced Truncation* is to first classify, which states $\mathbf{x}$ are hard to reach and which states $\mathbf{x}$ are hard to deduce from observing the output $\mathbf{y}$, then to reformulate the system such that the two sets of states coincide and finally truncate the system such that the reduced system does not attach importance to these problematic cases.

The system (4.4) can be driven to the state $\bar{\mathbf{x}}$ in time T if an input $\bar{\mathbf{u}}(t)$, with $t \in [0, T]$ can be defined such that the solution at time T, i.e., $\mathbf{x}(T)$ takes the value $\bar{\mathbf{x}}$ where $\mathbf{x}(0) = \mathbf{0}$. We perceive the L_2-norm $\| \cdot \|_2$, with $\|\bar{\mathbf{u}}\|_2^2 = \int_0^T \bar{\mathbf{u}}(t)^T \bar{\mathbf{u}}(t)\, dt$ as energy of the input signal. If the system is in state $\tilde{\mathbf{x}}$ at time $t = 0$ and no input is applied at its ports we can observe the output $\tilde{\mathbf{y}}(t)$ for $t \in [0, T]$ and the energy $\|\tilde{\mathbf{y}}\|_2$ emitted at the system's output ports.

We consider a state as hard to reach if the minimal energy needed to steer the system to that state is large. Similarly, a state whose output energy is small leaves a weak mark and is therefore considered to be hard to be observed.

The minimal input energy needed and the maximal energy emitted can be calculated via the finite and the infinite *controllability Gramian*

$$\mathscr{P}(T) = \int_0^T e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^T e^{\mathbf{A}^T t} dt \quad \text{and} \quad \mathscr{P} = \int_0^\infty e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^T e^{\mathbf{A}^T t} dt \tag{4.18a}$$

and the finite and infinite *observability Gramian*

$$\mathscr{Q}(T) = \int_0^T e^{\mathbf{A}^T t}\mathbf{C}^T\mathbf{C}e^{\mathbf{A}t}dt \quad \text{and} \quad \mathscr{Q} = \int_0^\infty e^{\mathbf{A}^T t}\mathbf{C}^T\mathbf{C}e^{\mathbf{A}t}dt, \tag{4.18b}$$

respectively. Note that the system (4.4) is assumed to be stable. Furthermore, the above definition is valid for the case $\mathbf{E} = \mathbf{I}_{n\times n}$. The latter does not mean a limitation of the method of Balanced Truncation to standard state space systems. In fact, these considerations can be applied to descriptor systems as well, e.g., [54].

With the above definitions one can prove that the minimal energy needed, i.e., the energy connected to the most economical input $\bar{\mathbf{u}}$, to reach the state $\bar{\mathbf{x}}$ holds

$$\|\bar{\mathbf{u}}\|_2^2 = \bar{\mathbf{x}}^T\mathscr{P}^{-1}\bar{\mathbf{x}}.$$

Similarly, the energy emitted due to the state $\tilde{\mathbf{x}}$ holds

$$\|\tilde{\mathbf{y}}\|_2^2 = \tilde{\mathbf{x}}\mathscr{Q}\tilde{\mathbf{x}}.$$

The Gramians are positive definite. Applying a diagonalization of the controllability Gramian, it is easy to see that states that have a large component in the direction of eigenvectors corresponding to small eigenvalues of $\mathscr{P}$ are hard to reach. In the same way it is easy to see that states pointing in the direction of eigenvectors to small eigenvalues of the observability Gramian $\mathscr{Q}$ are hard to observe.

The basic idea of the Balanced Truncation MOR approach is to neglect states that are both hard to reach and hard to observe. This marks the truncation part. However, to reach this synchrony of a state being both hard to reach and hard to observe, the basis of the state space has to be transformed. This marks the balancing part. Generally, a basis transformation introduces new coordinates $\tilde{\mathbf{x}}$ such that $\mathbf{x} = \mathbf{T}^{-1}\tilde{\mathbf{x}}$ where $\mathbf{T}$ is the matrix representation of the basis transformation. Here the Gramians transform equivalently to

$$\tilde{\mathscr{P}} = \mathbf{T}\mathscr{P}\mathbf{T}^T \quad \text{and} \quad \tilde{\mathscr{Q}} = \mathbf{T}^{-1}\mathscr{Q}\mathbf{T}^{-T}.$$

The transformation $\mathbf{T}$ is called *balancing transformation* and the system arising from applying the transformation to the system (4.4) is called *balanced* if the transformed Gramians satisfy

$$\tilde{\mathscr{P}} = \tilde{\mathscr{Q}} = \operatorname{diag}(\sigma_1, \dots, \sigma_n). \tag{4.19}$$

The values $\sigma_1, \dots, \sigma_n$ are called *Hankel Singular Values*. They are the positive square roots of the eigenvalues of the product of the Gramians:

$$\sigma_l = \sqrt{\lambda_k(\mathscr{P}\cdot\mathscr{Q})}, \quad l = 1, \dots, n.$$

Now we assume that the eigenvalues are sorted in descending order, i.e., $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$. We introduce the cluster

$$\left(\begin{array}{ccc|ccc} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_r & & & \\ \hline & & & \sigma_{r+1} & & \\ & & & & \ddots & \\ & & & & & \sigma_n \end{array}\right) = \begin{pmatrix} \Sigma_1 & \\ & \Sigma_2 \end{pmatrix},$$

and adopt this to the tranformed input-output system[11]

$$\begin{aligned} \mathbf{0} &= \begin{pmatrix} \dot{\tilde{\mathbf{x}}}_1(t) \\ \dot{\tilde{\mathbf{x}}}_2(t) \end{pmatrix} + \begin{pmatrix} \tilde{\mathbf{A}}_{11} & \tilde{\mathbf{A}}_{12} \\ \tilde{\mathbf{A}}_{21} & \tilde{\mathbf{A}}_{22} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_1(t) \\ \tilde{\mathbf{x}}_2(t) \end{pmatrix} + \begin{pmatrix} \tilde{\mathbf{B}}_1 \\ \tilde{\mathbf{B}}_2 \end{pmatrix} \mathbf{u}(t), \\ \mathbf{y}(t) &= (\mathbf{C}_1, \mathbf{C}_2) \begin{pmatrix} \tilde{\mathbf{x}}_1(t) \\ \tilde{\mathbf{x}}_2(t) \end{pmatrix}, \end{aligned} \tag{4.20}$$

such that $\tilde{\mathbf{x}}_1 \in \mathbb{R}^r$ and $\tilde{\mathbf{x}}_2 \in \mathbb{R}^{n-r}$.

Finally we separate the cluster and derive the reduced order model

$$\mathbf{0} = \dot{\hat{\mathbf{x}}}_{11}(t) + \tilde{\mathbf{A}}_1 \hat{\mathbf{x}}_1(t) + \tilde{\mathbf{B}}_1 \mathbf{u}(t), \tag{4.21a}$$

$$\tilde{\mathbf{y}}_1(t) = \tilde{\mathbf{C}}_1 \hat{\mathbf{x}}_1(t) \tag{4.21b}$$

of dimension $r \ll n$, by skipping the part corresponding to the small eigenvalues $\sigma_{r+1}, \ldots, \sigma_n$ of both Gramians.

Important Properties

Balanced Truncation is an appealing MOR technique because it automatically preserves stability.

Furthermore, and even more attractive is that this MOR approach provides a *computable error bound*: Let $\sigma_{r+1}, \ldots, \sigma_k$ be the different eigenvalues that are truncated. Then, for the transfer function $\mathbf{H}_1$ corresponding to (4.21), it holds

$$\|\mathbf{H} - \mathbf{H}_1\|_{\mathbb{H}_\infty} \leq 2\left(\sigma_{r+1} + \cdots + \sigma_k\right), \tag{4.22}$$

where the $\mathbb{H}_\infty$ norm is defined as $\|\mathbf{H}\|_{\mathbb{H}_\infty} := \sup_{\omega \in \mathbb{R}} \|\mathbf{H}(i\omega)\|_2$ where $\|\cdot\|_2$ is the matrix spectral norm.

[11] To simplify matters we have chosen $\mathbf{E} = \mathbf{I}_{n \times n} = \mathrm{diag}(1, \ldots, 1) \in \mathbb{R}^{n \times n}$ and $\mathbf{D} = \mathbf{0}$.

Computation

Applying the method of Balanced Truncation as presented above makes it necessary to compute the Gramians and the simultaneous diagonalization of the Gramians.

The infinite Gramians $\mathscr{P}$ and $\mathscr{Q}$ are defined by infinity integrals. However, it is not hard to show that they arise from solving the *Lyapunov equations*:

$$\begin{aligned} \mathbf{A}\mathscr{P} + \mathscr{P}\mathbf{A}^T + \mathbf{B}\mathbf{B}^T &= \mathbf{0} \\ \mathbf{A}^T\mathscr{Q} + \mathscr{Q}\mathbf{A} + \mathbf{C}^T\mathbf{C} &= \mathbf{0} \end{aligned} \tag{4.23}$$

Having solved the Lyapunov equations, one way to determine the balancing transformation is described by the *square root algorithm* (see e.g. [1]). The basic steps in this approach are the computation of the Cholesky factorisations of the Gramians $\mathscr{P} = \mathbf{S}^T\mathbf{S}$ and $\mathscr{Q} = \mathbf{R}^T\mathbf{R}$ and the singular value decomposition of the product $\mathbf{S}\mathbf{R}^T$.

In the past Balanced Bruncation was not favored because the computation of the solution of the high dimensional matrix equations (4.23) and the balancing was very cumbersome and costly. In recent years however, progress was made in the development of techniques to overcome these difficulties. Techniques that can be applied to realize the Balanced Truncation include the ADI method [30], the sign function method [4] or other techniques, e.g. [49]. For a collection of techniques we also refer to [5].

Poor Man's TBR

Another method that should be mentioned is *Poor Man's TBR,*[12] introduced by Phillips and Silveira [37]. Balanced Truncation relies on the Gramians. The methods we mentioned so far compute these Gramians based on the Lyapunov equations (4.23).

The idea of Poor Man's TBR (PMTBR) however, is to compute the Gramians from their definition (4.18). If the system to be reduced is symmetric, i.e. $\mathbf{A} = \mathbf{A}^T$ and $\mathbf{C} = \mathbf{B}^T$, $\mathscr{P}$ and $\mathscr{Q}$ coincide. The (controllability and observability) Gramian is then defined as

$$\mathscr{P} = \int_0^\infty e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^T e^{\mathbf{A}^T t}\,dt.$$

As the Laplace transform of $e^{\mathbf{A}t}$ is $(s\mathbf{I}-\mathbf{A})^{-1}$, we can apply Parseval's lemma, which says that a signal's energy in the time domain is equal to its energy in the frequency

[12] TBR = Truncated Balanced Realization

domain and transfer the time domain integral to the frequency domain:

$$\mathscr{P} = \int_{-\infty}^{\infty} (i\omega\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{B}^T (i\omega\mathbf{I} - \mathbf{A})^{-H} d\omega.$$

PMTBR now starts with applying a numerical quadrature scheme: With nodes ω_k, weights w_k and defining $z_k = (i\omega_k\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$ an approximation $\hat{\mathscr{P}}$ to the gramian $\mathscr{P}$ can be computed as:

$$\mathscr{P} \approx \hat{\mathscr{P}} = \sum_k w_k\ z_k\ z_k^H = ZW \cdot (ZW)^H,$$

where $Z = (z_1, z_2, \ldots)$ and $W = \mathrm{diag}(\sqrt{w_1}, \sqrt{w_2}, \ldots)$.

For further details on the order reduction we refer to the original paper mentioned above.

4.1.5.2 Modal Truncation

Engineers usually investigate the transfer behavior of an input-output system by inspecting its frequency response $\mathbf{H}(i\omega) =: \mathbf{G}(\omega)$ for frequencies $\omega \in \mathbb{R}^+$. The Bode plot, i.e. the combination of the Bode magnitude and phase plot, expressing how much a signal component with a specific frequency is amplified or attenuated and which phase shift can be observed from in- to output, respectively, is a graphical representation of the frequency response.

One is especially interested in the peaks and sinks of the Bode magnitude plot, which are caused by the *poles* and *zeros* of the transfer function $\mathbf{H}$. The *Modal Truncation* [45] is aimed at constructing a reduced order model (4.10) such that peaks and sinks of the reduced order model's frequency response $\hat{\mathbf{G}}(\omega) = \hat{\mathbf{H}}(i\omega)$ match with the one of the full dynamical problem (4.4).

Applying Cramer's rule it is obvious that the transfer function is a rational function:

$$\mathbf{H}(s) = \frac{p_{n-1}(s)}{q_n(s)},$$

with polynomials p_{n-1} and q_n of degree $n-1$ and n respectively. The zeros of the numerator are the zeros of the transfer function and the zeros of the denominator are its poles.

The generalized eigenvalues of the matrix pencil $\{\mathbf{E}, \mathbf{A}\}$, or the eigenvalues of $\mathbf{A}$, if we assume $\mathbf{E} = \mathbf{I}_{n\times n}$, are the key to the poles of the transfer function. For a more detailed discussion we refer to [28]. To illustrate this relation we restrict to the latter case and consider a SISO system without direct feedtrough, i.e., $\mathbf{D} = \mathbf{0}$.

The eigentriples $(\lambda_i, \mathbf{v}_i, \mathbf{w}_i)$ for $i = 1, \ldots, n$ of $\mathbf{A}$ consist of the ith eigenvalue $\lambda_i \in \mathbb{C}$ and the ith right and left eigenvalue $\mathbf{v}_i, \mathbf{w}_i \in \mathbb{C}^n$, respectively, that satisfy

$$\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i \quad \text{and} \quad \mathbf{w}_i^H \mathbf{A} = \lambda_i \mathbf{w}_i^H.$$

From assuming that $\mathbf{A}$ is diagonalizable it can be derived that

$$\mathbf{L}^H \mathbf{A} \mathbf{R} = \boldsymbol{\Lambda},$$

where $\boldsymbol{\Lambda} = (\lambda_1, \ldots, \lambda_n)$, $\mathbf{R} = (\mathbf{v}_1, \ldots, \mathbf{v}_n)$ and $\mathbf{L} = (\mathbf{w}_1, \ldots, \mathbf{w}_n) \in \mathbb{C}^{n \times n}$, where the left and right eigenvectors are scaled such that $\mathbf{L}^H \mathbf{R} = \mathbf{I}_{n \times n}$.

We apply a change of coordinates $\mathbf{x} = \mathbf{R}\tilde{\mathbf{x}}$ and multiply the input to state mapping (4.4a) with $\mathbf{L}^H$ which is a projection on the space spanned by the columns of $\mathbf{R}$ along the space spanned by the columns of $\mathbf{L}$. This transforms the input-output system (4.4) to

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\tilde{\mathbf{x}} &= \boldsymbol{\Lambda}\tilde{\mathbf{x}} + \mathbf{L}^H \mathbf{b}\mathbf{u}, \\ \mathbf{y} &= \mathbf{c}^H \mathbf{R}\tilde{\mathbf{x}}. \end{aligned} \tag{4.24}$$

The transformed system is equivalent to the original system (4.4), the (scalar) transfer function can be represented as

$$\mathbf{H}(s) = \sum_{i=1}^{n} \frac{r_i}{s - \lambda_i} \quad \text{with} \quad r_i = \left(\mathbf{c}^H \mathbf{v}_i\right)\left(\mathbf{w}_i^H \mathbf{b}\right) \in \mathbb{C} \quad \text{for } i = 1, \ldots, n. \tag{4.25}$$

This form of displaying the transfer function is known as *Pole-Residue Representation*, where the quantities $r_i \in \mathbb{C}$ are called *residues* and where we can see that the eigenvalues of the matrix $\mathbf{A}$ are the poles of $\mathbf{H}(s)$.

The idea of modal truncation is to replace the full order problem with a reduced order model of say order $r < n$ whose transfer function has a pole-residue representation that is a truncation of the corresponding full model's representation (4.25), i.e.

$$\hat{\mathbf{H}}(s) = \sum_{i=1}^{r} \frac{r_i}{s - \lambda_i}, \tag{4.26}$$

where r_i and λ_i for $i = 1, \ldots, r$ are the same as in (4.25). The corresponding state-space representation (4.10) evolves from carrying out the matrix projections defined in (4.11) where $\mathbf{V}, \mathbf{W} \in \mathbb{C}^{n \times r}$ comprises r right and left eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_r$ and $\mathbf{w}_1, \ldots, \mathbf{w}_r$, respectively. As no new poles arise by constructing the reduced order model in this way, the stability property is inherited from the full order problem.

Immediately the question arises, which pairs (λ_i, r_i) of poles and residues and how many should be taken into account.

Rommes [45] and Martins et al. [31] sort these pairs according to decreasing *dominance* of the pole. Their measure for dominance of a pole is the magnitude of the relation

$$\frac{|r_i|}{|\mathrm{re}(\lambda_i)|}.$$

Hence, modal truncation takes into account the first r poles/residues according to this ordering scheme. The answer to the second part of the question, i.e., how many poles/residues to keep, arises from the error bound [20]

$$\|\mathbf{H} - \hat{\mathbf{H}}\|_{\mathbb{H}_\infty} \leq \sum_{j=r+1}^{n} \frac{|r_j|}{|\mathrm{re}(\lambda_j)|}, \tag{4.27}$$

and hence from the deviation one is willing to tolerate.

The computation of the error bound (4.27) necessitates a full eigenvalue decomposition. This is only feasible for moderate orders $n \leq 2{,}000$. For large scale systems methods using only a partial eigenvalue decomposition can be applied. Here the *Subspace Accelerated Dominant Pole Algorithm (SADPA)*, introduced by Rommes and Martins [46] produces a series of dominant poles. The main principle of SADPA is to search for the zeros of $\frac{1}{\mathbf{H}(s)}$ using a Newton-iteration. As the Newton-iteration can only find one zero sufficiently close to a starting value at a time, the iteration procedure has to be applied several times. In order to find less dominant poles at each time, the system the dominant pole algorithm is applied to is adjusted each time one dominant pole has been found. This adjustment is referred to as *subspace acceleration*.

Again, for further details we refer to the papers cited above.

4.1.6 Other Approaches

We shortly address some other approaches. In [26, 40, 41], port-Hamiltonian systems are considered to guarantee structure preserving reduced models. In [8, 10, 11], vector fitting techniques are used to obtain passivity preserving reduced models. In [25, 32, 47], one matches additional moments of Laurent expansions involving terms with $1/s$. These are applied to obtain passive reduced models for RLC circuits.

4.1.7 Examples

In this part we will introduce linear circuits and reduce them with techniques which have already been discussed. We give results for the methods PRIMA [36], SPRIM [12–16], and PMTBR [37].

In simulation a Bode magnitude plot of the transfer function shows the magnitude of $\mathbf{H}(i\omega)$, in decibel, for a number of frequencies ω in the frequency domain of interest. If the transfer function of the original system can be evaluated at enough points $s = i\omega$ to produce an accurate Bode plot, the original frequency response can be compared with the frequency response of the reduced model. In our examples, $\mathbf{H}$ is a scalar.

4.1.7.1 Example 1

We choose an RLC ladder network [33] shown in Fig. 4.1. We set all the capacitances and inductances to the same value 1 while $R_1 = \frac{1}{2}$ and $R_2 = \frac{1}{5}$, see [34, 53]. We arrange 201 nodes which gives us the order 401 for the mathematical model of the circuit.

If we write the standard MNA formulation the linear dynamical system is derived. The system matrices are as follows (for $K = 3$, for example):

$$\mathbf{E} = \mathbf{I}, \quad \mathbf{A} = \begin{bmatrix} -2 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & -5 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 5 \\ 0 \\ 0 \end{bmatrix},$$
$$\mathbf{C} = \begin{bmatrix} 0 \; 0 \; -5 \; 0 \; 0 \end{bmatrix}, \quad D = 5. \tag{4.28}$$

In the state variable $\mathbf{x}$, x_k is the voltage across capacitance C_k ($k = 1, \ldots, K$), or the current through inductor L_{k-K} ($k = K + 1, \ldots, 2K - 1$). In general the number of nodes K is odd. The voltage u and the current y are input and output, respectively. Note that when the number of nodes is K the order of the system becomes $n = 2K - 1$. In this test case we have an ODE instead of a DAE as $\mathbf{E} = \mathbf{I}$, see (4.28). The original transfer function is shown in Fig. 4.2. The plot already illustrates how difficult it is to reduce this transfer function as many oscillations appear.

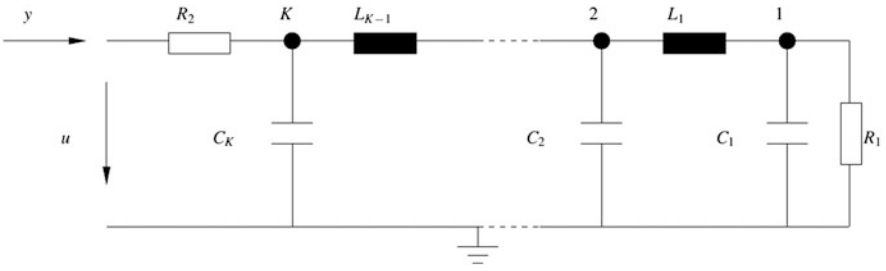

Fig. 4.1 RLC Circuit of order $n = 2K - 1$, Example 1

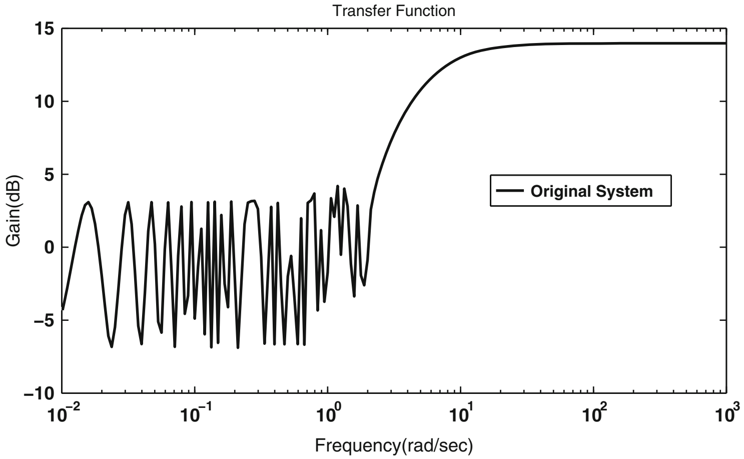

Fig. 4.2 Original transfer function for the first example of Fig. 4.1, order $n = 401$. The frequency domain parameter ω varies between 10^{-2} to 10^3

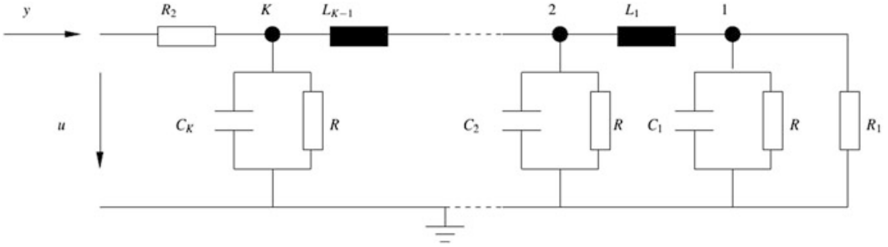

Fig. 4.3 RLC Circuit of order $n = 2K - 1$, Example 2

4.1.7.2 Example 2

Next, we use another RLC ladder network, given in Fig. 4.3 [33, 48], for the second example. The major difference to the previous example is that we introduced a resistor (all of equal value) in parallel to the capacitors at each node connected to the ground. We set all the capacitances and inductances to the same value 1 while $R_1 = \frac{1}{2}$, $R_2 = \frac{1}{5}$ and $R = 1$. We choose 201 nodes which results in a system having order 401 for the mathematical model of the circuit. Like the previous example we again derive a system of ODEs. The original transfer function of the second example is shown in Fig. 4.4.

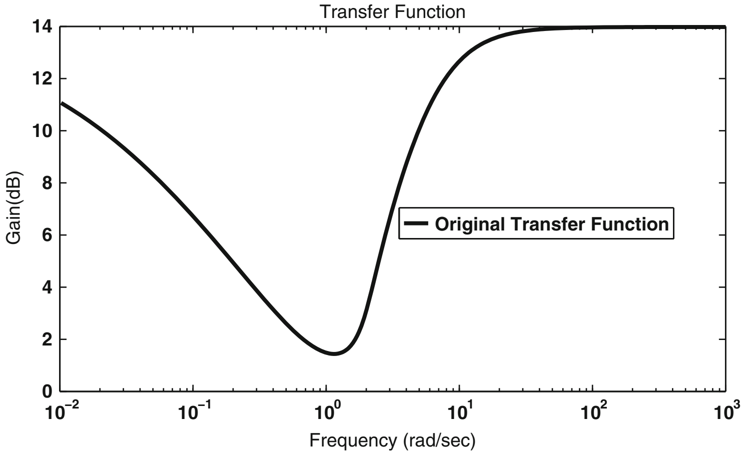

Fig. 4.4 Original transfer function for the second example of Fig. 4.3, order $n = 401$. The frequency domain parameter ω varies between 10^{-2} to 10^{3}

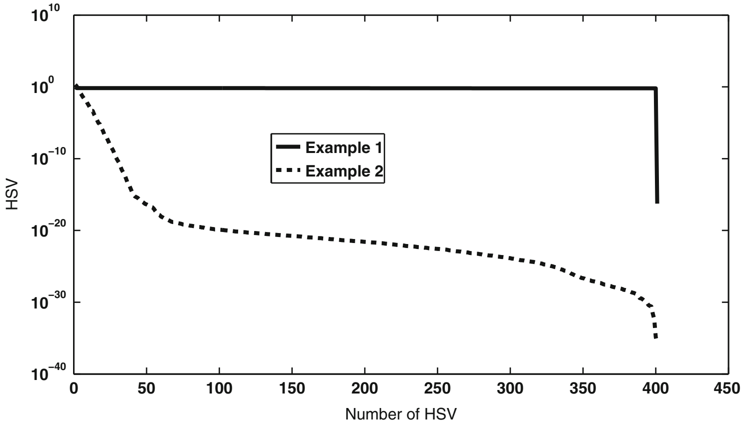

Fig. 4.5 Hankel Singular Values for Example 1 and 2, (semi-logarithmic scale)

4.1.7.3 MOR by PRIMA, SPRIM and PMTBR

The main reason for choosing these two examples is the behavior of the Hankel singular values, see Fig. 4.5. The Hankel singular values for the first example do not show any significant decay, while in the second example we observe a rapid decay in the values. The results are taken from [33].

The Figs. 4.6 and 4.7 show the absolute error between the transfer function of the full system and the transfer function of several reduced systems. The model is reduced by three linear techniques (PRIMA, SPRIM and PMTBR) for both examples.

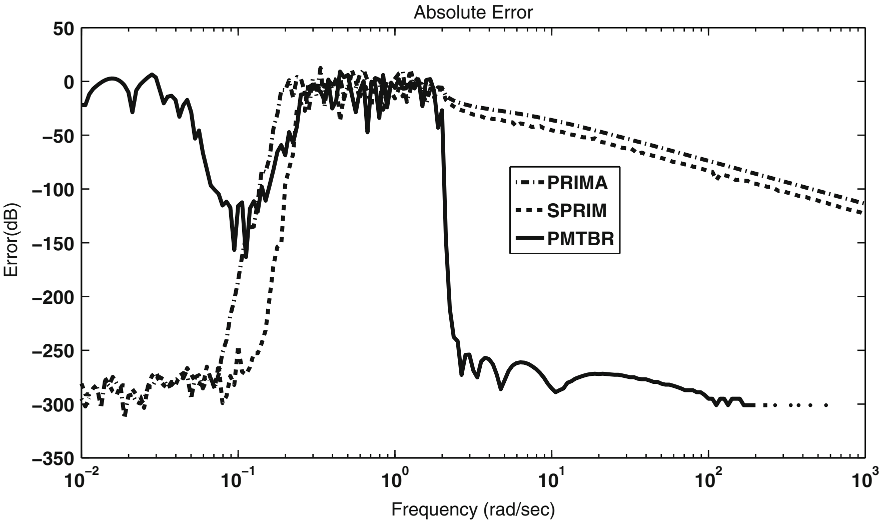

Fig. 4.6 Error plot, the frequency domain parameter ω varies between 10^{-2} to 10^{3}, Example 1

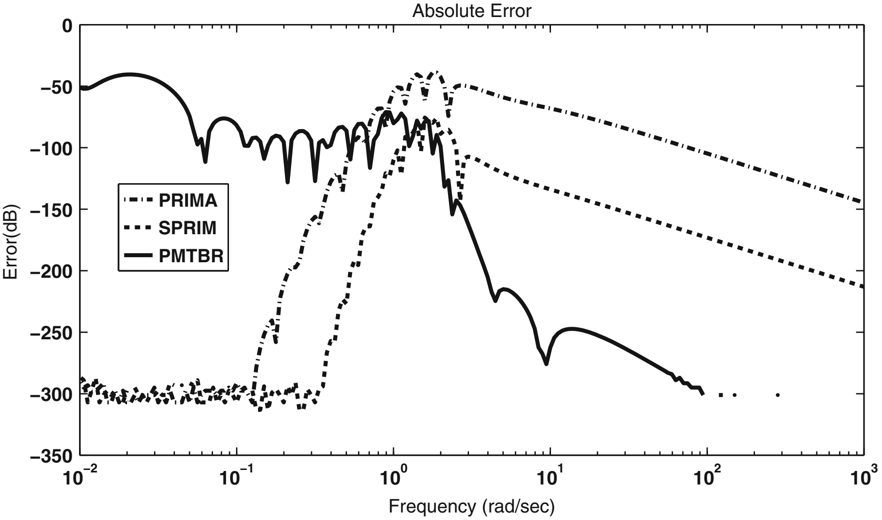

Fig. 4.7 Error plot, the frequency domain parameter ω varies between 10^{-2} to 10^{3}, Example 2

In the Example 1 we reduced the system from order $n = 401$ (number of nodes is $K = 201$) to order 34, which means that we reduced the system (in all three methods) by a factor of 10. The order of the reduced model is relatively large in this case as the dynamical system is somehow stubborn for any reductions, see Fig. 4.5. The price we will pay for a smaller system is too high as we loose a lot of information during the reduction and the error is becoming relatively large. As we expected, PRIMA and SPRIM in Fig. 4.6 produced reliable results close to the expansion point, in this case $s = 0$, but the error is immediately increasing for the rest of the oscillation part, see Fig. 4.2, and then smoothly decreases for higher frequencies. In the first example the PMTBR method matches a bit worse for the low frequencies as the error decreases just for a short interval and immediately starts to increase again. But PMTBR also cannot cover the oscillation part of the transfer function although it resolves the higher frequencies well. The order in PMTBR results from a prescribed tolerance.

For the second example the SPRIM and PRIMA produced a nice match around the expansion point, $s = 0$, like the first example, but for a larger interval, see Fig. 4.7. The peaks of error for both PRIMA and SPRIM are around -50 and -80 dB, respectively, which are much lower than in Example 1 where the peaks are around 0 dB for both PRIMA and SPRIM. We allowed PMTBR to reduce the system by a factor of 20 in this case although we keep the order of the reduced system the same as for the first example for the PRIMA and SPRIM. Despite the lower dimension for the reduced system PMTBR produced much better results for this test case compared to the first example as the error starts from -50 dB and smoothly decreases for low frequencies and suddenly falls to -300 dB for larger frequencies.

As we expected, the SPRIM produces a better approximation than PRIMA, especially for the second example, since it matches twice as much moments. Although both methods have a good agreement around the expansion point $s = 0$, the error increases as we are far from the expansion point. Since the Hankel singular values for the first example do not decay, the PMTBR method cannot produce an accurate model for low frequencies in that case. In the second example where the Hankel singular values rapidly decay PMTBR produced a more reliable result with a better match. This shows that we cannot stick to one method for reduction in general and the method should be chosen depending on the circuit's behavior.

4.1.8 Summary

In industrial applications of different disciplines, model order reduction is gaining more and more interest. As there is not the one and only type of model to describe all kinds of dynamics of different physical problems there is not and will never be the one and only MOR technique that fits best to all problems. Hence, research on MOR techniques is an ongoing process.

In the following contributions in this chapter you will find different approaches to different questions, aiming to attack different facets of reduced order models. This introductory contribution was ment to give an overview of the basic ideas and the motivation of some MOR techniques that are applied and refined throughout this chapter.

4.2 Eigenvalue Methods and Model Order Reduction

Physical structures and processes are modeled by dynamical systems in a wide range of application areas.[13] The increasing demand for complex components and large structures, together with an increasing demand for detail and accuracy, makes the models larger and more complicated. To be able to simulate these large-scale systems, there is need for reduced-order models of much smaller size, that approximate the behavior of the original model and preserve the important characteristics.

In order to preserve the important characteristics, one usually first has to know *what* are the important characteristics. For linear dynamical systems, two important characteristics are the dominant dynamics and stability. The dominant dynamics are determined by the dominant modes of the system, while stability of the system is determined by the location of the eigenvalues. Hence, both characteristics can be computed by solving eigenvalue problems: the dominant dynamics can be found by computing the dominant eigenvalues (poles) and corresponding eigenvectors, while stability can be assessed by determining whether the system has no eigenvalues in the right half-plane (the system is stable if there are no eigenvalues with real part greater than zero).

A large-scale dynamical system can have a large number of modes. Like a general square matrix can be approximated by its largest eigenvalues, i.e. by projecting it onto the space spanned by the eigenvectors corresponding to the largest eigenvalues, a dynamical system can be approximated by its dominant modes: a reduced order model, called the *modal equivalent*, can be obtained by projecting the state space on the subspace spanned by the dominant eigenvectors. This technique, *modal approximation* or *modal model reduction*, has been successfully applied to scalar and multivariable transfer functions of large-scale power systems, with applications such as stability analysis and controller design, see [81, 82].

The dominant eigenvectors, and the corresponding dominant poles of the system transfer function, are specific eigenvectors and eigenvalues of the state matrix. Because the systems are very large in practice, it is not feasible to compute all eigenvectors and to select the dominant ones.

Section 4.2 is concerned with the efficient computation of the dominant poles and eigenvectors specifically, and their use in model order reduction. The section is organized as follows. In Sect. 4.2.1 the concept of dominant poles and modal approximation is explained in more detail. Dominant poles can be computed with specialized eigensolution methods, as is described in Sect. 4.2.2. Some generalizations of the presented algorithms are shown in Sect. 4.2.3. Ideas on how to improve Krylov based MOR methods by using dominant poles are discussed in Sect. 4.2.4. Numerical examples are presented in Sect. 4.2.5. Section 4.2.6 concludes.

[13] Section 4.2 has been written by: Joost Rommes and Nelson Martins.

For general introductions to model order reduction we refer to the previous Sect. 4.1 and to [58, 60, 61, 88]; for eigenvalue problems, see [87, 93]. More detailed publications on the contents of this section are [80–85]. The pseudocode algorithms presented in this section are written using Matlab-like [92] notation.

4.2.1 Transfer Functions, Dominant Poles and Modal Equivalents

In Sect. 4.2, the dynamical systems $(E, A, \mathbf{b}, \mathbf{c}, d)$ are of the form

$$\begin{cases} E\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + \mathbf{b}u(t) \\ y(t) \quad = \mathbf{c}^*\mathbf{x}(t) + du(t), \end{cases} \tag{4.29}$$

where $A, E \in \mathbb{R}^{n\times n}$, E may be singular, $\mathbf{b}, \mathbf{c}, \mathbf{x}(t) \in \mathbb{R}^n$, $u(t), y(t), d \in \mathbb{R}$. The vectors $\mathbf{b}$ and $\mathbf{c}$ are called the input, and output map, respectively. The transfer function $H : \mathbb{C} \to \mathbb{C}$ of (4.29) is defined as

$$H(s) = \mathbf{c}^*(sE - A)^{-1}\mathbf{b} + d. \tag{4.30}$$

The poles of the transfer function in (4.30) are a subset of the eigenvalues $\lambda_i \in \mathbb{C}$ of the matrix pencil (A, E). An eigentriplet $(\lambda_i, \mathbf{x}_i, \mathbf{y}_i)$ is composed of an eigenvalue λ_i of (A, E) and corresponding right and left eigenvectors $\mathbf{x}_i, \mathbf{y}_i \in \mathbb{C}^n$:

$$\begin{aligned} A\mathbf{x}_i &= \lambda_i E\mathbf{x}_i, \qquad \mathbf{x}_i \neq 0, \\ \mathbf{y}_i^* A &= \lambda_i \mathbf{y}_i^* E, \qquad \mathbf{y}_i \neq 0, \qquad (i = 1, \dots, n). \end{aligned}$$

The actual occurring poles in (4.30) are identified by the components of the eigenvectors in in $\mathbf{b}$ and $\mathbf{c}$. Assuming that the pencil is nondefective, the right and left eigenvectors corresponding to the same finite eigenvalue can be scaled so that $\mathbf{y}_i^* E\mathbf{x}_i = 1$. Furthermore, it is well known that left and right eigenvectors corresponding to distinct eigenvalues are E-orthogonal: $\mathbf{y}_i^* E\mathbf{x}_j = 0$ for $i \neq j$. The transfer function $H(s)$ can be expressed as a sum of residues $R_i \in \mathbb{C}$ over the $\tilde{n} \leq n$ finite first order poles [68]:

$$H(s) = \sum_{i=1}^{\tilde{n}} \frac{R_i}{s - \lambda_i} + R_\infty + d, \tag{4.31}$$

where the residues R_i are

$$R_i = (\mathbf{c}^*\mathbf{x}_i)(\mathbf{y}_i^*\mathbf{b}),$$

and R_∞ is the constant contribution of the poles at infinity (often zero).

Although there are different indices of modal dominance [57, 64, 94], the following will be used in this chapter.

Definition 4.1 A pole λ_i of $H(s)$ with corresponding right and left eigenvectors $\mathbf{x}_i$ and $\mathbf{y}_i$ ($\mathbf{y}_i^* E\mathbf{x}_i = 1$) is called the dominant pole if $|R_i|/|\mathrm{Re}(\lambda_i)| > |R_j|/|\mathrm{Re}(\lambda_j)|$, for all $j \neq i$.

More generally, a pole λ_i is called dominant if $|R_i|/|\mathrm{Re}(\lambda_i)|$ is not small compared to $|R_j|/|\mathrm{Re}(\lambda_j)|$, for all $j \neq i$. A dominant pole is well observable and controllable in the transfer function. This can also be seen in the corresponding Bode-plot, which is a plot of the magnitude $|H(i\omega)|$ against $\omega \in \mathbb{R}$: peaks occur at frequencies ω close to the imaginary parts of the dominant poles of $H(s)$. In practise one also plots the corresponding phase of $H(i\omega)$. An approximation of $H(s)$ that consists of $k < n$ terms with $|R_j|/|\mathrm{Re}(\lambda_j)|$ above some value, determines the effective transfer function behavior [90] and is also known as transfer function modal equivalent:

Definition 4.2 A transfer function modal equivalent $H_k(s)$ is an approximation of a transfer function $H(s)$ that consists of $k < n$ terms:

$$H_k(s) = \sum_{j=1}^{k} \frac{R_j}{s - \lambda_j} + d. \tag{4.32}$$

A modal equivalent that consists of the most dominant terms determines the effective transfer function behavior [90]. If $X \in \mathbb{C}^{n\times k}$ and $Y \in \mathbb{C}^{n\times k}$ are matrices having the left and right eigenvectors $\mathbf{y}_i$ and $\mathbf{x}_i$ of (A, E) as columns, such that $Y^*AX = \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_k)$, with $Y^*EX = I$, then the corresponding (complex) reduced system follows by setting $\mathbf{x} = X\tilde{\mathbf{x}}$ and multiplying from the left by Y^*:

$$\begin{cases} \dot{\tilde{\mathbf{x}}}(t) = \Lambda\tilde{\mathbf{x}}(t) + (Y^*\mathbf{b})u(t) \\ \tilde{y}(t) = (\mathbf{c}^*X)\tilde{\mathbf{x}}(t) + du(t). \end{cases}$$

In practice, it is advisable to make a real reduced model in the following way: for every complex pole triplet $(\lambda, \mathbf{x}, \mathbf{y})$, construct real bases for the right and left eigenspace via $[\mathrm{Re}(\mathbf{x}), \mathrm{Im}(\mathbf{x})]$ and $[\mathrm{Re}(\mathbf{y}), \mathrm{Im}(\mathbf{y})]$, respectively. Let the columns of X_r and Y_r be such bases, respectively. Because the complex conjugate eigenvectors are also in this space, the real bases for the eigenspaces are still (at most) k dimensional. The real reduced model can be formed by using X_r and Y_r in $(Y_r^*EX_r, Y_r^*AX_r, Y_r^*\mathbf{b}, X_r^*\mathbf{c}, d)$.

For stable nondefective systems, the error in the modal equivalent can be quantified as [64]

$$\begin{aligned}\|H - H_k\|_\infty &= \| \sum_{j=k+1}^{n} \frac{R_j}{s-\lambda_j} \|_\infty \\ &\le \sum_{j=k+1}^{n} \frac{|R_j|}{|\mathrm{Re}(\lambda_j)|},\end{aligned}$$

where $\|H\|_\infty$ is the operator norm induced by the 2-norm in the frequency domain [58, 64]. An advantage of modal approximation is that the poles of the modal equivalent are also poles of the original system.

The dominant poles are specific (complex) eigenvalues of the pencil (A, E) and usually form a small subset of the spectrum of (A, E), so that rather accurate modal equivalents may be possible for $k \ll n$. Since the dominant poles can be located anywhere in the spectrum, specialized eigensolution methods are needed. Because the dominance of a pole is independent of d, without loss of generality $d = 0$ in the following.

4.2.2 *Specialized Eigensolution Methods*

In this section we describe the Dominant Pole Algorithm and its extension with deflation and subspace acceleration.

4.2.2.1 The Dominant Pole Algorithm (DPA)

The poles of the transfer function (4.30) are the $\lambda \in \mathbb{C}$ for which $\lim_{s\to\lambda} |H(s)| = \infty$ and can be computed via the roots of $G(s) = 1/H(s)$. Applying Newton's method leads to the scheme

$$s_{k+1} = s_k - \frac{\mathbf{c}^*\mathbf{v}_k}{\mathbf{w}_k^* E \mathbf{v}_k}, \tag{4.33}$$

where $\mathbf{v}_k = (s_k E - A)^{-1}\mathbf{b}$ and $\mathbf{w}_k = (s_k E - A)^{-*}\mathbf{c}$. The algorithm based on this scheme, also known as the Dominant Pole Algorithm (DPA) [72], is shown in Algorithm 4.1. Note that strictly speaking the definition of dominance used here is based on $|R_j|$ (and not on $|R_j|/|\mathrm{Re}(\lambda_j)|$ as in Definition 4.1); observe that in (4.32) $R_j = (\mathbf{c}^*\mathbf{x}_j)(\mathbf{y}_j^*\mathbf{b})$. The subsequent algorithms offer refinements that may lead to additional candidates, in any user-specified dominance criterion, including Definition 4.1.

Algorithm 4.1 The Dominant Pole Algorithm (DPA)

```
INPUT: System $(E, A, \mathbf{b}, \mathbf{c})$, initial pole estimate $s_0$, tolerance $\epsilon \ll 1$
OUTPUT: Approximate dominant pole $\lambda$ (close to $s_0$) and corresponding right and left eigenvectors $\mathbf{x}$ and $\mathbf{y}$
1: Set $k = 0$
2: while not converged do
3:     Solve $\mathbf{v}_k \in \mathbb{C}^n$ from $(s_k E - A)\mathbf{v}_k = \mathbf{b}$
4:     Solve $\mathbf{w}_k \in \mathbb{C}^n$ from $(s_k E - A)^* \mathbf{w}_k = \mathbf{c}$
5:     Compute the new pole estimate
           $$s_{k+1} = s_k - \frac{\mathbf{c}^* \mathbf{v}_k}{\mathbf{w}_k^* E \mathbf{v}_k} = \frac{\mathbf{w}_k^* A \mathbf{v}_k}{\mathbf{w}_k^* E \mathbf{v}_k}$$
6:     The pole $\lambda = s_{k+1}$ with $\mathbf{x} = \mathbf{v}_k$ and $\mathbf{y} = \mathbf{w}_k$ has converged if
           $$\|A\mathbf{v}_k - s_{k+1} E \mathbf{v}_k\|_2 < \epsilon$$
7:     Set $k = k + 1$
8: end while
```

The Dominant Pole Algorithm is closely related to Rayleigh Quotient Iteration [76, 77]: the only difference is that in DPA the right hand-sides in Step 3 and 4 remain fixed, while in Rayleigh Quotient Iterations these are updated with $\mathbf{b} = E\mathbf{v}_{k-1}$ and $\mathbf{c} = E^* \mathbf{w}_{k-1}$ every iteration. See [85] for a detailed comparison.

The two linear systems that need to be solved in step 3 and 4 of Algorithm 4.1 to compute the Newton update in (4.33) can be efficiently solved using one LU-factorization $LU = s_k E - A$, by noting that $U^* L^* = (s_k E - A)^*$. If an exact LU-factorization is not available, one has to use inexact Newton schemes, such as inexact Rayleigh Quotient Iteration and Jacobi-Davidson style methods [67, 89, 91]. In the next section, extensions of DPA are presented that are able to compute more than one eigenvalue in an effective and efficient way.

4.2.2.2 Deflation and Subspace Acceleration

In practical applications often more than one dominant pole is wanted: one is interested in all the dominant poles, no matter what definition of dominance is used. Simply running the single pole algorithm DPA for a number of different initial shifts will most likely result in duplicate dominant poles. A well known strategy to avoid computation of already found eigenvalues is deflation, see for instance [87]. It is also known that subspace acceleration may improve the global convergence: for an $n \times n$ problem, the subspace accelerated algorithm converges within at most n iterations, although in practice it may need only $k \ll n$ iterations and will almost never build a full search space of dimension n, but restart every $k_{max} \ll n$ iterations. The use of subspace acceleration requires that every iteration an approximate pole has to be selected from the available approximations. This also may improve the global convergence, since better approximations than the initial estimate, which may be a rather crude approximation, become available during the process.

In the next subsections, variants of DPA for the computation of more than one pole without and with subspace acceleration are discussed. This variant that does not use subspace acceleration can be implemented efficiently with only constant costs for deflation, while the subspace accelerated variant has better global convergence.

Throughout the rest of this chapter, let the $(n \times k)$ matrices X_k and Y_k have as their columns the normalized (found) right and left eigenvectors $\mathbf{x}_i$ and $\mathbf{y}_i$ $(i = 1, \ldots, k)$ of (A, E), respectively, and let Λ_k be a diagonal $(k \times k)$ matrix with the corresponding eigenvalues on its diagonal, i.e. $\Lambda_k = \mathrm{diag}(\lambda_1, \ldots, \lambda_k)$, $Y_k^* A X_k = \Lambda_k$ and $Y_k^* E X_k = I$. For ease of notation, the subscript k will be omitted if this does not lead to confusion.

4.2.2.3 DPA with Deflation by Restriction

It can be verified that if $X \equiv X_k$ and $Y \equiv Y_k$ have as their columns exact eigenvectors (normalized so that $Y^* E X = I$), then the system $(E_d, A_d, \mathbf{b}_d, \mathbf{c}_d)$, where

$$\begin{aligned} E_d &= (I - EXY^*)E(I - XY^*E), \\ A_d &= (I - EXY^*)A(I - XY^*E), \\ \mathbf{b}_d &= (I - EXY^*)\mathbf{b}, \\ \mathbf{c}_d &= (I - E^*YX^*)\mathbf{c}, \end{aligned}$$

has the same poles, eigenvectors and residues, but with the found λ_i $(i = 1, \ldots, k)$ and corresponding R_i transformed to 0. So in order to avoid recomputing the same pole, DPA could be applied to the deflated system $(E_d, A_d, \mathbf{b}_d, \mathbf{c}_d)$ after having found one or more poles. This would require solves with $(sE_d - A_d)$ and $(sE_d - A_d)^*$ in step 4 and 5 of Algorithm 4.2,[14] but the following theorem shows that it is sufficient to only replace $\mathbf{b}$ by $\mathbf{b}_d$ and $\mathbf{c}$ by $\mathbf{c}_d$ to ensure deflation.

Theorem 4.2 ([80, Thm. 3.3.1]) *The deflated transfer function $H_d(s) = \mathbf{c}_d^*(sE - A)^{-1}\mathbf{b}_d$, where*

$$\mathbf{b}_d = (I - EXY^*)\mathbf{b} \quad \textit{and} \quad \mathbf{c}_d = (I - E^*YX^*)\mathbf{c},$$

has the same poles λ_i and corresponding residues R_i as $H(s) = \mathbf{c}^*(sE - A)^{-1}\mathbf{b}$, but with the residues R_i corresponding to the found poles λ_i $(i = 1, \ldots, k)$ transformed to $R_i = 0$.

[14] Note that $(sE_d - A_d)$ would never be computed explicitly, and that sparse systems $(sE_d - A_d)\mathbf{x} = \mathbf{b}_d$ can be solved efficiently.

Algorithm 4.2 Dominant Pole Algorithm with deflation (DPAd)

```
INPUT: System $(E, A, \mathbf{b}, \mathbf{c})$, initial pole estimates $s_0^1, \dots, s_0^p$, tolerance $\epsilon \ll 1$
OUTPUT: Approximate dominant poles $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_p)$, and corresponding right and left
    eigenvectors $X = [\mathbf{x}_1, \dots, \mathbf{x}_p]$ and $Y = [\mathbf{y}_1, \dots, \mathbf{y}_p]$
Set $k = 0$, $i = 0$, $s_k = s_0^1$
while $i < p$ do
                                        ▷ Continue until $p$ poles have been found
    Solve $\mathbf{v}_k \in \mathbb{C}^n$ from $(s_k E - A)\mathbf{v}_k = \mathbf{b}$
    Solve $\mathbf{w}_k \in \mathbb{C}^n$ from $(s_k E - A)^* \mathbf{w}_k = \mathbf{c}$
    Compute the new pole estimate
        $$s_{k+1} = s_k - \frac{\mathbf{c}^* \mathbf{v}_k}{\mathbf{w}_k^* E \mathbf{v}_k} = \frac{\mathbf{w}_k^* A \mathbf{v}_k}{\mathbf{w}_k^* E \mathbf{v}_k}$$
    if $\|A\mathbf{v}_k - s_{k+1} E \mathbf{v}_k\|_2 < \epsilon$ (with $\|\mathbf{v}_k\|_2 = 1$) then
        Set $i = i + 1$
        Set $\lambda_{ii} = s_{k+1}$
        Set $\mathbf{v}_k = \mathbf{v}_k / (\mathbf{w}_k^* E \mathbf{v}_k)$
        Set $X = [X, \mathbf{v}_k]$ and $Y = [Y, \mathbf{w}_k]$
        Deflate: $\mathbf{b} = \mathbf{b} - E\mathbf{v}_k \mathbf{w}_k^* \mathbf{b}$
        Deflate: $\mathbf{c} = \mathbf{c} - E^* \mathbf{w}_k \mathbf{v}_k^* \mathbf{c}$
        Set $s_{k+1} = s_0^i$
    end if
    Set $k = k + 1$
end while
```

Proof The proof follows by using the definition of residues and basic linear algebra [80, Thm. 3.3.1]. □

In other words, by using $\mathbf{b}_d$ and $\mathbf{c}_d$ the found dominant poles are degraded to non dominant poles of $H_d(s)$, while not changing the dominance of the remaining poles. Hence these poles will not be recomputed by DPA applied to $H_d(s)$. Graphically, the peaks caused by the found poles are 'flattened' in the Bode plot (see also Fig. 4.8).

Note that if $H(s) = \mathbf{c}^*(sE - A)^{-1}\mathbf{b} + d$ with $d = 0$, then the deflated poles in fact become zeros of $H_d(s)$. It can be shown that DPA applied to $H_d(s) = \mathbf{c}_d^*(sE - A)^{-1}\mathbf{b}_d$ and DPA applied to $H_{\tilde{d}}(s) = \mathbf{c}_d^*(sE_d - A_d)^{-1}\mathbf{b}_d$ produce the same results [85].

The important result is that the single pole DPA can easily be extended, see Algorithm 4.2, to an algorithm that is able to compute more than one pole, while maintaining constant costs per iteration, except for iterations in which a pole is found. The only change to be made to Algorithm 4.1, is when a dominant pole triplet $(\lambda, \mathbf{x}, \mathbf{y})$ is found: in that case, the algorithm continues with $\mathbf{b}$ and $\mathbf{c}$ replaced by $(I - E\mathbf{x}\mathbf{y}^*)\mathbf{b}$ and $(I - E^*\mathbf{y}\mathbf{x}^*)\mathbf{c}$, respectively.

This approach has a number of advantages. The implementation is straightforward and efficient: search spaces, selection strategies and orthogonalization procedures are not needed, so that the computational costs per iteration remain constant, even if the number of found poles increases. For every found pole only two skew projections are needed once to compute the new $\mathbf{b}_d$ and $\mathbf{c}_d$, so the costs

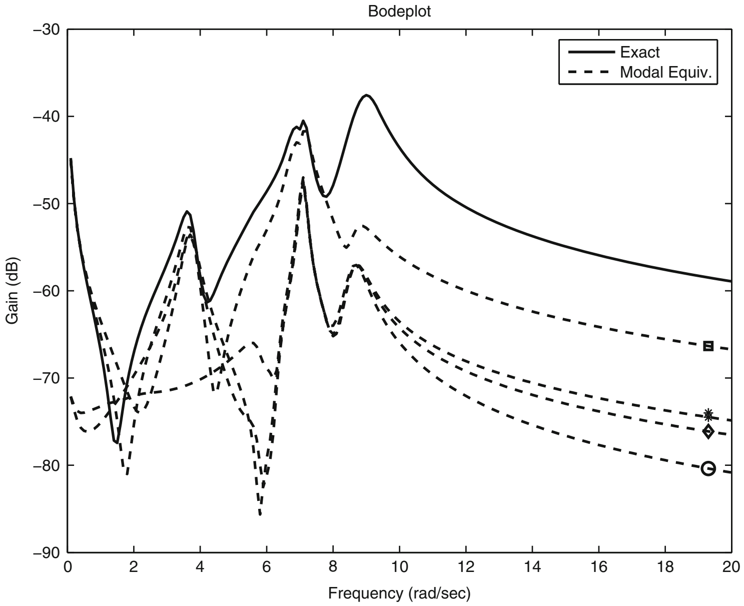

Fig. 4.8 Exact transfer function (*solid*) of the New England test system [72], and modal equivalents where the following dominant pole (*pairs*) are removed one by one: $-0.467 \pm 8.96i$ (*square*), $-0.297 \pm 6.96i$ (*asterisk*), -0.0649 (*diamond*), and $-0.249 \pm 3.69i$ (*circle*). Note that the corresponding peaks are removed from the Bode plot as well (to see this, check the Bode plot at the frequencies near the imaginary part of the removed pole)

for deflation are constant. The pseudo code in Algorithm 4.2 can almost literally be used as Matlab code. The special properties of DPA ensure convergence to dominant poles (locally). Furthermore, the deflation of found poles is numerically stable in the sense that even if the corresponding transformed residues are not exactly zero, which is usually the case in finite arithmetic, this will hardly influence the effect of deflation: firstly, all the poles are left unchanged, and secondly, already a decrease of dominance of the found poles to nondominance (because of the projected in- and output vectors $\mathbf{b}_d$ and $\mathbf{c}_d$) will shrink the local convergence neighborhood of these poles significantly, again because of the convergence behavior of DPA [85].

This approach, however, may still suffer from the fact that the convergence behavior can be very local and hence may heavily depend on the initial estimates s_0^i. Although in practice one often has rather accurate initial estimates of the poles of interest, this may be problematic if accurate information is not available. It may take many iterations until convergence if the initial estimate is not in the neighborhood of a dominant pole. On the other hand, the computational complexity of this problem depends on the costs of the *LU* factorization, which in certain practical examples can be computed very efficiently. In the next section a subspace accelerated version of DPA is described, that improves the global convergence by using search spaces.

Algorithm 4.3 Subspace Accelerated DPA (SADPA)

```
INPUT: System $(E, A, \mathbf{b}, \mathbf{c})$, initial pole estimate $s_1$ and the number of wanted poles $p$
OUTPUT: Dominant pole triplets $(\lambda_i, \mathbf{x}_i, \mathbf{y}_i)$, $i = 1, \ldots, p$
 1: $k = 1$, $p_{found} = 0$, $\Lambda = [\,]$, $X = Y = [\,]$
 2: while $p_{found} < p$ do
 3:     Solve $\mathbf{v}$ from $(s_k E - A)\mathbf{v} = \mathbf{b}$
 4:     Solve $\mathbf{w}$ from $(s_k E - A)^* \mathbf{w} = \mathbf{c}$
 5:     $\mathbf{v} = \mathrm{MGS}(V, \mathbf{v})$, $V = [V, \mathbf{v}/||\mathbf{v}||_2]$
 6:     $\mathbf{w} = \mathrm{MGS}(W, \mathbf{w})$, $W = [W, \mathbf{w}/||\mathbf{w}||_2]$
 7:     Compute $S = W^* A V$ and $T = W^* E V$
 8:     $(\tilde{\Lambda}, \tilde{X}, \tilde{Y}) = \mathrm{Sort}(S, T, W^*\mathbf{b}, V^*\mathbf{c})$          ▷ Algorithm 4.4
 9:     Dominant approximate eigentriplet of $(A, E)$ is

            $(\hat{\lambda}_1 = \tilde{\lambda}_1, \hat{\mathbf{x}}_1 = V\tilde{\mathbf{x}}_1/\|V\tilde{\mathbf{x}}_1\|_2, \hat{\mathbf{y}}_1 = W\tilde{\mathbf{y}}_1/\|W\tilde{\mathbf{y}}_1\|_2)$

10:     if $||A\hat{\mathbf{x}}_1 - \hat{\lambda}_1 E\hat{\mathbf{x}}_1||_2 < \epsilon$ then
11:         $(\Lambda, X, Y, V, W, \mathbf{b}, \mathbf{c}) =$
12:       Deflate$(\hat{\lambda}_1, \hat{\mathbf{x}}_1, \hat{\mathbf{y}}_1, \Lambda, X, Y, V\tilde{X}_{2:k}, W\tilde{Y}_{2:k}, E, \mathbf{b}, \mathbf{c})$          ▷ Algorithm 4.5
13:         $p_{found} = p_{found} + 1$
14:         Set $\tilde{\lambda}_1 = \tilde{\lambda}_2$, $k = k - 1$
15:     end if
16:     Set $k = k + 1$
17:     Set the new pole estimate $s_k = \tilde{\lambda}_1$
18: end while
```

4.2.2.4 Subspace Accelerated DPA

A drawback of DPA is that information obtained in the current iteration is discarded at the end of the iteration. The only information that is preserved is contained in the new pole estimate s_{k+1}. The current right and left approximate eigenvectors $\mathbf{v}_k$ and $\mathbf{w}_k$, however, may also contain components in the direction of eigenvectors corresponding to other dominant poles. Instead of discarding these approximate eigenvectors, they are kept in search spaces spanned by the columns of V and W, respectively. This idea is known as subspace acceleration.

A global overview of SADPA is shown in Algorithm 4.3. Starting with a single shift s_1, the first iteration is equivalent to the first iteration of the DPA (step 3–4). The right and left eigenvector approximations $\mathbf{v}_1$ and $\mathbf{w}_1$ are kept in spaces V and W. In the next iteration, these spaces are expanded orthogonally, by using modified Gram-Schmidt (MGS) [63], with the approximations $\mathbf{v}_2$ and $\mathbf{w}_2$ corresponding to the new shift s_2 (step 5–6). Hence the spaces grow and will contain better approximations.

It can be shown that subspace accelerated DPA, under certain conditions, is equivalent to subspace accelerated Rayleigh Quotient Iteration and the Jacobi-Davidson method, see [80, 85] for more details.

Algorithm 4.4 $(\tilde{\Lambda}, \tilde{X}, \tilde{Y}) = \mathrm{Sort}(S, T, \mathbf{b}, \mathbf{c})$

```
INPUT: $S, T \in \mathbb{C}^{k\times k}, \mathbf{b}, \mathbf{c} \in \mathbb{C}^k$
OUTPUT: $\tilde{\Lambda} \in \mathbb{C}^k$, $\tilde{X}, \tilde{Y} \in \mathbb{C}^{k\times k}$ with $\lambda_1$ the pole with largest (scaled) residue magnitude and
    $\tilde{\mathbf{y}}_1$ and $\tilde{\mathbf{x}}_1$ the corresponding right and left eigenvectors.
1: Compute eigentriplets of the pair $(S, T)$:
       $(\tilde{\lambda}_i, \tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i), \quad \tilde{\mathbf{y}}_i^* T \tilde{\mathbf{x}}_i = 1, \quad i = 1, \dots, k$
2: $\tilde{\Lambda} = [\tilde{\lambda}_1, \dots, \tilde{\lambda}_k]$
3: $\tilde{X} = [\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_k]$
4: $\tilde{Y} = [\tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_k]$
5: Compute residues $R_i = (\mathbf{c}^* \tilde{\mathbf{x}}_i)(\tilde{\mathbf{y}}_i^* \mathbf{b})$
6: Sort $\tilde{\Lambda}$, $\tilde{X}$, $\tilde{Y}$ in decreasing $|R_i|/|\mathrm{Re}(\tilde{\lambda}_i)|$ order
```

Selection Strategy

In iteration k the Petrov-Galerkin approach leads (step 7) to the projected eigenproblem

$$W^*AV\tilde{\mathbf{x}} = \tilde{\lambda} W^*EV\tilde{\mathbf{x}},$$
$$\tilde{\mathbf{y}}W^*AV = \tilde{\lambda}\tilde{\mathbf{y}}W^*EV.$$

Since the interaction matrices $S = W^*AV$ and $T = W^*EV$ are of low dimension $k \ll n$, the eigentriplets $(\tilde{\lambda}_i, \tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)$ of this reduced problem can be computed using the QZ method (or the QR method in the bi-E-orthogonal case) (step 1 of Algorithm 4.4). This provides k approximate eigentriplets $(\hat{\lambda}_i = \tilde{\lambda}_i, \hat{\mathbf{x}}_i = V\tilde{\mathbf{x}}_i, \hat{\mathbf{y}}_i = W\tilde{\mathbf{y}}_i)$ for (A, E). The most natural thing to do is to choose the triplet $(\hat{\lambda}_j, \hat{\mathbf{x}}_j, \hat{\mathbf{y}}_j)$ with the most dominant pole approximation (step 8–9): compute the corresponding residues $\hat{R}_i = (\mathbf{c}^*\hat{\mathbf{x}}_i)(\hat{\mathbf{y}}_i^*\mathbf{b})$ of the k pairs and select the pole with the largest $|\hat{R}_j|/|\mathrm{Re}(\hat{\lambda}_j)|$ (see Algorithm 4.4). The SADPA then continues with the new shift $s_{k+1} = \hat{\lambda}_j$ (step 16).

The residues $\hat{R}_i$ can be computed without computing the approximate eigenvectors explicitly (step 5 of Algorithm 4.4): if the $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{y}}_i$ are scaled so that $\tilde{\mathbf{y}}_i^*T\tilde{\mathbf{x}}_i = 1$ $(= \hat{\mathbf{y}}_i^*E\hat{\mathbf{x}}_i)$, then it follows that the $\hat{R}_i$ can be computed as $\hat{R}_i = ((\mathbf{c}^*V)\tilde{\mathbf{x}}_i)(\tilde{\mathbf{y}}_i^*(W^*\mathbf{b}))$ $(= (\mathbf{c}^*\hat{\mathbf{x}}_i)(\hat{\mathbf{y}}_i^*\mathbf{b}))$.

Instead of $\hat{\mathbf{y}}_i^*E\hat{\mathbf{x}}_i = 1$ one can also use the scaling $\|\hat{\mathbf{y}}_i\|_2 = \|\hat{\mathbf{x}}_i\|_2 = 1$ when computing approximate residues. In that case the product of the angles $\angle(\hat{\mathbf{x}}_i, \mathbf{c})$ and $\angle(\hat{\mathbf{y}}_i, \mathbf{b})$ is used in the computation of the approximate residues (see also [85]), which numerically may be more robust.

Algorithm 4.5 $(\Lambda, X, Y, \tilde{V}, \tilde{W}, \mathbf{b}_d, \mathbf{c}_d) = \text{Deflate}(\lambda, \mathbf{x}, \mathbf{y}, \Lambda, X, Y, V, W, E, \mathbf{b}, \mathbf{c})$

```
INPUT: $\lambda \in \mathbb{C}, \mathbf{x}, \mathbf{y} \in \mathbb{C}^n, \Lambda \in \mathbb{C}^p, X, Y \in \mathbb{C}^{n\times p}, V, W \in \mathbb{C}^{n\times k}, E \in \mathbb{C}^{n\times n}, \mathbf{b}, \mathbf{c} \in \mathbb{C}^n$
OUTPUT: $\Lambda \in \mathbb{C}^q, X, Y \in \mathbb{C}^{n\times q}, \tilde{V}, \tilde{W} \in \mathbb{C}^{n\times k-1}, \mathbf{b}_d, \mathbf{c}_d \in \mathbb{C}^n$, where $q = p + 1$ if $\lambda$ has zero
    imaginary part and $q = p + 2$ if $\lambda$ has nonzero imaginary part.
 1: $\Lambda = [\Lambda, \lambda]$
 2: Set $\mathbf{x} = \mathbf{x}/(\mathbf{y}^* E\mathbf{x})$
 3: $X = [X, \mathbf{x}]$
 4: $Y = [Y, \mathbf{y}]$
 5: Deflate: $\mathbf{b}_d = \mathbf{b} - E\mathbf{x}(\mathbf{y}^*\mathbf{b})$
 6: Deflate: $\mathbf{c}_d = \mathbf{c} - E^*\mathbf{y}(\mathbf{x}^*\mathbf{c})$
 7: if imag($\lambda$) $\neq 0$ then
 8:                                                    ▷ Also deflate complex conjugate
 9:     $\Lambda = [\Lambda, \bar{\lambda}]$
10:     $\mathbf{x} = \bar{\mathbf{x}}$, $X = [X, \mathbf{x}]$
11:     $\mathbf{y} = \bar{\mathbf{y}}$, $Y = [Y, \mathbf{y}]$
12:     Deflate: $\mathbf{b}_d = \mathbf{b}_d - E\mathbf{x}(\mathbf{y}^*\mathbf{b}_d)$
13:     Deflate: $\mathbf{c}_d = \mathbf{c}_d - E^*\mathbf{y}(\mathbf{x}^*\mathbf{c}_d)$
14: end if
15: $\tilde{V} = \tilde{W} = [\,]$
16: for $j = 1, \ldots, k$ do
17:     $\tilde{V} = \text{Expand}(\tilde{V}, X, Y, E, \mathbf{v}_j)$                ▷ Algorithm 4.6
18:     $\tilde{W} = \text{Expand}(\tilde{W}, Y, X, E^*, \mathbf{w}_j)$              ▷ Algorithm 4.6
19: end for
```

Deflation

In each iteration step a convergence test (step 10) is done like in DPAd (Algorithm 4.2): if for the selected eigentriplet $(\hat{\lambda}_1, \hat{\mathbf{x}}_1, \hat{\mathbf{y}}_1)$ the norm of the residual $||A\hat{\mathbf{x}}_1 - \hat{\lambda}_1 E\hat{\mathbf{x}}_1||_2$ is smaller than some tolerance ϵ, it is converged. In general more than one dominant eigentriplet is wanted and it is desirable to avoid repeated computation of the same eigentriplet. The same deflation technique as used in DPAd can be applied here (steps 5–6 and 12–13 of Algorithm 4.5, see also Sect. 4.2.2.3), and since SADPA continues with $\mathbf{b}_d$ and $\mathbf{c}_d$, no explicit E-orthogonalization of expansion vectors against found eigenvectors is needed in step 3 and 4. The effect is similar to the usual deflation in Jacobi-Davidson methods [62]: found eigenvectors are hard-locked, i.e. once deflated, they do not participate and do not improve during the rest of the process (contrary to soft-locking, where deflated eigenvectors still participate in the Rayleigh-Ritz (Ritz-Galerkin) procedure and may be improved, at the cost of additional computations and administration, see [69, 70]). In fact, there is cheap explicit deflation without the need for implicit deflation (cf. [62, remark 5, p. 106], where a combination of explicit and implicit deflation is used).

Algorithm 4.6 $V = \text{Expand}(V, X, Y, E, \mathbf{v})$

```
INPUT: $V \in \mathbb{C}^{n\times k}$ with $V^*V = I$, $X, Y \in \mathbb{C}^{n\times p}$, $E \in \mathbb{C}^{n\times n}$, $\mathbf{v} \in \mathbb{C}^n$, $Y^*EX$ diagonal,
    $Y^*EV = 0$
OUTPUT: $V \in \mathbb{C}^{n\times(k+1)}$ with $V^*V = I$ and
 1: $\mathbf{v}_{k+1} = \prod_{j=1}^{p}(I - \frac{\mathbf{x}_j\mathbf{y}_j^*E}{\mathbf{y}_j^*E\mathbf{x}_j}) \cdot \mathbf{v}$
 2: $\mathbf{v} = \prod_{j=1}^{p}(I - \frac{\mathbf{x}_j\mathbf{y}_j^*E}{\mathbf{y}_j^*E\mathbf{x}_j}) \cdot \mathbf{v}$
 3: $\mathbf{v} = \text{MGS}(V, \mathbf{v})$
 4: $V = [V, \mathbf{v}/||\mathbf{v}||_2]$
```

If an eigentriplet has converged (steps 11–13 of Algorithm 4.3), the eigenvectors are deflated from the search spaces by reorthogonalizing the search spaces against the found eigenvectors. This can be done by using modified Gram-Schmidt (MGS) and by recalling that, if the exact vectors are found, the pencil

$$((I - EXY^*)A(I - XY^*E), \quad (I - EXY^*)E(I - XY^*E))$$

has the same eigentriplets as (A, E), but with the found eigenvalues transformed to zero (Algorithm 4.6, see also [62, 67]). Since in finite arithmetic only *approximations* to exact eigentriplets are available, the computed eigenvalues are transformed to $\eta \approx 0$. The possible numerical consequences of this, however, are limited, since SADPA continues with $\mathbf{b}_d$ and $\mathbf{c}_d$, and as argued in Sect. 4.2.2.3, the residues of the found poles are transformed to (approximately) zero.

If a complex pole has converged, its complex conjugate is also a pole and the corresponding complex conjugate right and left eigenvectors can also be deflated. A complex conjugate pair is counted as one pole. The complete deflation procedure is shown in Algorithm 4.5.

After deflation of the found pole(s), SADPA continues with the second most dominant approximate pole (steps 13–16 of Algorithm 4.3).

Further Improvements and Remarks

It may happen that the search spaces V and W become high-dimensional, especially when a large number of dominant poles is wanted. A common way to deal with this is to do a thick restart [62, 87]: if the subspaces V and W reach a certain maximum dimension $k_{max} \ll n$, they are reduced to a dimension $k_{min} < k_{max}$ by keeping the k_{min} most dominant approximate eigentriplets; the process is restarted with the reduced V and W (already converged eigentriplets are not part of the active subspaces V and W). This procedure is repeated until all poles are found.

Furthermore, as more eigentriplets have converged, approximations of new eigentriplets may become poorer or convergence may be hampered, due to rounding errors in the orthogonalization phase and the already converged eigentriplets. It is therefore advised to take a small tolerance $\epsilon \leq 10^{-10}$. Besides that, as the estimate converges to a dominant pole, the right and left eigenvectors computed in step 3 and 4 of Algorithm 4.3 are usually more accurate than the approximations computed in the selection procedure: if the estimate s_k is close to an eigenvalue λ, then $s_k E - A$ may become ill-conditioned, but, as is discussed in [79] and [78, Section 4.3], the solutions $\mathbf{v}_k$ and $\mathbf{w}_k$ have large components in the direction of the corresponding right and left eigenvectors (provided $\mathbf{b}$ and $\mathbf{c}$ have sufficiently large components in those directions). In the deflation phase, it is therefore advised to take the most accurate of both, i.e., the approximate eigenvector with smallest residual. It may also be advantageous to do an additional step of two-sided Rayleigh quotient iteration to improve the eigenvectors.

SADPA requires only one initial estimate. If rather accurate initial estimates are available, one can take advantage of this in SADPA by setting the next estimate after deflation to a new initial estimate (step 16 of Algorithm 4.3).

Every iteration, two linear systems are to be solved (step 3 and 4). As was already mentioned, this can efficiently be done by computing one LU-factorization and solving the systems by using L and U, and U^* and L^*, respectively. Because in practice the system matrices A and E are often very sparse and structured, computation of the LU-factorizations can be relatively inexpensive.

The selection criterion can easily be changed to another of the several existing indices of modal dominance [57, 64, 94]. Furthermore, the strategy can be restricted to considering only poles in a certain frequency range. Also, instead of providing the number of wanted poles, the procedure can be automated even further by providing the desired maximum error $|H(s) - H_k(s)|$ for a certain frequency range: the procedure continues computing new poles until the error bound is reached. Note that such an error bound requires that the transfer function of the complete model can be evaluated efficiently for the frequency range of interest.

A Numerical Example

For illustrational purposes, SADPA was applied to a transfer function of the New England test system, a model of a power system. This small benchmark system has 66 state variables (for more information, see [72]). The tolerance used was $\epsilon = 10^{-10}$ and no restarts were used. Every iteration, the pole approximation $\hat{\lambda}_j$ with largest $|\hat{R}_j|/|\mathrm{Re}(\hat{\lambda}_j)|$ was selected. Table 4.1 shows the found dominant poles and the iteration number for which the pole satisfied the stopping criterion. Bodeplots of two modal equivalents are shown in Fig. 4.9. The quality of the modal equivalent increases with the number of found poles, as can be observed from the better match of the exact and reduced transfer function.

Table 4.1 Results for SADPA applied to the New England test system ($s_1 = 1i$)

#Poles	#States	New pole	Iteration	Bodeplot
1	2	$-0.4672 \pm 8.9644i$	13	–
2	4	$-0.2968 \pm 6.9562i$	18	–
3	5	-0.0649	21	Fig. 4.9 (left)
4	7	$-0.2491 \pm 3.6862i$	25	–
5	9	$-0.1118 \pm 7.0950i$	26	–
6	11	$-0.3704 \pm 8.6111i$	27	Fig. 4.9 (right)

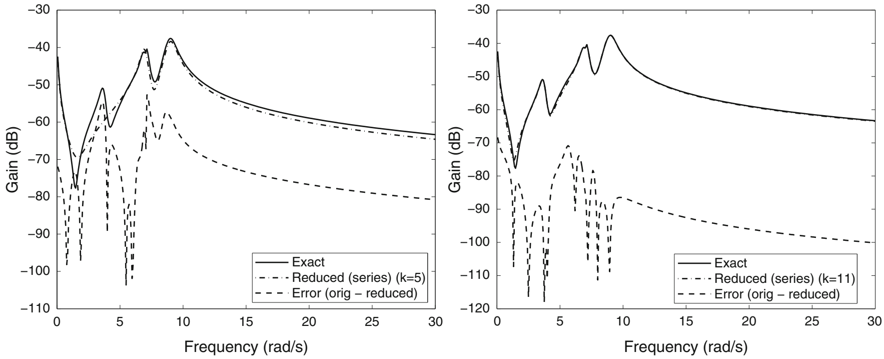

Fig. 4.9 Bode plot of 5th order (*left*) and 11th order (*right*) modal equivalent, complete model and error for the transfer function of the New England test system (66 states in the complete model)

4.2.3 Generalizations to Other Eigenvalue Problems

In this section, four variants of the dominant pole algorithm presented in the previous section are briefly discussed. In Sect. 4.2.3.1, the theory is extended to multi-input multi-output systems. A variant of DPA that computes the dominant zeros of a transfer function is described in Sect. 4.2.3.2. Section 4.2.3.3 describes the extension to higher-order dynamical systems. Finally, in Sect. 4.2.3.4 it is shown how DPA can be used for the computation of eigenvalues most sensitive to parameter changes.

4.2.3.1 MIMO Systems

For a multi-input multi-output (MIMO) system

$$\begin{cases} E\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \\ \mathbf{y}(t) \quad = C^*\mathbf{x}(t) + D\mathbf{u}(t), \end{cases}$$

where $A, E \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{n\times p}$, $\mathbf{x}(t) \in \mathbb{R}^n$, $\mathbf{u}(t) \in \mathbb{R}^m$, $\mathbf{y}(t) \in \mathbb{R}^p$ and $D \in \mathbb{R}^{p\times m}$, the transfer function $H(s) : \mathbb{C} \to \mathbb{C}^{p\times m}$ is defined as

$$H(s) = C^*(sE - A)^{-1}B + D. \tag{4.34}$$

The dominant poles of (4.34) are those $s \in \mathbb{C}$ for which the largest singular value $\sigma_{\max}(H(s)) \to \infty$. For square transfer functions ($m = p$), there is an equivalent criterion: the dominant poles are those $s \in \mathbb{C}$ for which the absolute smallest eigenvalue $|\lambda_{\min}(H^{-1}(s))| \to 0$. This leads, for square transfer functions, to the following Newton scheme:

$$s_{k+1} = s_k - \frac{1}{\mu_{\min}} \frac{1}{\mathbf{v}^* C^*(s_k E - A)^{-2} B\mathbf{u}},$$

where $(\mu_{\min}, \mathbf{u}, \mathbf{v})$ is the eigentriplet of $H^{-1}(s_k)$ corresponding to $\lambda_{\min}(H^{-1}(s_k))$. For a dominant pole, the mountain spreads of $\sigma_{\max}$ are larger and, therefore, the neighborhood of convergence attraction is larger than for a less dominant pole, which would show just a spike. An algorithm for computing the dominant poles of a MIMO transfer function can be readily derived from Algorithm 4.1. The reader is referred to [74] for the initial MIMO DPA algorithm and to [81] for an algorithm SAMDP, similar to SADPA, generalizations to non-square MIMO systems and more details.

4.2.3.2 Computing the Zeros of a Transfer Function

The zeros of a transfer function $H(s) = \mathbf{c}^*(sE - A)^{-1}\mathbf{b} + d$ are those $s \in \mathbb{C}$ for which $H(s) = 0$. An algorithm, similar to Algorithm 4.1, can be derived by noting that a Newton scheme for computing the zeros of a transfer function is given by

$$s_{k+1} = s_k + \frac{\mathbf{c}^*(s_k E - A)^{-1}\mathbf{b} + d}{\mathbf{c}^*(s_k E - A)^{-2}\mathbf{b}}. \tag{4.35}$$

In fact, it can be shown that the dominant zeros can be computed as the dominant poles of the inverse transfer function $[H(s)]^{-1} = \mathbf{c}_z^*(sE_z - A_z)^{-1}\mathbf{b}_z + d_z$, which has the realization

$$A_z = \begin{bmatrix} A & \mathbf{b} \\ \mathbf{c}^T & d \end{bmatrix}, \; E_z = \begin{bmatrix} E & 0 \\ 0 & 0 \end{bmatrix},$$

$$\mathbf{b}_z = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad \mathbf{c}_z = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad d_z = 0,$$

In other words, the dominant zeros of $H(s)$ can be computed by applying DPA to $[H(s)]^{-1}$. See [73] for further details.

4.2.3.3 Polynomial Eigenvalue Problems

The main idea of using Newton's method to find dominant poles can be generalized to higher order systems [84]. For the second-order transfer function $H(s) = \mathbf{c}^*(s^2M + sC + K)^{-1}\mathbf{b}$, for instance, the scheme becomes

$$s_{k+1} = s_k - \frac{\mathbf{c}^*\mathbf{v}}{\mathbf{w}^*(2s_kM + C)\mathbf{v}},$$

where $\mathbf{v} = (s_k^2M + s_kC + K)^{-1}\mathbf{b}$ and $\mathbf{w} = (s_k^2M + s_kC + K)^{-*}\mathbf{c}$. The efficient use of subspace acceleration on large scale second-order eigenvalue problems is described in [84].

4.2.3.4 Computing Eigenvalues Sensitive to Parameter Changes

Let $p \in \mathbb{R}$ be a system parameter (e.g., a resistor value R, or $1/R$, in an electric circuit), and let $A(p)$ and $E(p)$ be matrices that depend on p. The derivative of an eigenvalue λ of the pencil $(A(p), E(p))$, with left and right eigenvectors $\mathbf{y} \equiv \mathbf{y}(p)$ and $\mathbf{x} \equiv \mathbf{x}(p)$, to a parameter p is given by [66, 75]

$$\frac{\partial\lambda}{\partial p} = \frac{\mathbf{y}^*(\frac{\partial A}{\partial p} - \lambda\frac{\partial E}{\partial p})\mathbf{x}}{\mathbf{y}^*E\mathbf{x}}. \tag{4.36}$$

The derivative (4.36) is often called the sensitivity (coefficient) of λ. Assuming that $\frac{\partial E}{\partial p} = 0$, with $\mathbf{y}$ and $\mathbf{x}$ scaled so that $\mathbf{y}^*E\mathbf{x} = 1$, the eigenvalue derivative (4.36) becomes

$$\frac{\partial\lambda}{\partial p} = \mathbf{y}^*\frac{\partial A}{\partial p}\mathbf{x}. \tag{4.37}$$

The larger the magnitude of the derivative (4.37), the more sensitive eigenvalue λ is to changes in parameter p. In practical applications such information is useful when, for instance, a system needs to be stabilized by moving poles from the right half-plane to the left half-plane [83, 95].

Suppose that the derivative of A to parameter p has rank 1 and hence can be written as

$$\frac{\partial A}{\partial p} = \mathbf{b}\mathbf{c}^*, \tag{4.38}$$

where $\mathbf{b}, \mathbf{c} \in \mathbb{R}^n$ are vectors. Then the sensitivity of an eigenvalue λ with left and right eigenvectors $\mathbf{y}$ and $\mathbf{x}$ (with $\mathbf{y}^*E\mathbf{x} = 1$) becomes

$$\frac{\partial\lambda}{\partial p} = \mathbf{y}^*\frac{\partial A}{\partial p}\mathbf{x} = (\mathbf{y}^*\mathbf{b})(\mathbf{c}^*\mathbf{x}) = (\mathbf{c}^*\mathbf{x})(\mathbf{y}^*\mathbf{b}). \tag{4.39}$$

In the right-hand side of (4.39) one recognizes the residues of the transfer function $H(s) = \mathbf{c}^*(sE - A)^{-1}\mathbf{b}$. Consequently, the most sensitive eigenvalues of the pencil $(A(p), E)$ can be computed by applying DPA to $(E, A, \mathbf{b}, \mathbf{c})$, with $\mathbf{b}$ and $\mathbf{c}$ defined by (4.38).

If $\frac{\partial A}{\partial p}$ has rank higher than 1, one can change Algorithm 4.1 as follows to compute the most sensitive eigenvalues: replace $\mathbf{b}$ and $\mathbf{c}$ by $\frac{\partial A}{\partial p}\mathbf{v}_{k-1}$ and $\left(\frac{\partial A}{\partial p}\mathbf{w}_{k-1}\right)^*$, respectively. The algorithm based on this is called SASPA. For more details and generalizations to higher rank derivatives and multiparameter systems, see [83].

Having obtained, with the use of SADPA [82] or SAMDP [81], a reduced model for a large scale system incorporating feedback controllers at nominal parameters, one may want to find other reduced models for off-nominal parameters in these controllers. The SADPA and SAMDP are ideal algorithms for this application, since they benefit from the reduced model information for the nominal parameters. Note that only a true modal equivalent can benefit from this sensitivity feature, through the use of the SASPA [83].

4.2.4 *Improving Krylov Models by Using Dominant Poles*

It is well known that for some examples moment matching works well, while reduced order models computed by modal approximation are of low quality, and the other way around [58, 80]. Generally speaking, modal approximation performs best if there are $k \ll n$ dominant poles with residues much larger than the residues of the non-dominant poles. In other words, there is a $k \ll n$ for which one has $|R_1| \geq |R_2| \geq \ldots \geq |R_k| \gg |R_{k+1}| \geq |R_{n-1}| \geq |R_n|$, so that truncation at the kth pole does not give a large error [64]. Moment matching based approaches, on the other hand, perform best if the moments show a similar steep decay. There is, however, one additional complication for Krylov based moment matching approaches, that is best explained by an example. Figure 4.10 shows the Bode magnitude plots of an exact transfer function and of two reduced order models: one modal approximation and a moment matching approximation. While the modal approximation captures the dominant dynamics, the moment matching model deviates for $\omega > 4$ rad/s.

The modal approximation matches the original transfer function well because it is built from the 7 most dominant poles. The moment matching Arnoldi model ($k = 30$) was built using left and right Krylov subspaces with shift $s_0 = 0$. Therefore, the match for frequencies up to $\omega = 4$ rad/s is good. For higher frequencies, however, this approach suffers from a well known property of Arnoldi methods, that were originally developed for the computation of eigenvalues: the eigenvalue approximations, or Ritz values, tend to approximate the eigenvalues at the outside of the spectrum [93]. This can also be seen in Fig. 4.11, where the circles denote the poles of the moment matching model (note the inverses of the poles are shown): they match the eigenvalues at the outside. The dominant poles, however,

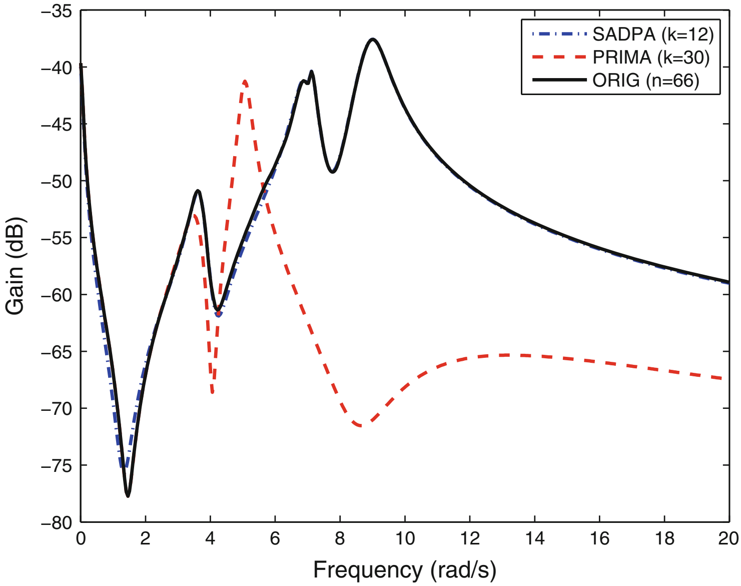

Fig. 4.10 Frequency response of complete system ($n = 66$), modal approximation ($k = 12$), and dual Arnoldi model ($k = 30$)

may be located anywhere in the spectrum, as can also be seen in Fig. 4.11 (squares). This explains why the Arnoldi model fails to capture the peaks.

Based on the above observations and theory in [65], the idea is to use the imaginary parts of dominant poles as shifts for the rational Krylov approach, so that resonance peaks located well within the system frequency bandwidth can also be captured by Krylov methods. The combined dominant pole – Krylov approach can be summarized as follows:

1. Compute $k \ll n$ dominant poles $\lambda_j = \alpha_j \pm \beta_j i$, with $j = 1, \dots k$ and $i = \sqrt{-1}$.
2. Choose interpolation points $s_j = \beta_j i$.
3. Construct $V_j, W_j \in \mathbb{C}^{n \times k_j}$ such that their columns are bases for the rational Krylov [86] spaces

$$\text{colspan}(V_j) = \mathscr{K}^{k_j}((s_j E - A)^{-1} E, (s_j E - A)^{-1} \mathbf{b})$$

and

$$\text{colspan}(W_j) = \mathscr{K}^{k_j}((s_j E - A)^{-*} E^*, (s_j E - A)^{-*} \mathbf{c}),$$

respectively.

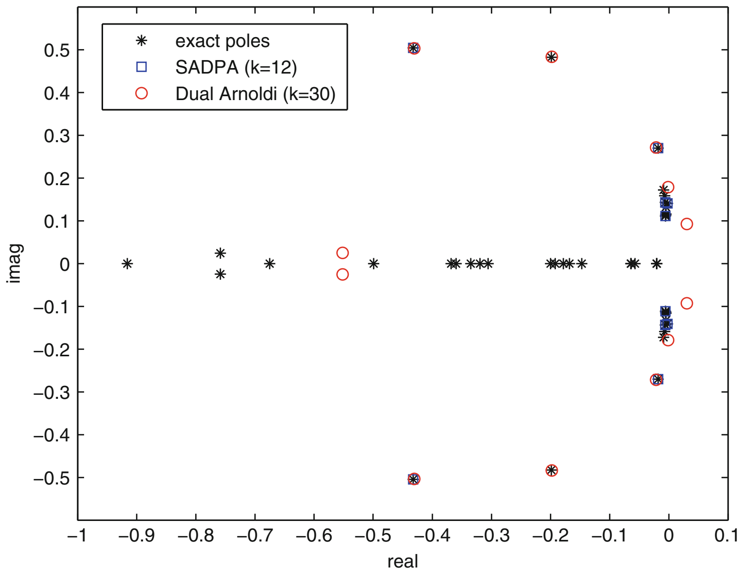

Fig. 4.11 Relevant part of pole spectrum of complete system ($n = 66$), modal approximation ($k = 12$), and dual Arnoldi model ($k = 30$)

4. Project with $V = \text{orth}([V_1, \ldots, V_k])$ and $W = \text{orth}([W_1, \ldots, W_k])$, where orth constructs an orthogonal basis for the spaces. The size of the reduced model is at most $K = \sum_{j=1}^{k} k_j$, matching $2K$ moments.

4.2.5 Numerical Examples

4.2.5.1 Brazilian Interconnected Power System (BIPS)

The Brazilian Interconnected Power System (BIPS) is a year 1999 planning model that has been used in practice (see [82] for more technical details). The size of the sparse matrices A and E is $n = 13{,}251$ (the number of states in the dense state space realization is 1,664). The corresponding transfer function has a non-zero direct transmission term d. Figure 4.12 shows the frequency response of the complete model and the reduced model (41 states) together with the error. Both the magnitude and the phase plots show good matches of the exact and the reduced transfer functions (a relative error of approximately $||H(s) - H_k(s)||/||H_k(s)|| = 0.1$, also for the DC-gain $H(0)$). Figure 4.13 shows the corresponding step response

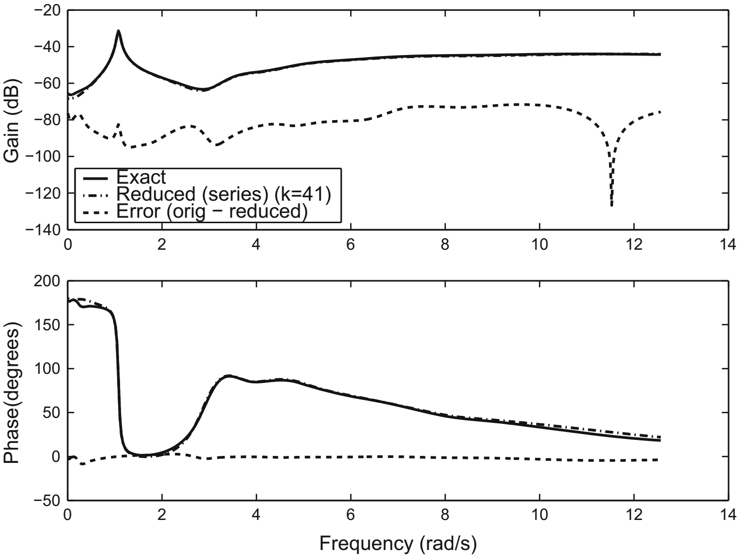

Fig. 4.12 Bode plot (with modulus and phase) of the modal equivalent, the complete model and the error for the transfer function $P_{sc}(s)/B_{sc}(s)$ of BIPS (41 in the modal equivalent, 1664 in the complete model)

(step $u = 0.01$).[15] The reduced model nicely captures the system oscillations. The reduced model (30 poles, 56 states) was computed by SADPA in 341 *LU*-factorizations ($k_{min} = 1$, $k_{max} = 10$). This reduced model could be reduced further to 41 states (22 poles) by removing less dominant contributions, without decreasing the quality of the reduced model much.

Sensitivity of BIPS

To study the sensitivity of the dominant poles and system stability of BIPS, the gain (Kpss) of one of the generators is varied between 0 and 30, with increments of 0.5. Figure 4.14 shows the traces for the most sensitive poles as computed by SASPA (Sect. 4.2.3.4, see also [83]). The CPU time needed for the 60 runs was 1,450 s. A root-locus plot for all poles, computed using the QR method, confirmed that the most sensitive poles were found, but needed 33,600 s. Hence, for large-scale systems, SASPA is a very effective and efficient way to produce relevant root-locus plots.

[15] If $h_k(t)$ is the inverse Laplace transform of $H_k(s)$, the step response for step $u(t) = c$ of the reduced model is given by $y(t) = \int_0^t h(t)u(t) = c(\sum_{i=1}^{k}(\frac{R_i}{\lambda_i}(\exp(\lambda_i t) - 1)) + d)$.

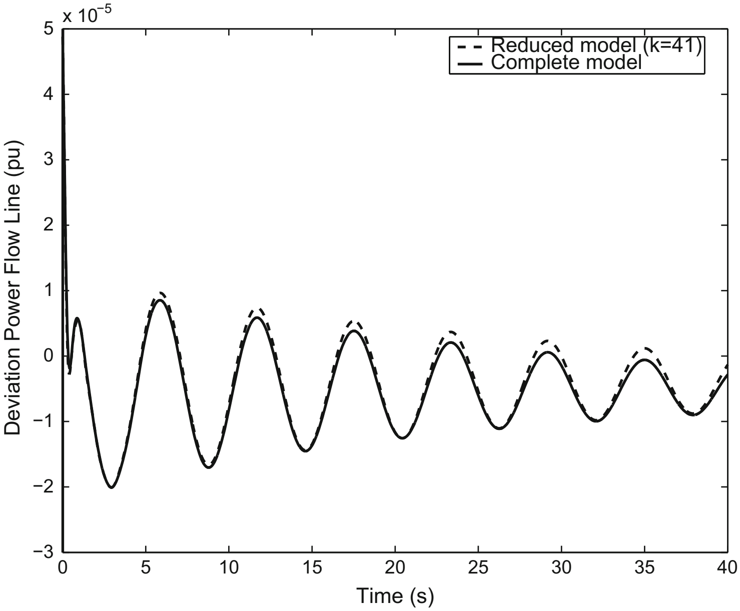

Fig. 4.13 Step responses for transfer function $P_{sc}(s)/B_{sc}(s)$ of BIPS, complete model and modal equivalent (41 states in the modal equivalent, 1664 in the complete model, step disturbance of 0.01 pu)

4.2.5.2 The Breathing Sphere

Figure 4.15 shows the frequency response of a 70th order Second-Order Arnoldi [59] reduced model of vibrating body from sound radiation analysis ($n = 17{,}611$ degrees of freedom, see [71]), that was computed using the complex parts $i\beta$ of five dominant poles $\lambda = \alpha + i\beta$ (computed by Quadratic DPA [84]) as interpolation points, as described in Sect. 4.2.4. This model is more accurate than reduced order models based on standard Krylov methods and matches the peaks up to $\omega = 1$ rad/s, because of use of shifts near the resonance frequencies. This model is a good example of the combined dominant pole – rational Krylov approach, since modal approximations of similar quality require too much CPU time, while Krylov models with uniformly spaced shifts do not capture the peaks.

4.2.6 Concluding Remarks

In this chapter eigenvalue methods, based on the Dominant Pole Algorithm, for the computation of a few specific eigenvalues were discussed. The methods can be used to solve large-scale eigenvalue problems arising in real-life applications

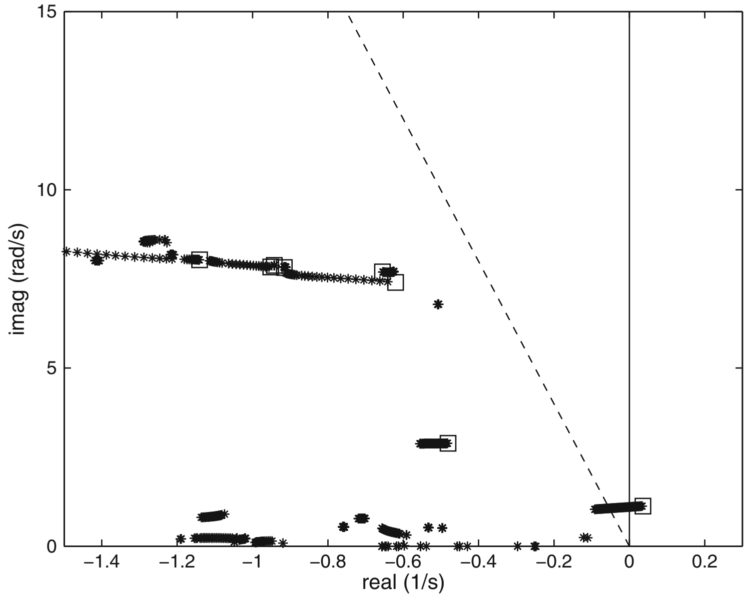

Fig. 4.14 Root locus plot of sensitive poles computed by SASPA. As the gain increases, the critical rightmost pole crosses the imaginary axis and the 5 % damping ratio boundary. Squares denote initial pole locations

and simulation of dynamical systems, for instance for the computation of transfer function dominant poles and zeros, and eigenvalues most sensitive to parameter changes. Furthermore, the corresponding eigenvectors can be used for construction of reduced-order models (modal equivalents) or to improve Krylov-based models. The dominant poles can be used to determine shifts in rational Krylov methods for computing reduced-order models. The practical application of the algorithms was illustrated by numerical experiments with real-life examples.

4.3 Passivity Preserving Model Order Reduction

In this Section we are concerned with dynamical systems $\sum = (\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ of the form

$$\begin{cases} \mathbf{E}\dot{\mathbf{x}}(t) = \ \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \ \mathbf{y}(t) \ = \mathbf{C}^*\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \end{cases} \tag{4.40}$$

where $\mathbf{A}, \mathbf{E} \in \mathbb{R}^{n\times n}$, $\mathbf{E}$ may be singular (we assume $\mathbf{E}$ is symmetric and positive (semi) definite), $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{n\times p}$, $\mathbf{D} \in \mathbb{R}^{p\times m}$, $\mathbf{x}(t) \in \mathbb{R}^n$, $\mathbf{y}(t) \in \mathbb{R}^p$ and

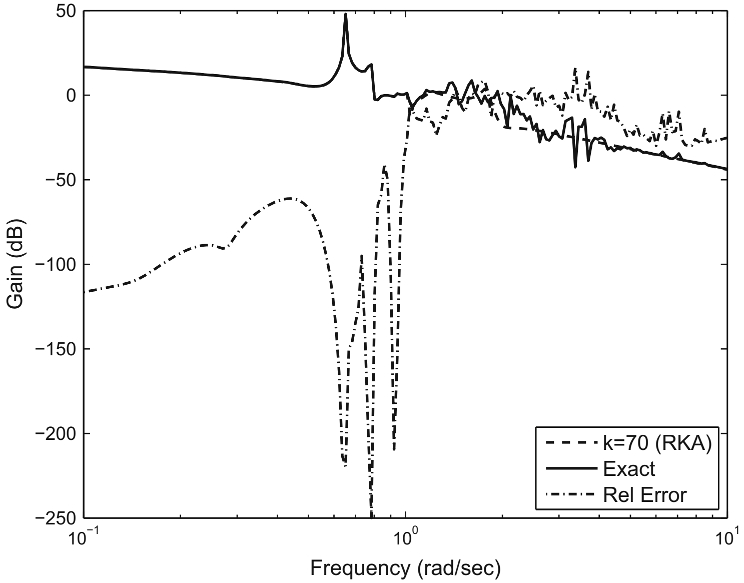

Fig. 4.15 Exact and reduced system transfer functions for a vibrating body, computed by a rational Krylov method with resonance frequencies as complex interpolation points

$\mathbf{u}(t) \in \mathbb{R}^m$.[16] The matrix **E** is called the *descriptor matrix*, the matrix **A** is called the *state space matrix*, the matrices **B** and **C** are called the *input* and *output map*, respectively, and **D** is the *direct transmission map*. The vectors $\mathbf{u}(t)$ and $\mathbf{x}(t)$ are called the *input* and the *state vector*, respectively, and $\mathbf{y}(t)$ is called the *output of the system*. The dimension n of the state is defined as the *complexity* of the system $\sum$. These systems often arise in circuit simulation, for instance, and in this application the system $\sum$ is often *passive*.[17]

The transfer function $\mathbf{G} : \mathbb{C}^m \rightarrow \mathbb{C}^p$, of (4.40),

$$\mathbf{G}(s) = \mathbf{C}^*(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D},$$

can be obtained by applying the Laplace transform to (4.40) under the condition $\mathbf{x(0)} = \mathbf{0}$. The transfer function relates outputs to inputs in the frequency domain via $\mathbf{Y}(s) = \mathbf{G}(s)\mathbf{U}(s)$, where $\mathbf{Y}(s)$ and $\mathbf{U}(s)$ are the Laplace transforms to $\mathbf{y}(t)$ and $\mathbf{u}(t)$, respectively.

[16] Section 4.3 has been written by: Maryam Saadvandi and Joost Rommes. For further details see the MSc-Thesis of the first author [108]. Her further research is found in her Ph.D.-Thesis [109].

[17] Passivity condition is one of the important concepts and many researches have been studying it, [97–101, 104, 106, 107].

We want to reduce the original system $\sum$ to a reduced order model $\hat{\sum} = (\hat{\mathbf{E}}, \hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}, \mathbf{D})$

$$\begin{cases} \hat{E}\dot{\hat{\mathbf{x}}}(t) = & \hat{A}\hat{\mathbf{x}}(t) + \hat{B}\mathbf{u}(t) \\ \hat{\mathbf{y}}(t) & = \hat{\mathbf{C}}^*\hat{\mathbf{x}}(t) + D\mathbf{u}(t), \end{cases} \tag{4.41}$$

where $\hat{\mathbf{A}}, \hat{\mathbf{E}} \in \mathbb{R}^{k\times k}$, $\hat{\mathbf{B}} \in \mathbb{R}^{k\times m}$, $\hat{\mathbf{C}} \in \mathbb{R}^{k\times p}$, $\mathbf{D} \in \mathbb{R}^{p\times m}$, $\hat{\mathbf{x}}(t) \in \mathbb{R}^k$, $\hat{\mathbf{y}}(t) \in \mathbb{R}^p$, $\mathbf{u}(t) \in \mathbb{R}^m$ and $k \ll n$.

It is important to produce a reduced model that preserves stability and passivity.

Remark 4.1 Throughout the reminder of this chapter it is assumed that:

- $m = p$ such that $\mathbf{B} \in \mathbb{R}^{n\times p}$, $\mathbf{C} \in \mathbb{R}^{p\times n}$ and $\mathbf{D} \in \mathbb{R}^{p\times p}$.
- $\mathbf{A}$ is a stable matrix i.e. $Re(\lambda_i) < 0$ with $\lambda_i \in \sigma(\mathbf{A}), i = 1, \cdots, n$.
- The system $\sum$ is observable and controllable [112] and it is passive.

Spectral zeros play an important role in guaranteeing passivity as will be explained in the next sections. In Section 4.3.3 the spectral zeros and the method for computing them are introduced. In the following we describe two projection reduced order methods from literature for reducing the system, that aim to produce a reduced transfer function, which has the specified roots at selected spectral zeros. These methods have been developed by Sorensen [110] and Antoulas [96].

4.3.1 Model Reduction via Projection Matrices

We assume that $\mathcal{M}$ and $\mathcal{N}$ are k-dimensional subspaces of $\mathbb{R}^n$. $\mathbf{V}$ and $\mathbf{W}$ are built for reducing the system by a projection method. So we construct $\mathbf{V} = \{\mathbf{v}_1, \cdots, \mathbf{v}_k\} \in \mathbb{R}^{n\times k}$, of which the column vectors $\mathbf{v}_i$ form a basis of $\mathcal{M}$, and $\mathbf{W} = \{\mathbf{w}_1, \cdots, \mathbf{w}_k\} \in \mathbb{R}^{n\times k}$, of which the column vectors $\mathbf{w}_j$ form a basis of $\mathcal{N}$ (we are interested in $W^*V = I_k$). We assume that $\mathbf{V}$ and $\mathbf{W}$ are time-invariant.

We suppose $\mathbf{x} \in \mathcal{M}$ is an approximate solution of Σ. Hence we can write $\mathbf{x} = \mathbf{V}\hat{\mathbf{x}}$, where $\hat{\mathbf{x}} \in \mathbb{R}^k$ and $\dot{\mathbf{x}} = \mathbf{V}\dot{\hat{\mathbf{x}}}$. Then the residual is

$$\mathbf{E}\dot{\mathbf{x}} - \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{u} = \mathbf{E}\mathbf{V}\dot{\hat{\mathbf{x}}} - \mathbf{A}\mathbf{V}\hat{\mathbf{x}} - \mathbf{B}\mathbf{u}.$$

Next, we assume that this residual is orthogonal to $\mathcal{N}$

$$\begin{aligned} \mathbf{W}^*(\mathbf{E}\mathbf{V}\dot{\hat{\mathbf{x}}} - \mathbf{A}\mathbf{V}\hat{\mathbf{x}} - \mathbf{B}\mathbf{u}) \quad &= \mathbf{0}, \\ \Rightarrow \mathbf{W}^*\mathbf{E}\mathbf{V}\dot{\hat{\mathbf{x}}} - \mathbf{W}^*\mathbf{A}\mathbf{V}\hat{\mathbf{x}} - \mathbf{W}^*\mathbf{B}\mathbf{u} &= \mathbf{0}. \end{aligned}$$

Then the reduced model $\hat{\Sigma}$ becomes:

$$\begin{cases} \hat{\mathbf{E}}\dot{\hat{\mathbf{x}}}(t) = & \hat{\mathbf{A}}\hat{\mathbf{x}}(t) + \hat{\mathbf{B}}\mathbf{u}(t), \\ \hat{\mathbf{y}}(t) & = \hat{\mathbf{C}}^*\hat{\mathbf{x}}(t) + \mathbf{D}\mathbf{u}(t), \end{cases}$$

where $\hat{\mathbf{A}} = \mathbf{W}^*\mathbf{A}\mathbf{V} \in \mathbb{R}^{k\times k}$, $\hat{\mathbf{E}} = \mathbf{W}^*\mathbf{E}\mathbf{V} \in \mathbb{R}^{k\times k}$, $\hat{\mathbf{B}} = \mathbf{W}^*\mathbf{B} \in \mathbb{R}^{k\times m}$, $\hat{\mathbf{C}} = \mathbf{C}\mathbf{V} \in \mathbb{R}^{k\times p}$, $\hat{\mathbf{x}}(t) = \mathbf{V}\hat{\mathbf{x}} \in \mathbb{R}^k$ and $\mathbf{y} = \hat{\mathbf{y}}(t) \in \mathbb{R}^p$ [105].

4.3.2 Passive Systems

We can reduce the model by $\mathbf{V}$ and $\mathbf{W}$, which are constructed in the previous Sect. 4.3.1. With arbitrary $\mathbf{V}$ and $\mathbf{W}$, some features of the original system may not be preserved. One of these properties, which we are interested in to preserve, is *passivity*.

The matrix $\mathbf{A}$ is assumed to be stable, which means all its eigenvalues are in the open left half-plane. Passivity is defined using an energy concept.

Definition 4.3 A system is *passive* if it does not generate energy internally, and *strictly passive* if it consumes or dissipates input energy [110].

In other words Σ is *passive if*

$$Re \int_{-\infty}^{t} \mathbf{u}(\tau)^*\mathbf{y}(\tau)d\tau \geq 0, \qquad \forall t \in \mathbb{R}, \quad \forall \mathbf{u} \in \mathcal{L}_2(\mathbb{R})$$

or *strictly passive* if

$$\exists\delta > 0 \quad \text{s.t. } Re \int_{-\infty}^{t} \mathbf{u}(\tau)^*\mathbf{y}(\tau)d\tau \geq \delta \;\cdot Re \int_{-\infty}^{t} \mathbf{u}(\tau)^*\mathbf{u}(\tau)d\tau, \;\; \forall t \in \mathbb{R}, \;\; \forall \mathbf{u} \in \mathcal{L}_2(\mathbb{R})$$

Another more practical definition of passivity is based on the transfer function $\mathbf{G}(s)$ in the frequency domain:

Definition 4.4 [110] The system Σ is passive iff the transfer function $\mathbf{G}(s)$ is positive real, which means that:

1. $\mathbf{G}(s)$ is analytic for $Re(s) > 0$,
2. $\mathbf{G}(\bar{s}) = \overline{\mathbf{G}(s)}$, $\forall s \in \mathbb{C}$,
3. $\mathbf{G}(s) + (\mathbf{G}(s))^* \geq 0$ for $Re(s) > 0$ where

$$(\mathbf{G}(s))^* = \mathbf{B}^*(s\mathbf{E}^* - \mathbf{A}^*)^{-1}\mathbf{C} + \mathbf{D}^*.$$

We try to construct the **V** and **W** in such a way that the transfer function of the reduced model has the above three properties. Property 3 implies the existence of a stable rational matrix function $\mathbf{K}(s) \in \mathbb{R}^{p\times p}$ (with stable inverse) such that

$$\mathbf{G}(s) + (\mathbf{G}(-s))^* = \mathbf{K}(s)\mathbf{K}^*(-s).$$

We prove this only for the scalar case $p = 1$ of the transfer function. Let $G(s)$ be a scalar, positive-real transfer function with real coefficients. The spectral zeros of G are defined as the zeros of $G(s) + G^*(-s)$. Since all coefficients of G are real, we have $G^*(-s) = G(-s)$. Since $G(s)$ is scalar, we can write $G(s) = \frac{n(s)}{d(s)}$, where $n(s)$ and $d(s)$ are polynomials of degree $\leq k + 1$ (in this note we assume k is even; a similar explanation holds when k is odd). Note that $(G(-s))^* = \frac{n^*(-s)}{d^*(-s)}$. Now we have

$$\begin{aligned} G(s) + (G(-s))^* &= \frac{n(s)}{d(s)} + \frac{n^*(-s)}{d^*(-s)} \\ &= \frac{n(s)d^*(-s) + d(s)n^*(-s)}{d(s)d^*(-s)} \\ &= \frac{r(s)r^*(-s)}{d(s)d^*(-s)}. \end{aligned} \tag{4.42}$$

We focus on proving (4.42). We will use the following identies:

$$\begin{aligned} n(s) &= \sum_{i=0}^{k/2} \nu_{2i} s^{2i} + \sum_{i=0}^{k/2} \nu_{2i+1} s^{2i+1} = A + B, \\ d(s) &= \sum_{i=0}^{k/2} \delta_{2i} s^{2i} + \sum_{i=0}^{k/2} \delta_{2i+1} s^{2i+1} = C + D. \end{aligned}$$

It is easy to see that

$$\begin{aligned} n(-s) &= \sum_{i=0}^{k/2} \nu_{2i} s^{2i} - \sum_{i=0}^{k/2} \nu_{2i+1} s^{2i+1} = A - B, \\ d(-s) &= \sum_{i=0}^{k/2} \delta_{2i} s^{2i} - \sum_{i=0}^{k/2} \delta_{2i+1} s^{2i+1} = C - D. \end{aligned}$$

For the sum $n(s)d(-s) + n(-s)d(s)$ we then have

$$\begin{aligned} n(s)d(-s) + n(-s)d(s) &= (A+B)(C-D) + (A-B)(C+D) = 2AC - 2BD \\ &= 2\left[\sum_{i=0}^{k/2} \nu_{2i} s^{2i}\right]\left[\sum_{i=0}^{k/2} \delta_{2i} s^{2i}\right] \\ &\quad -2\left[\sum_{i=0}^{k/2} \nu_{2i+1} s^{2i+1}\right]\left[\sum_{i=0}^{k/2} \delta_{2i+1} s^{2i+1}\right] \\ &= \tilde{v}(s) - \tilde{w}(s). \end{aligned}$$

Note that

$$\begin{aligned} \tilde{v}(s) &= \alpha_0 + \alpha_1 s^2 + \alpha_2 s^4 + \cdots + \alpha_k s^{2k}, \\ \tilde{w}(s) &= \beta_1 s^2 + \beta_2 s^4 + \beta_3 s^6 + \cdots + \beta_{k+1} s^{2k+2}. \end{aligned}$$

So, we have

$$t(s) := \tilde{v}(s) - \tilde{w}(s) = \alpha_0 + (\alpha_1 - \beta_1)s^2 + \cdots + (\alpha_k - \beta_k)s^{2k} - \beta_{k+1}s^{2k+2}.$$

Clearly, if s_0 is a zero of $t(s)$, so is $-s_0$. Consequently, we can factorize $t(s)$ as $t(s) = r(s)r(-s)$. Summarizing, we finally have

$$n(s)d(-s) + n(-s)d(s) = \tilde{v}(s) - \tilde{w}(s) = t(s) = r(s)r(-s),$$

which proves (4.42). ■

This last result equals $K(s)K^*(-s)$, i.e., this is the *spectral factorization* of G. Here K is a called the *spectral factor* of G. The zeros of K, i.e. the λ_i, $i = 1, \cdots, n$ such that $det(K(\lambda_i)) = 0$, are the *spectral zeros* of G.

4.3.3 Spectral Zeros and Generalized Eigenvalue Problem

We start this section with explaining a generalized eigenvalue problem, which Sorensen used in [110]. It brings together the theory of positive real interpolation by Antoulas [96] and the invariant subspace method for interpolating the spectral zeros by Sorensen.

First we recall that for the transfer function $\mathbf{G}(s)$ we have

$$\begin{aligned} \mathbf{G}(s) &= \mathbf{C}^*(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}, \quad \text{and thus,} \\ (\mathbf{G}(-s))^* &= \mathbf{B}^*(-s\mathbf{E}^* - \mathbf{A}^*)^{-1}\mathbf{C} + \mathbf{D}^*, \\ &= \mathbf{B}^*(s\mathbf{E}^* - (-\mathbf{A}^*))^{-1}(-\mathbf{C}) + \mathbf{D}^*. \end{aligned}$$

Then we compute $\mathbf{G}+\mathbf{G}^*$,[18]

$$\begin{aligned}\mathbf{G}(s)+(\mathbf{G}(-s))^* = \;& (\mathbf{C}^*(s\mathbf{E}-\mathbf{A})^{-1}\mathbf{B}+\mathbf{D})+(\mathbf{B}^*(s\mathbf{E}^*-(-\mathbf{A}^*))^{-1}(-\mathbf{C})+\mathbf{D}^*)\\ =& \begin{bmatrix}\mathbf{C}^* & \mathbf{B}^*\end{bmatrix}\begin{bmatrix}(s\mathbf{E}-\mathbf{A})^{-1} & 0\\ 0 & (s\mathbf{E}^*-(-\mathbf{A}^*))^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{B}\\ -\mathbf{C}\end{bmatrix}+(\mathbf{D}+\mathbf{D}^*)\\ =& \begin{bmatrix}\mathbf{C}^* & \mathbf{B}^*\end{bmatrix}\left(s\begin{bmatrix}\mathbf{E} & 0\\ 0 & \mathbf{E}^*\end{bmatrix}-\begin{bmatrix}\mathbf{A} & 0\\ 0 & -\mathbf{A}^*\end{bmatrix}\right)^{-1}\begin{bmatrix}\mathbf{B}\\ -\mathbf{C}\end{bmatrix}+(\mathbf{D}+\mathbf{D}^*).\end{aligned}$$

Note that this is the transfer function of the following system:

$$\begin{cases}\begin{bmatrix}\mathbf{E} & 0\\ 0 & \mathbf{E}^*\end{bmatrix}\dot{\mathbf{x}}(t) = \begin{bmatrix}\mathbf{A} & 0\\ 0 & -A^*\end{bmatrix}\mathbf{x}(t)+\begin{bmatrix}\mathbf{B}\\ -C\end{bmatrix}\mathbf{u}(t)\\ \\ \mathbf{y}(t) \qquad\quad = \begin{bmatrix}\mathbf{C} & \mathbf{B}\end{bmatrix}^*\mathbf{x}(t)+(\mathbf{D}+\mathbf{D}^*)u(t)\end{cases} \tag{4.43}$$

Let

$$\mathcal{A}=\begin{bmatrix}\mathbf{A} & \mathbf{0} & \mathbf{B}\\ \mathbf{0} & -\mathbf{A}^* & -\mathbf{C}\\ \mathbf{C}^* & \mathbf{B}^* & \mathbf{D}+\mathbf{D}^*\end{bmatrix} \quad\text{and}\quad \mathcal{E}=\begin{bmatrix}\mathbf{E} & & \\ & \mathbf{E}^* & \\ & & \mathbf{0}\end{bmatrix}.$$

The finite *spectral zeros* of G are the set of all finite complex numbers λ such that

$$\text{Rank}(\mathcal{A}-\lambda\mathcal{E}) < 2n+p,$$

i.e., the finite generalized eigenvalues $\sigma(\mathcal{A},\mathcal{E})$. The set of spectral zeros is denoted as $\mathcal{S}_G$.

Lemma 4.1 *If λ is a generalized eigenvalue $\sigma(\mathcal{A}-\lambda\mathcal{E})$ in $\mathcal{S}_G$ then $-\bar{\lambda}$ also belongs to $\mathcal{S}_G$, i.e.,*

$$\lambda\in\mathcal{S}_G \Rightarrow -\bar{\lambda}\in\mathcal{S}_G \quad \textit{since} \quad \mathcal{A}\mathbf{q}=\lambda\mathcal{E}\mathbf{q}\Rightarrow \tilde{\mathbf{q}}^*\mathcal{A}=(-\bar{\lambda})\tilde{\mathbf{q}}^*\mathcal{E},$$

where $\mathbf{q}^*=[\mathbf{x}^*,\mathbf{y}^*,\mathbf{z}^*]$ is a right eigenvector and $\tilde{\mathbf{q}}^*=[\mathbf{y}^*,-\mathbf{x}^*,\mathbf{z}^*]$. Also

$$\lambda\in\mathcal{S}_G \Rightarrow -\bar{\lambda}\in\mathcal{S}_G \quad \textit{since} \quad \mathbf{r}\mathcal{A}=\lambda\mathbf{r}\mathcal{E}\Rightarrow \mathcal{A}\tilde{\mathbf{r}}^*=(-\bar{\lambda})\mathcal{E}\tilde{\mathbf{r}}^*,$$

where $\mathbf{r}^*=[\mathbf{x1}^*,\mathbf{y1}^*,\mathbf{z1}^*]$ is a left eigenvector and $\tilde{\mathbf{r}}^*=[-\mathbf{y1}^*,\mathbf{x1}^*,\mathbf{z1}^*]$.

[18] Block wise inversion:

$$\begin{bmatrix}A & B\\ C & D\end{bmatrix}^{-1}=\begin{bmatrix}A^{-1}+A^{-1}B(D-CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D-CA^{-1}B)^{-1}\\ -(D-CA^{-1}B)^{-1}CA^{-1} & (D-CA^{-1}B)^{-1}\end{bmatrix}$$

Proof If $\lambda \in \sigma(\mathcal{A} - \lambda\mathcal{E})$ and $\mathbf{q}$ is the corresponding eigenvector then

$$\mathcal{A}\mathbf{q} = \lambda\mathcal{E}\mathbf{q}$$

or

$$\begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & -\mathbf{A}^* & -\mathbf{C} \\ \mathbf{C}^* & \mathbf{B}^* & \mathbf{D}+\mathbf{D}^* \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{E} & & \\ & \mathbf{E}^* & \\ & & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix}$$

By taking conjugates and changing rows one obtains

$$\begin{bmatrix} \mathbf{y}^* & -\mathbf{x}^* & \mathbf{z}^* \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & -\mathbf{A}^* & -\mathbf{C} \\ \mathbf{C}^* & \mathbf{B}^* & \mathbf{D}+\mathbf{D}^* \end{bmatrix} = -\bar{\lambda} \begin{bmatrix} \mathbf{y}^* & -\mathbf{x}^* & \mathbf{z}^* \end{bmatrix} \begin{bmatrix} \mathbf{E} & & \\ & \mathbf{E}^* & \\ & & \mathbf{0} \end{bmatrix}, \quad \text{or}$$

$$\tilde{\mathbf{q}}^*\mathcal{A} = -\bar{\lambda}\tilde{\mathbf{q}}^*\mathcal{E}.$$

Now we can conclude that $-\bar{\lambda} \in \mathcal{S}_G$ and that $\tilde{\mathbf{q}}^*$ is its corresponding eigenvector. The proof is similar for the left eigenvectors [110]. ■

If specified spectral zeros are preserved (interpolated) in the reduced model with $\mathcal{S}_{\hat{G}}$ then a passive reduced model will result. For real systems, $\mathcal{S}_{\hat{G}}$ must include conjugate pairs of spectral zeros. This result is based on Antoulas' theorem [96]:

Theorem 4.3 (Antoulas) *Suppose $\mathcal{S}_{\hat{G}} \subset \mathcal{S}_G$ and also that $\hat{G}(\lambda) = G(\lambda)$ for all $\lambda \in \mathcal{S}_{\hat{G}}$ and that $\hat{G}$ is a minimal degree rational interpolant of the values of G on the set $\mathcal{S}_{\hat{G}}$. Then the reduced system $\hat{\sum}$ with transfer function $\hat{G}$ is both stable and passive.*

4.3.4 Passivity Preserving Model Reduction

Theorem 4.3 indicates that Antoulas's approach [96] *preserves passivity for the reduced model when spectral zero interpolation is applied*. The interpolation is guaranteed by building the projection matrices using a Krylov subspace method [103, 111]. Antoulas' method [96] significantly differs from PRIMA [107]. For a detailed comparison between PRIMA and Antoulas's approach we refer to [105].

In Antoulas' method it is assumed that the system Σ with transfer function $\mathbf{G}(s)$ is passive. Then one defines a set $\mathcal{S}_1 \subset \mathcal{S}_{\mathbf{G}_{stable}}$ where $\mathcal{S}_{\mathbf{G}_{stable}}$ is the set of stable *spectral zeros* and one takes $\mathcal{S}_2 = -\mathcal{S}_1$. Antoulas [96] has shown that the reduced system $\hat{\Sigma}$ with transfer function $\hat{\mathbf{G}}(s)$ is passive if the set of interpolation points is $\mathcal{S}_1 \cup \mathcal{S}_2$.

A second approach has been introduced by Sorensen [110], which can be seen as an interpolatory model reduction too. It is based on invariant subspaces. In this method it is not necessary that the spectral zeros (interpolation points) are computed in advance. Sorensen's approach transfers the model reduction problem into an eigenvalue problem. In this case the eigenvalues are the spectral zeros of the transfer function of the original system. Then the projection matrices are built from a basis for a chosen invariant subspace.

Choosing different spectral zeros gives us different invariant subspaces, which return different reduced models. Although these reduced models are passive, they may not be a good approximation to the original system. So the selection of spectral zeros must guarantee that the reduced model is a good approximation to the original ones.

In large scale problems in which the eigen computation of the resulting highly-structured eigenvalue problem should be done iteratively, all selection criteria can not be satisfied. So the problem has two goals: the first one is to have a good approximation of the original model, the second one is to be suitable as an iterative scheme for large-scale dynamical systems.

4.3.5 Model Reduction by Projection

We will construct a basis for a selected invariant subspace of the pair $(\mathcal{A}, \mathcal{E})$ (Sorensen [110]). Let

$$\mathcal{A}\mathbf{Q} = \mathcal{E}\mathbf{Q}\mathbf{R}$$

be a partial real Schur decomposition for the pair $(\mathcal{A}, \mathcal{E})$. Then, $\mathbf{Q}^*\mathbf{Q} = I$ and $\mathbf{R}$ is real and quasi-upper triangular. Let $Q = [\mathbf{X}^*, \mathbf{Y}^*, \mathbf{Z}^*]^*$ be partitioned in accordance with the block structure of $\mathcal{A}$:

$$\begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & -\mathbf{A}^* & -\mathbf{C} \\ \mathbf{C}^* & \mathbf{B}^* & \mathbf{D}+\mathbf{D}^* \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{bmatrix} = \begin{bmatrix} \mathbf{E} & & \\ & \mathbf{E}^* & \\ & & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{bmatrix} \mathbf{R}$$

$$\Rightarrow \begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & -\mathbf{A}^* & -\mathbf{C} \\ \mathbf{C}^* & \mathbf{B}^* & \mathbf{D}+\mathbf{D}^* \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{bmatrix} = \begin{bmatrix} \mathbf{E}\mathbf{X} \\ \mathbf{E}^*\mathbf{Y} \\ \mathbf{0} \end{bmatrix} \mathbf{R} \tag{4.44}$$

The projection will be constructed from $\mathbf{X}$ and $\mathbf{Y}$ and the reduced model will be obtained out of these. Here it will be useful to have the following lemma [110].

Lemma 4.2 *Suppose that* $\mathbf{R}$ *in* (4.44) *satisfies* $Re(\lambda) > 0, \ \forall \lambda \in \sigma(\mathbf{R})$. *Then* $\mathbf{X}^*\mathbf{E}^*\mathbf{Y} = \mathbf{Y}^*\mathbf{E}\mathbf{X}$ *is symmetric.*

Proof We start with

$$\mathcal{A}\mathbf{Q} = \mathcal{E}\mathbf{QR}. \tag{4.45}$$

By (4.45) and according to the previous proof we have

$$\hat{\mathbf{Q}}^*\mathcal{A} = (-\mathbf{R}^*)\hat{\mathbf{Q}}^*\mathcal{E} \quad \text{where} \quad \hat{\mathbf{Q}}^* = \begin{bmatrix} \mathbf{Y}^* & -\mathbf{X}^* & \mathbf{Z}^* \end{bmatrix}, \tag{4.46}$$

If we multiply equation (4.45) with $\hat{\mathbf{Q}}^*$ from the left, then we get

$$\hat{\mathbf{Q}}^*\mathcal{A}\mathbf{Q} = \hat{\mathbf{Q}}^*\mathcal{E}\mathbf{QR}. \tag{4.47}$$

We substitute the right part of equation (4.46) in the left part of equation (4.47), giving

$$\begin{aligned} (-\mathbf{R}^*)\hat{\mathbf{Q}}^*\mathcal{E}\mathbf{Q} &= \hat{\mathbf{Q}}^*\mathcal{E}\mathbf{QR} \\ \Rightarrow \quad \mathbf{R}^*\hat{\mathbf{Q}}^*\mathcal{E}\mathbf{Q} + \hat{\mathbf{Q}}^*\mathcal{E}\mathbf{QR} &= \mathbf{0}. \end{aligned} \tag{4.48}$$

Here

$$\begin{aligned} \hat{\mathbf{Q}}^*\mathcal{E}\mathbf{Q} &= \begin{bmatrix} \mathbf{Y}^* & -\mathbf{X}^* & \mathbf{Z}^* \end{bmatrix} \begin{bmatrix} \mathbf{E} & & \\ & \mathbf{E}^* & \\ & & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{bmatrix} \\ &= \mathbf{Y}^*\mathbf{EX} - \mathbf{X}^*\mathbf{E}^*\mathbf{Y}. \end{aligned} \tag{4.49}$$

If we substitute (4.49) in (4.48) we obtain

$$\mathbf{R}^*(\mathbf{Y}^*\mathbf{EX} - \mathbf{X}^*\mathbf{E}^*\mathbf{Y}) + (\mathbf{Y}^*\mathbf{EX} - \mathbf{X}^*\mathbf{E}^*\mathbf{Y})\mathbf{R} = \mathbf{0}. \tag{4.50}$$

Therefore the equation (4.50) has the unique solution[19]:

$$\mathbf{Y}^*\mathbf{EX} - \mathbf{X}^*\mathbf{E}^*\mathbf{Y} = \mathbf{0},$$

and hence

$$\mathbf{Y}^*\mathbf{EX} = \mathbf{X}^*\mathbf{E}^*\mathbf{Y},$$

which completes the proof. ■

[19] Equation (4.50) is a simple form ($R^*X + XR = 0$) of a Lyapunov equation of the more general type $AX - XB = C$ (which has a unique solution if $\sigma(A) \cap \sigma(B) = \oslash$). Due to the condition $Re(\lambda) > 0$ for λ in $\sigma(R)$, we have that $\sigma(R^*) \cap \sigma(-R) = \oslash$. Hence the Lyapunov equation (4.50) has a unique (zero) solution.

For the construction of $\mathbf{V}$ and $\mathbf{W}$ as projections, we first have to find a basis for an invariant subspace [102] with all eigenvalues of $\mathbf{R}$ in the right half-plane.

Let $\mathbf{Q}_x\mathbf{S}^2\mathbf{Q}_y^* = \mathbf{X}^*\mathbf{Y}$ be the SVD of $\mathbf{X}^*\mathbf{Y}$ and note that $\mathbf{Q}_y = \mathbf{Q}_x\mathbf{J}$ where $\mathbf{J}$ is a signature matrix by virtue of the fact that $\mathbf{X}^*\mathbf{Y}$ is symmetric.

If $\mathbf{S} \geq \mathbf{0}$ is nonsingular, put

$$\begin{aligned} \mathbf{V} &= \mathbf{X}\mathbf{Q}_x\mathbf{S}^{-1} \\ \mathbf{W} &= \mathbf{Y}\mathbf{Q}_y\mathbf{S}^{-1}. \end{aligned} \tag{4.51}$$

It follows that

$$\begin{aligned} \mathbf{W}^*\mathbf{V} &= (\mathbf{Y}\mathbf{Q}_y\mathbf{S}^{-1})^*\mathbf{X}\mathbf{Q}_x\mathbf{S}^{-1} \\ &= \mathbf{S}^{-*}\mathbf{Q}_y^*\mathbf{Y}^*\mathbf{X}\mathbf{Q}_x\mathbf{S}^{-1} \\ \text{(include the SVD form of } \mathbf{X}^*\mathbf{Y}) &= \mathbf{S}^{-*}\mathbf{Q}_y^*\mathbf{Q}_y(\mathbf{S}^2)^*\mathbf{Q}_x^*\mathbf{Q}_x\mathbf{S}^{-1} \\ (\mathbf{Q}_x \text{ and } \mathbf{Q}_y \text{ are unitary matrices}) &= \mathbf{S}^{-*}\mathbf{S}^*\mathbf{S}^*\mathbf{S}^{-1} \\ &= \mathbf{I}. \end{aligned}$$

and also we have

$$\mathbf{V}^*\mathbf{W} = (\mathbf{W}^*\mathbf{V})^* = \mathbf{I}.$$

Now from the SVD of $\mathbf{X}^*\mathbf{Y}$, let

$$\begin{aligned} \hat{\mathbf{X}} &= \mathbf{S}(\mathbf{Q}_x)^* \\ \hat{\mathbf{Y}} &= \mathbf{S}(\mathbf{Q}_y)^*, \end{aligned}$$

and define

$$\mathcal{V} = \begin{bmatrix} \mathbf{V} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \quad \text{and} \quad \mathcal{W} = \begin{bmatrix} \mathbf{W} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{V} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}.$$

It is obvious that $\mathcal{W}^*\mathcal{V} = \mathbf{I}$ and that

$$\begin{aligned} \mathbf{V}\hat{\mathbf{X}} &= (\mathbf{X}\mathbf{Q}_x\mathbf{S}^{-1})(\mathbf{S}\mathbf{Q}_x^*), \\ &= \mathbf{X}\mathbf{Q}_x\mathbf{Q}_x^*, \\ (\mathbf{Q}_x^* \quad \text{is unitary matrix}) &= \mathbf{X}. \end{aligned}$$

Similarly, $\mathbf{W}\hat{\mathbf{Y}} = \mathbf{Y}$, so we have

$$\begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{bmatrix} = \begin{bmatrix} \mathbf{V} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{X}} \\ \hat{\mathbf{Y}} \\ \hat{\mathbf{Z}} \end{bmatrix}.$$

Therefore

$$\hat{\mathcal{A}} = \mathcal{W}^*\mathcal{A}\mathcal{V} = \begin{bmatrix} \hat{\mathbf{A}} & \mathbf{0} & \hat{\mathbf{B}} \\ \mathbf{0} & -\hat{\mathbf{A}}^* & -\hat{\mathbf{C}} \\ \hat{\mathbf{C}}^* & \hat{\mathbf{B}}^* & \mathbf{D}+\mathbf{D}^* \end{bmatrix} \quad \text{and} \quad \hat{\mathcal{E}} = \mathcal{W}^*\mathcal{E}\mathcal{V} = \begin{bmatrix} \hat{\mathbf{E}} & & \\ & \hat{\mathbf{E}}^* & \\ & & \mathbf{0} \end{bmatrix},$$

and

$$\begin{bmatrix} \hat{\mathbf{A}} & \mathbf{0} & \hat{\mathbf{B}} \\ \mathbf{0} & -\hat{\mathbf{A}}^* & -\hat{\mathbf{C}} \\ \hat{\mathbf{C}}^* & \hat{\mathbf{B}}^* & \mathbf{D}+\mathbf{D}^* \end{bmatrix} \begin{bmatrix} \hat{\mathbf{X}} \\ \hat{\mathbf{Y}} \\ \hat{\mathbf{Z}} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{E}} & & \\ & \hat{\mathbf{E}}^* & \\ & & \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{X}} \\ \hat{\mathbf{Y}} \\ \hat{\mathbf{Z}} \end{bmatrix} \mathbf{R},$$

or

$$\begin{bmatrix} \hat{\mathbf{A}} & \mathbf{0} & \hat{\mathbf{B}} \\ \mathbf{0} & -\hat{\mathbf{A}}^* & -\hat{\mathbf{C}} \\ \hat{\mathbf{C}}^* & \hat{\mathbf{B}}^* & \mathbf{D}+\mathbf{D}^* \end{bmatrix} \begin{bmatrix} \hat{\mathbf{X}} \\ \hat{\mathbf{Y}} \\ \hat{\mathbf{Z}} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{E}}\hat{\mathbf{X}} \\ \hat{\mathbf{E}}^*\hat{\mathbf{Y}} \\ \mathbf{0} \end{bmatrix} \mathbf{R}.$$

where $\hat{\mathbf{A}} = \mathbf{W}^*\mathbf{A}\mathbf{V}$, $\hat{\mathbf{E}} = \mathbf{W}^*\mathbf{E}\mathbf{V}$, $\hat{\mathbf{B}} = \mathbf{W}^*\mathbf{B}$, and $\hat{\mathbf{C}} = \mathbf{V}^*\mathbf{C}$.
This shows that $\mathcal{S}_{\hat{G}} \subseteq \mathcal{S}_G$ and since $\mathcal{S}_{\hat{G}} = \sigma(\mathbf{R}) \cup \sigma(-\mathbf{R}^*)$[20] and $\sigma(\mathbf{R})$ is in the open right half-plane, the reduced model has no spectral zeros on the imaginary axis.

The previous result is also valid when $\mathbf{S}$ is nonsingular. Now we consider the case $\mathbf{S}$ is singular. Beginning with $\mathbf{X}, \mathbf{Y}$ from (4.44) and with the SVD of $\mathbf{X}^*\mathbf{Y}$,

[20] We know that if we have a real matrix $\mathbf{A}$ and $\lambda \in \sigma(\mathbf{A})$ then $\bar{\lambda} \in \sigma(\mathbf{A})$. In Lemma 4.1 we showed that if $\lambda \in \mathcal{S}_G$ then $-\bar{\lambda} \in \mathcal{S}_G$. Therefore

$$\lambda,\ \bar{\lambda},\ -\lambda \quad \text{and} -\bar{\lambda} \in \mathcal{S}_G.$$

On the other hand, $\mathbf{R}$ is a selected invariant subspace of $(\mathcal{A}, \mathcal{E})$, which means that $\sigma(\mathbf{R}) \subset \mathcal{S}_G$. Now, we need to find a basis for an invariant subspace with eigenvalues of $\mathbf{R}$ in the open right half-plane. As we mentioned above $\sigma(\mathbf{R})$ and $\sigma(-\mathbf{R}^*)$ are a subset of $\mathcal{S}_G$. Thus take

$$\mathcal{S}_{\hat{G}} = \sigma(\mathbf{R}) \cup \sigma(-\mathbf{R}^*).$$

Algorithm 4.7 Sorensen's Algorithm [110]

```
INPUT: System $(E, A, B, C, D)$,
OUTPUT: Reduced System $(\hat{E}, \hat{A}, \hat{B}, \hat{C}, D)$
Compute $\mathcal{A}, \mathcal{E}$
$[\mathcal{A}_1, \mathcal{E}_1, Z, Q, V, W] = \mathtt{qz}(\mathcal{A}, \mathcal{E})$;
Find spectral zeros, $\Lambda = \mathtt{eig}(\mathcal{A}, \mathcal{E})$;
Find the real basis for the right eigenvector matrix $V$,
Find the positive real spectral zeros and corresponding eingenvectors, $\Lambda_1 = [\,]$; $V_1 = [\,]$;
for $i = 1 : \mathtt{length}(\Lambda)$ do
    if $(\mathtt{real}(\Lambda(i)) > 0$ and $\Lambda(i)$ are chosen spectral zeros) or $\mathtt{imag}\Lambda(i) = 0$ then
        $\Lambda_1 = [\Lambda_1 \quad \Lambda(i)]$; $V_1 = [V_1 \quad V(:, i)]$;
    end if
end for
$X = V_1(1 : n, :)$; $Y = V_1(n + 1 : 2n, :)$;
$[Q_x, S^2, Q_y] = \mathtt{svd}(X^*Y)$;
Construct the projection matrices, $V = XQ_x S^{-1}$; $W = YQ_y S{-}1$;
$\hat{E} = W^*EV$; $\hat{A} = W^*AV$; $\hat{B} = W^*B$; $\hat{C} = CV$;
```

where $\mathbf{Q}_x\mathbf{S}^2\mathbf{Q}_y = \mathbf{X}^*\mathbf{Y}$, specify a cut-off tolerance $\tau_c \in (0, 1)$ and let j be the largest positive integer such that

$$\sigma_j \geq \tau_c\sigma_1 \qquad \text{where} \quad \sigma_j = \mathbf{S}(j, j).$$

Define $\mathbf{Q}_j = \mathbf{Q}_x(:, 1 : j)$, $\mathbf{S}_j = \mathbf{S}(\mathbf{1} : \mathbf{j}, \mathbf{1} : \mathbf{j})$ and then let $(\mathbf{X}_j)^I = \mathbf{Q}_j(\mathbf{S}_j)^{-1}$. Replace $\hat{\mathbf{X}}^{-1} = (\mathbf{X}_j)^I$, $\mathbf{V} = \mathbf{X}(\mathbf{X}_j)^I$ and $\mathbf{W} = -\mathbf{Y}(\mathbf{X}_j)^I$. According to [110], in this way, the reduced system is passive and also the stability of the reduced model is obtained if $\mathbf{Z}$ is full rank.

Sorenson's Algorithm is described in Algorithm 4.7.

4.3.6 Model Reduction by Projection

We want to reduce the original system $\sum$ to $\hat{\sum}$ where the complexity k of $\hat{\sum}$ is (much) less than that of $\sum$ $(k \ll n)$ (Antoulas [96]). This reduction must preserve both stability and passivity and it must be numerically efficient. Antoulas' Algorithm is described in Algorithm 4.8.

We will look for $\mathbf{V}, \mathbf{W} \in \mathbb{R}^{n\times k}$ such that $\mathbf{V}\mathbf{W}^*$ is a projection with the additional condition $\mathbf{W}^*\mathbf{V} = \mathbf{I}_k$ (recall that P is a projection matrix if $P^2 = P$). So, if we have $\mathbf{V}$ and $\mathbf{W}$ with $\mathbf{W}^*\mathbf{V} = \mathbf{I}_k$, then indeed

$$(\mathbf{V}\mathbf{W}^*)^2 = \mathbf{V}\mathbf{W}^*.$$

Algorithm 4.8 Antoulas's Algorithm [96]

```
INPUT: System $(E, A, B, C, D)$,
OUTPUT: Reduced System $(\hat{E}, \hat{A}, \hat{B}, \hat{C}, D)$
Compute $\mathcal{A}, \mathcal{E}$
Find spectral zeros, $\Lambda = \texttt{eig}(\mathcal{A}, \mathcal{E})$;
$\Lambda R = [\,]$; $\Lambda C = [\,]$;
while $n \geq \texttt{length}(\Lambda)$ do
    if $\Lambda(n)$ is positive real, $\Lambda R = [\Lambda R \quad \Lambda(n)]$;
    if $\Lambda(n)$ is complex and in right half-plane, $\Lambda C = [\Lambda C \quad \Lambda(n)]$;
end while
for $m = 1 : \texttt{length}(\Lambda C)$ do
    if $\Lambda C(m)$ chosen spectral zeros then
        $\Lambda R = [\Lambda R \quad \Lambda C(m)]$;
    end if
end for
$V = [\,]$; $W = [\,]$;
for $q = 1 : \texttt{length}(\Lambda R)$ do
    $v = (\Lambda R(q)E - A)^{-1}B$; $w = (-\Lambda R(q)E^* - A^*)^{-1}C^*$;
    $V = [V \quad v]$; $W = [W \quad w]$;
end for
Make a real basis for $V$ and $W$
$W = (W^*V)^{-1}W$;
$\hat{E} = W^*EV$; $\hat{A} = W^*AV$; $\hat{B} = W^*B$; $\hat{C} = CV$;
```

Given $2k$ distinct points $s_1, \cdots, s_{2k}$, let

$$\tilde{\mathbf{V}} = \left[\, (s_1\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{B} \cdots (s_k\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{B} \,\right],$$
$$\tilde{\mathbf{W}} = \left[\, (s_{k+1}\mathbf{I}_n - \mathbf{A}^*)^{-1}\mathbf{C} \cdots (s_{2k}\mathbf{I}_n - \mathbf{A}^*)^{-1}\mathbf{C} \,\right]. \tag{4.52}$$

Now take $\mathbf{V} = \tilde{\mathbf{V}}$ and $\mathbf{W} = \tilde{\mathbf{W}}(\tilde{\mathbf{V}}^*\tilde{\mathbf{W}})^{-1}$. We define

$$\hat{\mathbf{A}} = \mathbf{W}^*\mathbf{A}\mathbf{V}, \quad \hat{\mathbf{B}} = \mathbf{W}^*\mathbf{B}, \quad \hat{\mathbf{C}} = \mathbf{V}^*\mathbf{C}. \tag{4.53}$$

Then we have the following theorem (Antoulas [96])

Proposition 4.1 *Assuming that det*$(\tilde{\mathbf{W}}^*\tilde{\mathbf{V}}) \neq \mathbf{0}$*, the projected system* $\hat{\sum}$*, defined by* (4.53)*, interpolates the transfer function of* $\sum$ *at the points* s_i*:*

$$\hat{\mathbf{G}}(s_i) = \mathbf{G}(s_i) \qquad i = 1, 2, \cdots, 2k.$$

where s_i *are the spectral zeros.*

4.3.7 Numerical Results

In [108] several numerical results are presented for an RLC-circuit that is also found in [96, 110]. The transfer function is a scalar function $G(s)$. The starting point is to compute the spectral zeros (using a generalized eigenvalue method) and then to try to categorize them related to their magnitude, like distance from the real and the imaginary axis in order to have a good match in low or high frequency. The reduced method was obtained by Algorithm 4.8 of Antoulas. A large distance from the real axis results in a good approximation at high frequencies. A large distance from the imaginary axis results in a good approximation at low frequencies. In both situations including the real spectral zeros plays an important role for having a good reduced model at low frequencies.

One should check if a spectral zero also occurs as a pole and as a zero, both, in which case the factors $(\lambda\mathbf{I} - \mathbf{A})$ are singular. These spectra zeros should be left out of the reduction.

In this section we study a circuit which has a descriptor matrix $\mathbf{E} \neq \mathbf{I}_n$. We consider the circuit shown in Fig. 4.16. We assume that all capacitors and inductors have a unit value, $R_1 = \frac{1}{2}\Omega$, $R_2 = \frac{1}{5}\Omega$, $R_{2k} = \frac{1}{3}\Omega$, where $k = 2, 3, \cdots, n$ and $R_{2k+1} = \frac{1}{4}\Omega$, where $k = 1, 2, \cdots, n$.

The order of the original system is 1003 and the selected spectral zeros close to the real axis are shown in Fig. 4.17. In this case, like before, the reduced model has a good match at low and at high frequencies, as shown in Fig. 4.18.

4.3.8 Conclusion

We have considered two approaches for passive and stable reduction of dynamical systems in circuit simulation, based on the methods by Antoulas [96] and Sorenson [110] that both exploit interpolating spectral zeros. The reduced models preserve passivity and stability. The original system is reduced by projection matrices, which are built via spectral zero interpolation. Different selections of spectral zeros give us different approximations of the original model, which may/may not produce

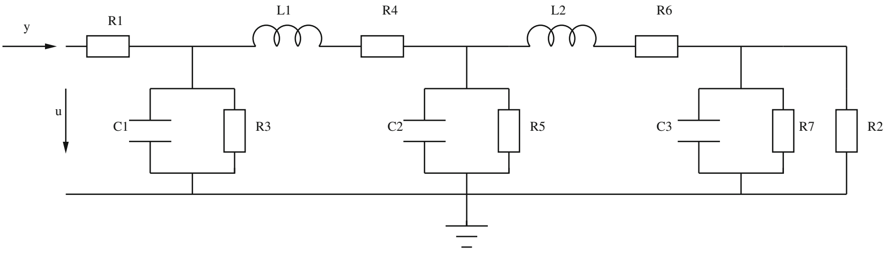

Fig. 4.16 RLC Circuit of Order 7

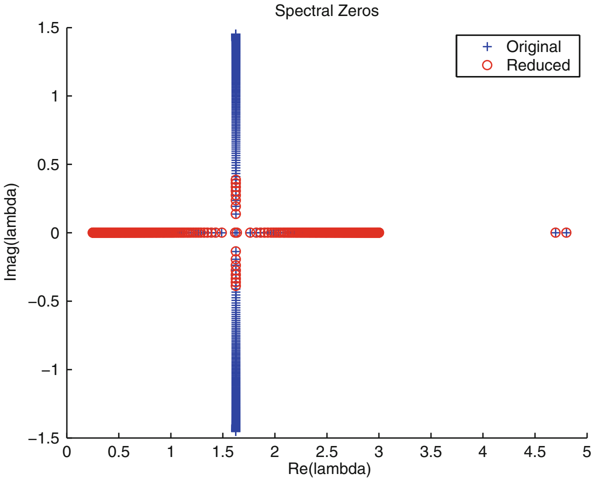

Fig. 4.17 Spectral zeros of the original model (+), and spectral zeros of the reduced model (o). For interpolation, the spectral zeros close to the real axis are chosen. All selected spectral zeros are preserved after reduction. The order of the original model is 1003 and it is reduced to 341

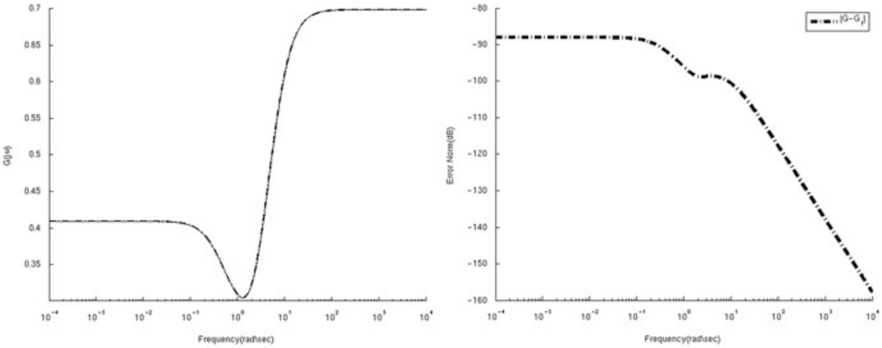

Fig. 4.18 Effect of several real spectral zeros, *Left*: Frequency responses of the original system and reduced model. The spectral zeros close to the real axis are interpolated. *Right*: Frequency response of the error $\|\Sigma - \hat{\Sigma}\|_2$

acceptable reduction. We have considered criteria for selecting the spectral zeros and also to approximate the original system well in low and high frequency. When the spectral zeros are chosen close to the real axis, the reduced model matches the original response well for low frequencies. On the other hand, when they are far from the real axis, the reduced model is more accurate for high frequencies. As already shown preserving the real spectral zero plays an important role for having a good reduction in the whole frequency domain, specially in low frequency. It means that one should try to save all the real spectral zeros of the system.

The approaches of Antoulas and Sorensen are equivalent but as Sorensen's algorithm works directly with eigenvalues and eigenvectors, it is more usable for constructing the projection matrices. For the same reason Sorensen's approach is more suitable for large scale systems.

4.4 Passivity Preserving Model Reduction Using the Dominant Spectral Zero Method

The design of integrated circuits has become increasingly complex, thus electromagnetic couplings between components on a chip are no longer negligible.[21] To verify coupling effects, on-chip interconnections are modeled as RLC circuits and simulated. As these circuits contain millions of electrical components, the underlying dynamical systems have millions of internal variables and cannot be simulated in full dimension. Model order reduction (MOR) aims at approximating the mathematical description of a large scale circuit with a model of smaller dimension, which replaces the original model during verification and speeds up simulation. The reduction method should preserve important properties of the original model (i.e., stability, passivity) and have an efficient, robust implementation, suitable for large-scale applications. RLC circuits describing the interconnect are *passive* systems, with *positive real* transfer functions [113, 116], thus reduced models should also be passive. A passive reduced model can be synthesized back into an RLC circuit [113], which is placed instead of the original in the simulation flow. Passive reduced circuits also guarantee stable simulations when integrated with the overall nonlinear macro-model [117, 128, 133] during later simulation stages.

The proposed *Dominant Spectral Zero Method (dominant SZM)* is a model reduction method which preserves passivity and stability, and is efficiently implemented using *the subspace accelerated dominant pole algorithm (SADPA)* [130, 131]. Passivity preservation is ensured via a new approach, that of interpolation at *dominant spectral zeros*, a subset of spectral zeros of the original model. Dominant SZM reduces automatically all passive systems, including those with formulations unsuitable for PRIMA (first order susceptance-based models for inductive couplings (RCS circuits) [140] or models involving controlled sources, such as vector potential equivalent circuit (VPEC) [139] and partial element equivalent circuit (PEEC) models [136]). In comparison to *positive real balanced truncation (PRBT)* [129], dominant SZM efficiently handles systems with a possibly singular $\mathbf{E}$ matrix [see (4.54)]. Unlike *modal approximation (MA)* [131, 135] where interpolation is at dominant poles, our method matches the dominant spectral zeros of the original system, guaranteeing passivity.

[21] Section 4.4 has been written by: Roxana Ionutiu, Joost Rommes and Athanasios C. Antoulas. For an extended treatment on the topics of this Section see also the Ph.D. Thesis of the first author [121].

The remainder of this section is structured as follows. The introduction continues with the mathematical setup of MOR in Sect. 4.4.1, and with a brief description of MOR via spectral zero interpolation in Sect. 4.4.2. Dominant SZM is presented concisely in Sect. 4.4.3.1 (following [125]). It is extended with the concept of dominance at ∞ (Sect. 4.4.3.2), and with an approach for converting the reduced models to circuit representations (Sect. 4.4.3.3). Numerical results follow in Sect. 4.4.4 and the section concludes with Sect. 4.4.5. Algorithmic pseudocode for the dominant SZM – SADPA implementation is given in the Appendix 4.4.6.

4.4.1 Background on MOR

The model reduction framework involves approximation of an original dynamical system described by a set of differential algebraic equations in the form:

$$\mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t),\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \tag{4.54}$$

where the entries of $\mathbf{x}(t)$ are the system's internal variables, $\mathbf{u}(t)$ is the system input and $\mathbf{y}(t)$ is the system output, with dimensions $\mathbf{x}(t) \in \mathbb{R}^n$, $\mathbf{u}(t) \in \mathbb{R}^m$, $\mathbf{y}(t) \in \mathbb{R}^p$. Correspondingly, $\mathbf{E} \in \mathbb{R}^{n\times n}$, $\mathbf{A} \in \mathbb{R}^{n\times n}$, $(\mathbf{A}, \mathbf{E})$ is a regular pencil, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{p\times n}$, $\mathbf{D} \in \mathbb{R}^{p\times m}$. The original system $\Sigma(\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ is stable and passive and has *dimension* n, usually very large. We seek a reduced order model $\hat{\Sigma}(\hat{\mathbf{E}}, \hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}, \mathbf{D})$, which satisfies: $\hat{\mathbf{E}}\dot{\hat{\mathbf{x}}}(t) = \hat{\mathbf{A}}\hat{\mathbf{x}}(t) + \hat{\mathbf{B}}\mathbf{u}(t)$, $\hat{\mathbf{y}}(t) = \hat{\mathbf{C}}\hat{\mathbf{x}}(t) + \mathbf{D}\mathbf{u}(t)$, where $\hat{\mathbf{x}} \in \mathbb{R}^k$, $\hat{\mathbf{E}} \in \mathbb{R}^{k\times k}$, $\hat{\mathbf{A}} \in \mathbb{R}^{k\times k}$, $\hat{\mathbf{B}} \in \mathbb{R}^{k\times m}$, $\hat{\mathbf{C}} \in \mathbb{R}^{p\times k}$, $\mathbf{D} \in \mathbb{R}^{p\times m}$. $\hat{\Sigma}$ is obtained by projecting the internal variables of the original system $\mathbf{x}$ onto a subspace $ColSpan(\mathbf{V}) \subset \mathbb{R}^{n\times k}$, along $Null(\mathbf{W}^*) \subset \mathbb{R}^{k\times n}$. The goal is to construct $\mathbf{V}$ and $\mathbf{W}$, such that $\hat{\Sigma}$ is stable and passive. Additionally, $\mathbf{V}$ and $\mathbf{W}$ should be computed efficiently. The reduced matrices are obtained as follows:

$$\hat{\mathbf{E}} = \mathbf{W}^*\mathbf{E}\mathbf{V},\ \hat{\mathbf{A}} = \mathbf{W}^*\mathbf{A}\mathbf{V},\ \hat{\mathbf{B}} = \mathbf{W}^*\mathbf{B},\ \hat{\mathbf{C}} = \mathbf{C}\mathbf{V}. \tag{4.55}$$

4.4.2 MOR by Spectral Zero Interpolation

We revise the spectral zero interpolation approach for model reduction as proposed in [114, 134]. The ingredient for passivity preservation are the *spectral zeros* of $\Sigma(\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$, defined as follows:

Definition 4.5 For system Σ with transfer function: $\mathbf{H}(s) := \mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$, the spectral zeros are all $s \in \mathbb{C}$ such that $\mathbf{H}(s) + \mathbf{H}^*(-s) = 0$, where $\mathbf{H}^*(-s) = \mathbf{B}^*(-s\mathbf{E}^* - \mathbf{A}^*)^{-1}\mathbf{C}^* + \mathbf{D}^*$.

According to [114, 134], model reduction via spectral zero interpolation involves forming rational Krylov subspaces:

$$\begin{aligned}\mathbf{V} &= [(s_1\mathbf{E}-\mathbf{A})^{-1}\mathbf{B},\ \cdots,\ (s_k\mathbf{E}-\mathbf{A})^{-1}\mathbf{B}],\\ \mathbf{W} &= [(-s_1^*\mathbf{E}^*-\mathbf{A}^*)^{-1}\mathbf{C}^*,\ \cdots,\ (-s_k^*\mathbf{E}^*-\mathbf{A}^*)^{-1}\mathbf{C}^*],\end{aligned} \tag{4.56}$$

where $s_1 \dots s_k, -s_1^* \dots -s_k^*$ are a subset of the spectral zeros of Σ. By projecting the original system with matrices (4.56) according to (4.55), the reduced $\hat{\Sigma}$ interpolates Σ at the chosen s_i and their mirror images $-s_i^*$, $i = 1, \dots, k$ [113, 114]. Projection matrices $\mathbf{V}$ and $\mathbf{W}$ insure that the reduced system satisfies the positive real lemma [113, 114, 116, 134], thus passivity is preserved. If in the original system $\mathbf{D} \neq \mathbf{0}$, the reduced system is strictly passive, and realizable with RLC circuit elements. In Sect. 4.4.3.2 we show one way of obtaining strictly passive reduced systems also when $\mathbf{D} = \mathbf{0}$.

4.4.3 The Dominant Spectral Zero Method

The new Dominant Spectral Zero Method (dominant SZM) is presented. The spectral zero method [114, 134] is extended with a dominance criterion for selecting finite spectral zeros. These are computed efficiently and automatically using the subspace accelerated dominant pole algorithm (SADPA) [130, 131]. We show in addition how, for certain RLC models, dominant spectral zeros at ∞ can also be easily interpolated.

4.4.3.1 Dominant Spectral Zeros and Implementation

In [134] it was shown that spectral zeros are solved efficiently from an associated Hamiltonian eigenvalue problem [127, 137]. In [114, 134] however, the selection of spectral zeros was still an open problem. We propose a solution as follows: we extend the concept of *dominance* from poles [130] to spectral zeros, and adapt the iterative solver SADPA for the computation of *dominant spectral zeros*. The corresponding invariant subspaces are obtained as a by-product of SADPA, and are used to construct the passivity preserving projection matrices $\mathbf{V}$ and $\mathbf{W}$. Essentially, dominant SZM is the SADPA-based implementation of modal approximation for the Hamiltonian system associated with $\mathbf{G}(s) = [\mathbf{H}(s)+\mathbf{H}^*(-s)]^{-1}$. Recalling Def. 4.5, the spectral zeros of Σ are the poles of $\mathbf{G}(s)$, with partial fraction expansion: $\mathbf{G}(s) = \sum_{j=1}^{2n} \frac{\mathscr{R}_j}{s-s_j}$, where s_i are the poles of $\mathbf{G}$ with associated residues $\mathscr{R}_j$ [126, 131]. The modal approximate of $\mathbf{G}(s)$ is obtained by truncating this sum: $\hat{\mathbf{G}}(s) = \sum_{j=1}^{2k} \frac{\mathscr{R}_j}{s-s_j}$. The procedure is outlined next.

1. Given $\varSigma(\mathbf{E},\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D})$, construct the associated Hamiltonian system $\varSigma_s$, associated with transfer function $\mathbf{G}(s)$:

   a. $\varSigma_s$ when $\mathbf{D}+\mathbf{D}^*$ is invertible:

$$\mathbf{A}_s = \begin{pmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & -\mathbf{A}^* & -\mathbf{C}^* \\ \mathbf{C} & \mathbf{B}^* & \mathbf{D}+\mathbf{D}^* \end{pmatrix}, \mathbf{E}_s = \begin{pmatrix} \mathbf{E} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{E}^* & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}, \mathbf{B}_s = \begin{pmatrix} \mathbf{B} \\ -\mathbf{C}^* \\ \mathbf{0} \end{pmatrix} \varDelta,$$

$$\mathbf{C}_s = -\varDelta \begin{pmatrix} \mathbf{C} & \mathbf{B}^* & \mathbf{0} \end{pmatrix},\ \mathbf{D}_s = \varDelta = (\mathbf{D}+\mathbf{D}^*)^{-1} \tag{4.57}$$

   b. $\varSigma_s$ when $\mathbf{D} = \mathbf{0}$:

$$\mathbf{A}_s = \begin{pmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & -\mathbf{A}^* & -\mathbf{C}^* \\ \mathbf{C} & \mathbf{B}^* & \mathbf{0} \end{pmatrix},\ \mathbf{E}_s = \begin{pmatrix} \mathbf{E} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{E}^* & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix},\ \mathbf{B}_s = \begin{pmatrix} \mathbf{B} \\ -\mathbf{C}^* \\ \mathbf{I} \end{pmatrix},\ \mathbf{C}_s = -\begin{pmatrix} \mathbf{C} & \mathbf{B}^* & \mathbf{I} \end{pmatrix} \tag{4.58}$$

2. Solve the Hamiltonian eigenvalue problem $(\varLambda, \mathbf{R}, \mathbf{L}) = \mathrm{eig}(\mathbf{A}_s, \mathbf{E}_s)$, i.e., $\mathbf{A}_s\mathbf{R} = \mathbf{E}_s\mathbf{R}\varLambda$, $\mathbf{L}^*\mathbf{A}_s = \varLambda\mathbf{L}^*\mathbf{E}_s$. $\mathbf{R} = [\mathbf{r}_1, \ldots, \mathbf{r}_{2n}]$, $\mathbf{L} = [\mathbf{l}_1, \ldots, \mathbf{l}_{2n}]$ and eigenvalues $\varLambda = \mathrm{diag}(s_1, \ldots, s_n, -s_1^*, \ldots, -s_n^*)$ are the spectral zeros of $\varSigma$.
3. Compute residues $\mathscr{R}_j$ associated with the stable[22] spectral zeros s_j, $j = 1 \ldots n$ as follows: $\mathscr{R}_j = \gamma_j \beta_j$, $\gamma_j = \mathbf{C}_s\mathbf{r}_j(\mathbf{l}_j^*\mathbf{E}_s\mathbf{r}_j)^{-1}$, $\beta_j = \mathbf{l}_j^*\mathbf{B}_s$.
4. Sort spectral zeros descendingly according to *dominance criterion* $\frac{\|\mathscr{R}_j\|}{|Re(s_j)|}$ [130, Chapter 3], and reorder right eigenvectors $\mathbf{R}$ accordingly.
5. Retain the right eigenspace $\hat{\mathbf{R}} = [\mathbf{r}_1,\ \ldots,\ \mathbf{r}_k] \in \mathbb{C}^{2n\times k}$, corresponding to the stable k most dominant spectral zeros.
6. Construct passivity projection matrices $\mathbf{V}$ and $\mathbf{W}$ from the rows of $\hat{\mathbf{R}}$: $\mathbf{V} = \hat{\mathbf{R}}_{[1:n,1:k]}$, $\mathbf{W} = \hat{\mathbf{R}}_{[n+1:2n,1:k]}$, and reduce $\varSigma$ according to (4.55).

As explained in [114, 125, 134], by projecting with (4.55), $\hat{\varSigma}$ interpolates the k most dominant spectral zeros of $\varSigma$, guaranteeing passivity and stability. For large-scale applications, a full solution to the eigenvalue problem in step 2, followed by the dominant sort 3–4 is computationally unfeasible. Instead, the iterative solver SADPA [130, Chapter 3] is applied with appropriate adaptations for spectral zero computation (see Appendix 4.4.6 for the pseudocode). SADPA implements steps 2–4 efficiently and automatically gives the k most dominant spectral zeros and associated $2n \times k$ right eigenspace $\hat{\mathbf{R}}$. The implementation requires performing an LU factorization of $(s_j\mathbf{E} - \mathbf{A})$ at each iteration. The relevant s_j are nevertheless computed automatically in SADPA, which may have several advantages over other methods (see [125] for a more detailed cost analysis).

[22] $s \in \mathbb{C}$ is stable if $Re(s) < 0$.

4.4.3.2 D = 0 and Dominance at $s \to \infty$

Systems arising in circuit simulation often satisfy $\mathbf{D} = \mathbf{0}$ in (4.54). In this case, the projection (4.55), with **W** and **V** obtained in step 6 in Sect. 4.4.3.1, gives a lossless system [125]. This is because **W** and **V** only interpolate dominant finite spectral zeros, whereas the original system has spectral zeros at ∞, some of which may be dominant [120]. A strictly passive system (with all poles in the left half plane) can nevertheless be obtained by recovering this dominant behavior. For systems often occurring in circuit simulation this is achieved as follows. Consider the modified nodal analysis (MNA) description of an RLC circuit:

$$\underbrace{\begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathscr{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathscr{L} \end{pmatrix}}_{\mathbf{E}} \underbrace{\frac{d}{dt} \begin{pmatrix} v_p \\ \boldsymbol{v}_i \\ \boldsymbol{i}_L \end{pmatrix}}_{\dot{\mathbf{x}}} + \underbrace{\begin{pmatrix} \mathscr{G}_{11} & \mathscr{G}_{12} & \mathscr{E}_1 \\ \mathscr{G}_{12}^* & \mathscr{G}_{22} & \mathscr{E}_2 \\ -\mathscr{E}_1^* & -\mathscr{E}_2^* & \mathbf{0} \end{pmatrix}}_{-\mathbf{A}} \underbrace{\begin{pmatrix} \boldsymbol{v}_p \\ \boldsymbol{v}_i \\ \boldsymbol{i}_L \end{pmatrix}}_{\mathbf{x}} = \underbrace{\begin{pmatrix} \mathscr{B}_1 \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}}_{\mathbf{B}} \mathbf{u}, \tag{4.59}$$

where $\mathbf{u}(t) \in \mathbb{R}^m$ are input currents and $\mathbf{y}(t) = \mathbf{C}\mathbf{x} \in \mathbb{R}^m$ are output voltages, $\mathbf{C} = \mathbf{B}^*$. The states are $\mathbf{x}(t) = [\mathbf{v}_p(t),\ \mathbf{v}_i(t),\ \mathbf{i}_L(t)]^T$, with $\mathbf{v}_p(t) \in \mathbb{R}^{n_p}$ the voltages at the input nodes (circuit terminals), $\mathbf{v}_i(t) \in \mathbb{R}^{n_i}$ the voltages at the internal nodes, and $\mathbf{i}_L(t) \in \mathbb{R}^{n_{i_L}}$ the currents through the inductors, $n_p + n_i + n_{i_L} = n$. $\mathscr{C}$ and $\mathscr{L}$ are the capacitor and inductor matrix stamps, respectively. With (4.59) it is assumed that no capacitors or inductors are directly connected to the input nodes, thus $\mathbf{B} \in Null(\mathbf{E})$ and $\mathbf{C}^* \in Null(\mathbf{E}^*)$. As **B** and **C** are right and left eigenvectors corresponding to dominant poles (and spectral zeros) at ∞ [120], the modified projection matrices are:

$$\tilde{\mathbf{W}} = [\mathbf{W}, \mathbf{C}^*],\ \tilde{\mathbf{V}} = [\mathbf{W}, \mathbf{B}], \tag{4.60}$$

where **W** and **V** are obtained from step 6 in Sect. 4.4.3.1. With (4.60), the finite dominant spectral zeros are interpolated as well as the dominant spectral zeros at ∞, and the reduced system is strictly passive [120]. In [125] two alternatives were proposed for ensuring strict passivity for systems in the more general form (4.54) with $\mathbf{D} = \mathbf{0}$.

4.4.3.3 Circuit Representation of Reduced Impedance Transfer Function

Reduced models obtained with dominant SZM and other Krylov-type methods (PRIMA [128], SPRIM [117, 118], SPRIM/IOPOR [115, 138]) are mathematical abstractions of an underlying small RLC circuit. Circuit simulators however can only handle mathematical representations to a limited extent, and reduced models have to be synthesized with RLC circuit elements. We reduce all circuits with respect to the input impedance transfer function (i.e., the inputs are the currents injected into the circuit terminals and the outputs are the voltages measured at the terminals) [123]. After converting the reduced input impedance transfer function to netlist format, the reduced circuit can be driven easily by currents or voltages when simulated. Thus both the input impedance and admittance of an original model can be reproduced (see Sect. 4.4.4). Here, models obtained with dominant SZM are converted to netlist representations using the Foster impedance realization approach [119, 122]. Netlist formats for the SPRIM/IOPOR [115, 117, 138] reduced models are obtained via the RLCSYN unstamping procedure in [123, 138]. With both approaches, the resulting netlists may still contain circuit elements with negative values, nevertheless this does not impede the circuit simulation. Obtaining realistic synthesized models with positive circuit elements only is still an open problem.

4.4.4 Numerical Results

Two transmission line models are reduced with the proposed dominant spectral zero method and compared with the input-output structure preserving method SPRIM/IOPOR [115, 117, 138]. For both circuits, the circuit simulators[23] yield systems in the form (4.59), thus the dominant SZM projection is (4.60). RLC netlist representations for the reduced models are obtained (see Sect. 4.4.3.3) and simulated with Pstar.

The RLC transmission line with connected voltage controlled current sources (VCCSs) from [125] is reduced with dominant SZM, SPRIM/IOPOR [117, 138] and modal approximation (MA). The transfer function is an input impedance i.e., the circuit is current driven. Matlab simulations of the original and reduced models, as well as the Pstar netlist simulations are shown in Fig. 4.19: the model reduced with Dominant SZM gives the best approximation. Table 4.2 summarizes the reduction: the number of circuit elements and the number of states were reduced significantly and the simulation time was sped up.

In [125], the voltage driven *input admittance* of an RLC transmission line (consisting of cascaded RLC blocks) was reduced directly as shown in Fig. 4.20. Here we reduce and synthesize the underlying *input impedance* of the same transmission line (see Figs. 4.21 and 4.22). When driving the reduced netlist by an input voltage during the actual circuit simulation, the same input admittance is obtained as if the input admittance had been reduced directly, as seen in Figs. 4.20 and 4.23. Table 4.3 summarizes the reduction results. Although the reduced mathematical

[23] Pstar and Hstar are in-house simulators at NXP Semiconductors, Eindhoven, The Netherlands

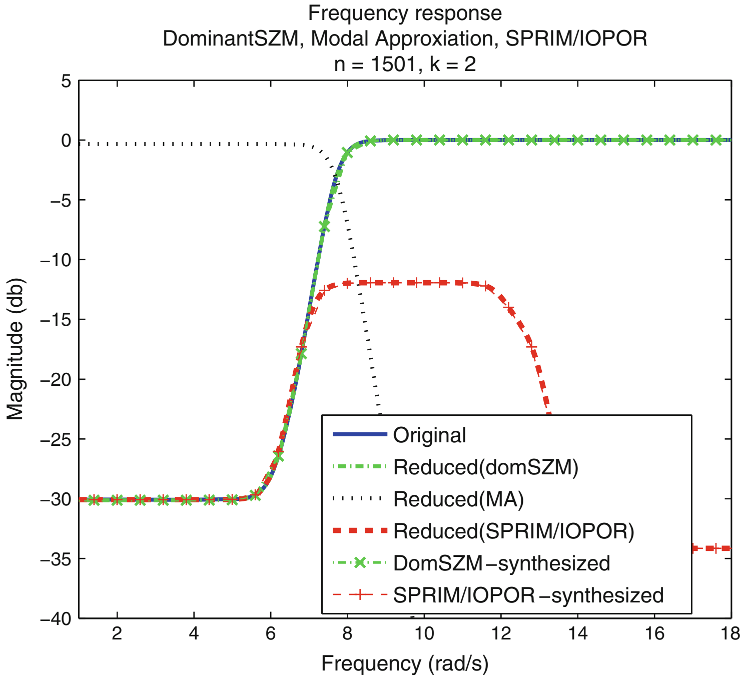

Fig. 4.19 Original, reduced and synthesized systems: Dominant SZM, SPRIM/IOPOR

Table 4.2 Transmission line with VCCSs: reduction and synthesis summary

System	Dimension	R	C	L	VCCs	States	Simulation time
Original	1501	1001	500	500	500	1,500	0.5 s
Dominant SZM	2	3	2	0	–	4	0.01 s
SPRIM/IOPOR	2	6	3	1	–	4	0.01 s

models have the same dimension ($k = 23$), the reduction effect can only be determined after obtaining the netlist representations. Although the SPRIM/IOPOR synthesized model has fewer states, it has more circuit elements than the dominant SZM model, i.e., the matrix stamp of the model is more dense. This suggests that simulation time is jointly determined by the number of states and the number of circuit elements. Thus for practical purposes it is critical to synthesize reduced models with RLC components.

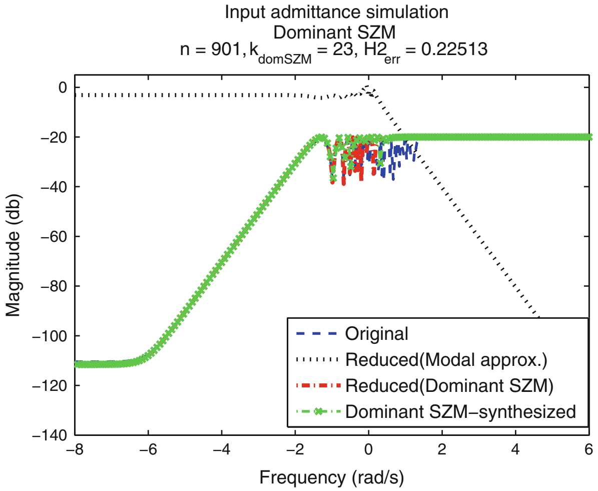

Fig. 4.20 Input admittance transfer function: original, synthesized Dominant SZM model

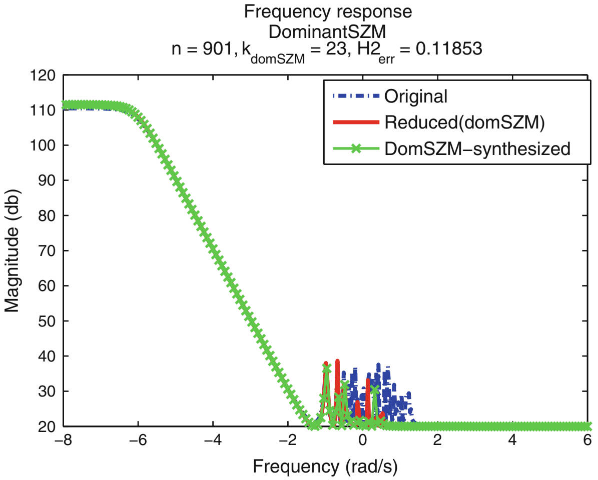

Fig. 4.21 Input impedance transfer function: original and reduced with Dominant SZM

4.4.5 Concluding Remarks

A novel passivity preserving model reduction method is presented, which is based on interpolation of dominant spectral zeros. Implemented with the SADPA eigenvalue solver, the method computes the partial eigenvalue decomposition of an

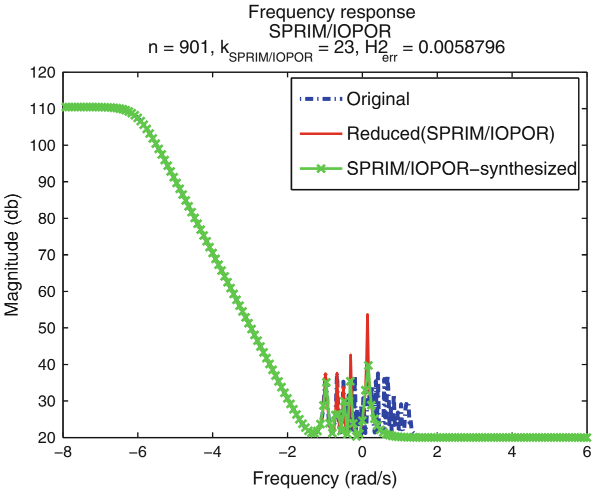

Fig. 4.22 Input impedance transfer function: original, reduced with SPRIM/IOPOR

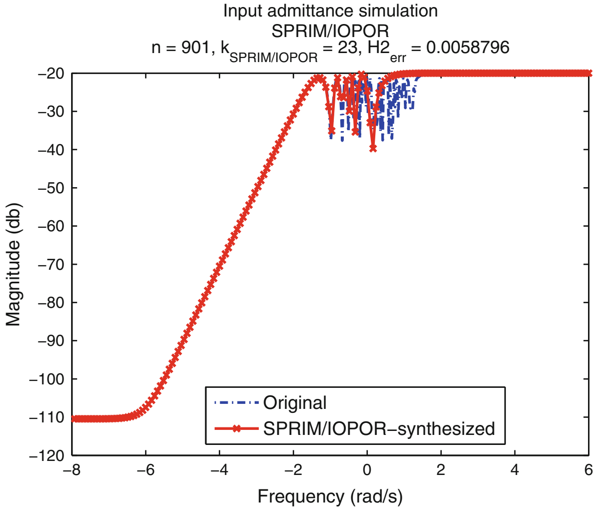

Fig. 4.23 Input admittance transfer function: original, synthesized SPRIM/IOPOR model

associated Hamiltonian matrix pair, and constructs the passivity preserving projection. Netlist equivalents for the reduced models are simulated and directions for future work are revealed. Especially in model reduction of multi-terminal circuits, achieving structure preservation, sparsity and small dimensionality simultaneously

Table 4.3 RLC transmission line: Input impedance reduction and synthesis summary

System	Dimension	R	C	L	States	Simulation time
Original	901	500	300	300	901	1.5 s
Dominant SZM	23	22	11	10	34	0.02 s
SPRIM/IOPOR	23	78	66	6	18	0.02 s

is an open question. New developments on sparsity-preserving model reduction for multi-terminal *RC* circuits can be found in [124]. In this context, RLC synthesis with positive circuit elements will also be addressed.

4.4.6 Appendix: SADPA for Computing Dominant Spectral Zeros

We outline SADPA for SISO systems; the MIMO implementation is similar and the code for computing dominant poles can be found in [132] or online [130]. The following pseudocode is extracted from [130, Chapter 3] and [131], with efficient modifications to automatically account for the four-fold symmetry $(\lambda, -\lambda^*, \lambda^*, -\lambda)$ of spectral zeros. In particular, as soon as a Hamiltonian eigenvalue (spectral zero) λ has converged, we immediately deflate the right/left eigenvectors corresponding to $-\lambda^*$ as well. It turns out that the right/left eigenvectors corresponding to $-\lambda^*$ need not be solved for explicitly. Rather, due to the structure of the Hamiltonian matrices [127, 137], they can be written down directly from the already converged left/right eigenvectors for λ, as shown in steps 14–17 of Algorithm 4.9. As for modal approximation [131], [130, Chapter 3] deflation for λ^* and $-\lambda$ is automatically handled in Algorithm 4.11. To summarize, once the right/left eigenvectors corresponding to an eigenvalue λ have converged, the right/left eigenvectors corresponding to $-\lambda^*, \lambda^*, -\lambda$ are also readily available at no additional computational cost, and can be immediately deflated.

In Algorithm 4.10, the MATLAB `qz` routine is proposed for solving the small, projected eigenvalue problem in step 1. This reveals the right/left eigenvectors $\tilde{\mathbf{X}}, \tilde{\mathbf{V}}$ of the projected pencil directly, however they are neither orthogonal nor bi-$\mathbf{G}$-orthogonal. Thus the normalization in step 3 is needed when computing the residues.

A modified Gram-Schimdt procedure (MGS) is used for orthonormalization. We used the implementation in [130, Algorithm 1.4]. For complete mathematical and algorithmic details of SADPA we refer to [130, Chapter 3] and [131].

Algorithm 4.9 $(\Lambda, \mathbf{R}, \mathbf{L})$ =SADPA$(\mathbf{E}_h, \mathbf{A}_h, \mathbf{B}_h, \mathbf{C}_h, s_1, \ldots p_{max}, k_{min}, k_{max})$

```
INPUT: $(\mathbf{E}_h, \mathbf{A}_h, \mathbf{B}_h, \mathbf{C}_h)$, $\mathbf{E}_h \in \mathbb{C}^{2n\times 2n}$, $\mathbf{A}_h \in \mathbb{C}^{2n\times 2n}$, $\mathbf{B}_h \in \mathbb{C}^{2n\times 1}$, $\mathbf{C}_h \in \mathbb{C}^{1\times 2n}$ an initial pole
    estimate $s_1$ and number of desired poles $p_{max}$ (in the restarted version, $k_{min}$ and $k_{max}$ are also
    specified)
OUTPUT: $\Lambda$, the $p_{max}$ most dominant eigenvalues and associated right, left eigenspaces R, L of
    $(\mathbf{A}_h, \mathbf{E}_h)$
$k = 1,\ p_{found} = 0,\ \Lambda = [],\ \mathbf{R} = [],\ \mathbf{L} = []$
while $p_{found} < p_{max}$ do
    Solve for x from $(s_k\mathbf{E}_h - \mathbf{A}_h)\mathbf{x} = \mathbf{B}_h$
    Solve for v from $(s_k\mathbf{E}_h - \mathbf{A}_h)^*\mathbf{v} = \mathbf{C}_h^*$
    $\mathbf{x}$ =MGS$(\mathbf{X}, \mathbf{x})$, $\mathbf{X} = [\mathbf{X}, \mathbf{x}/\|\mathbf{x}\|]$
    $\mathbf{v}$ =MGS$(\mathbf{V}, \mathbf{v})$, $\mathbf{V} = [\mathbf{V}, \mathbf{v}/\|\mathbf{v}\|]$
    Compute $\mathbf{G} = \mathbf{V}^*\mathbf{E}_h\mathbf{X}$ and $\mathbf{T} = \mathbf{V}^*\mathbf{A}_h\mathbf{X}$
    $(\tilde{\Lambda}, \tilde{\mathbf{X}}, \tilde{\mathbf{V}})$ = DomSort$(\mathbf{T}, \mathbf{G}, \mathbf{X}, \mathbf{V}, \mathbf{B}_h, \mathbf{C}_h)$        ▷ ▷ Algorithm 4.10
    Compute dominant approximate eigentriplet $(\hat{\lambda}_1, \hat{\mathbf{x}}_1, \hat{\mathbf{v}}_1)$:

        $$\hat{\lambda}_1 = \tilde{\lambda}_1, \hat{\mathbf{x}}_1 = (\mathbf{X}\tilde{\mathbf{x}}_1)/\|\mathbf{X}\tilde{\mathbf{x}}_1\|, \hat{\mathbf{v}}_1 = (\mathbf{V}\tilde{\mathbf{v}}_1)/\|\mathbf{V}\tilde{\mathbf{v}}_1\|$$

    if $\|\mathbf{A}_h\hat{\mathbf{x}}_1 - \mathbf{E}_h\hat{\mathbf{x}}_1\hat{\lambda}_1\| < \epsilon$ then
        $(\Lambda, \mathbf{R}, \mathbf{L}, \mathbf{X}, \mathbf{V}, \mathbf{B}_h, \mathbf{C}_h)$ = Deflate$(\hat{\lambda}_1, \hat{\mathbf{x}}_1, \hat{\mathbf{v}}_1, \Lambda, \mathbf{R}, \mathbf{L}, \mathbf{X}\tilde{\mathbf{X}}_{(:,2:k)}, \mathbf{V}\tilde{\mathbf{V}}_{(:,2:k)}, \mathbf{E}_h, \mathbf{B}_h, \mathbf{C}_h)$
                                                                    ▷ ▷ Algorithm 4.11
        $p_{found}++$ ▷ ▷ Also find eigenvectors for the antistable spectral zero $-\hat{\lambda}_1^*$ and deflate
        $\mathbf{x} = [\ -\hat{\mathbf{v}}_{1_{(n+1:2n,:)}};\ \hat{\mathbf{v}}_{1_{(1:n,:)}}\ ]$
        $\mathbf{v} = [\ \hat{\mathbf{x}}_{1_{(n+1:2n,:)}};\ -\hat{\mathbf{x}}_{1_{(1:n,:)}}\ ]$
        $(\Lambda, \mathbf{R}, \mathbf{L}, \mathbf{X}, \mathbf{V}, \mathbf{B}_h, \mathbf{C}_h)$ = Deflate$(-\hat{\lambda}_1^*, \mathbf{x}, \mathbf{v}, \Lambda, \mathbf{R}, \mathbf{L}, \mathbf{X}, \mathbf{V}, \mathbf{E}_h, \mathbf{B}_h, \mathbf{C}_h)$        ▷ ▷
    Algorithm 4.11
        $p_{found}++$
        $\tilde{\lambda}_1 = \tilde{\lambda}_2$
    else if ncols$(\tilde{\mathbf{X}}) > k_{max}$ then
                                                                    ▷ ▷ Possible restart
        ▷ ▷ Retain first $k_{min}$ most dominant approximate eigenvectors and re-orthonormalize
        $\mathbf{X}$ =MGS$(\mathbf{X}\tilde{\mathbf{X}}_{(:,1:k_{min})})$        ▷ ▷ Orthornormalize all columns sequentially
        $\mathbf{V}$ =MGS$(\mathbf{V}\tilde{\mathbf{V}}_{(:,1:k_{min})})$
    end if
    Increment $k = k + 1$
    Select new most dominant pole estimate $s_k = \tilde{\lambda}_1$
end while
```

Algorithm 4.10 $(\tilde{\Lambda}, \tilde{\mathbf{X}}, \tilde{\mathbf{V}})$ = DomSort$(\mathbf{T}, \mathbf{G}, \mathbf{X}, \mathbf{V}, \mathbf{B}_h, \mathbf{C}_h)$

```
INPUT: $(\mathbf{T}, \mathbf{G}), \mathbf{X}, \mathbf{V}, \mathbf{B}_h, \mathbf{C}_h$
OUTPUT: $(\tilde{\Lambda}, \tilde{\mathbf{X}}, \tilde{\mathbf{V}})$, $k$ dominant approximate eigenvalues and associated right, left eigenvectors
    of $(\mathbf{T}, \mathbf{G})$, sorted such that $\tilde{\lambda}_1$ is most dominant
$(AA, BB, Q, Z, \tilde{\mathbf{X}}, \tilde{\mathbf{V}})$ = QZ$(\mathbf{T}, \mathbf{G})$
$\tilde{\Lambda}$ = diag$(AA)$./diag$(BB)$ and $|\tilde{\lambda}_i| \neq \infty$, $i = 1 \ldots k$
$R_i = \frac{[\mathbf{C}_h\tilde{x}_i][\tilde{v}_i^*\mathbf{B}_h]}{\tilde{v}_i^*\mathbf{G}\tilde{x}_i}$        ▷ ▷ Compute residues
Sort $(\tilde{\Lambda}, \tilde{\mathbf{X}}, \tilde{\mathbf{V}})$ in decreasing $|R_i|/|Re(\tilde{\lambda}_i)|$ order
```

Algorithm 4.11 $(\Lambda, \mathbf{R}, \mathbf{L}, \mathbf{X}, \mathbf{V}, \mathbf{B}_h, \mathbf{C}_h) = \text{Deflate}(\hat{\lambda}, \hat{\mathbf{x}}, \hat{\mathbf{v}}, \ldots \Lambda, \mathbf{R}, \mathbf{L}, \hat{\mathbf{X}}, \hat{\mathbf{V}}, \mathbf{E}_h, \mathbf{B}_h, \mathbf{C}_h)$

INPUT: $(\hat{\lambda}, \hat{\mathbf{x}}, \hat{\mathbf{v}})$: the newly converged most dominant eigentriplet, $(\Lambda, \mathbf{R}, \mathbf{L})$: the dominant eigentriplets already found correctly, $\hat{\mathbf{X}}, \hat{\mathbf{V}}$: the approximate right/left eigenvectors not yet checked for convergence, $\mathbf{E}_h, \mathbf{B}_h, \mathbf{C}_h$

OUTPUT: $(\Lambda, \mathbf{R}, \mathbf{L})$: updated converged eigentriplets, $\mathbf{X}, \mathbf{V}$: deflated approximate eigenspaces, $\mathbf{B}_h, \mathbf{C}_h$: deflated matrices

```
$\Lambda = [\Lambda, \hat{\lambda}]$
$\hat{\mathbf{r}} = \hat{\mathbf{x}}/(\hat{\mathbf{v}}^*\mathbf{E}_h\hat{\mathbf{x}})$    ▷ ▷ For keeping converged eigenvectors bi-$\mathbf{E}$-orthogonal
$\hat{\mathbf{l}} = \hat{\mathbf{v}}$
$\mathbf{R} = [\mathbf{R}, \hat{\mathbf{r}}]$, $\mathbf{L} = [\mathbf{L}, \hat{\mathbf{l}}]$
Deflate $\mathbf{B}_h = \mathbf{B}_h - \mathbf{E}_h\hat{\mathbf{r}}(\hat{\mathbf{l}}^*\mathbf{B}_h)$
Deflate $\mathbf{C}_h = \mathbf{C}_h - (\mathbf{C}_h\hat{\mathbf{r}})\hat{\mathbf{l}}^*\mathbf{E}_h$
if imag($\hat{\lambda} \neq 0$) then
                                             ▷ ▷ Also deflate complex conjugate
    $\Lambda = [\Lambda, \hat{\lambda}^*]$
    $\hat{\mathbf{r}} = \hat{\mathbf{r}}^*$, $\hat{\mathbf{l}} = \hat{\mathbf{l}}^*$
    $\mathbf{R} = [\mathbf{R}, \hat{\mathbf{r}}]$, $\mathbf{L} = [\mathbf{L}, \hat{\mathbf{l}}]$
    Deflate $\mathbf{B}_h = \mathbf{B}_h - \mathbf{E}_h\hat{\mathbf{r}}(\hat{\mathbf{l}}^*\mathbf{B}_h)$
    Deflate $\mathbf{C}_h = \mathbf{C}_h - (\mathbf{C}_h\hat{\mathbf{r}})\hat{\mathbf{l}}^*\mathbf{E}_h$
end if
$\mathbf{X} = \mathbf{Y} = []$
for $j = 1 \ldots \#\text{cols}(\hat{\mathbf{X}})$ do
    $\mathbf{X} = \text{Expand}(\mathbf{X}, \mathbf{R}, \mathbf{L}, \mathbf{E}_h, \hat{\mathbf{x}}_j)$    ▷ ▷ Algorithm 4.12
    $\mathbf{V} = \text{Expand}(\mathbf{V}, \mathbf{R}, \mathbf{L}, \mathbf{E}_h^*, \hat{\mathbf{v}}_j)$    ▷ ▷ Algorithm 4.12
end for
```

Algorithm 4.12 $\mathbf{X} = \text{Expand}(\mathbf{X}, \mathbf{R}, \mathbf{L}, \mathbf{E}_h, \hat{\mathbf{x}})$

INPUT: $\mathbf{X} \in \mathbb{C}^{2n \times k}$ such that $\mathbf{X}\mathbf{X}^* = \mathbf{I}$, $(\mathbf{R}, \mathbf{L}) \in \mathbb{C}^{2n \times p}$: the correctly found right/left eigenvectors such that: $\mathbf{L}^*\mathbf{E}_h\mathbf{R}$ is diagonal and $\mathbf{L}^*\mathbf{E}_h\mathbf{X} = \mathbf{0}$, $\hat{\mathbf{x}}$: approximate eigenvector not yet checked for convergence, $\mathbf{E}_h$

OUTPUT: $\mathbf{X} \in \mathbb{C}^{2n \times (k+1)}$ expanded such that $\mathbf{X}\mathbf{X}^* = \mathbf{I}$

```
$\mathbf{x}_{k+1} = \prod_{j=1}^{p}\left(\mathbf{I} - \frac{\mathbf{r}_j\mathbf{l}_j^*\mathbf{E}_h}{\mathbf{l}_j^*\mathbf{E}_h\mathbf{r}_j}\right)\hat{\mathbf{x}}$
$\mathbf{x} = \text{MGS}(\mathbf{X}, \mathbf{x}_{k+1})$
$\mathbf{X} = [\mathbf{X}, \mathbf{x}/\|\mathbf{x}\|]$
```

4.5 A Framework for Synthesis of Reduced Order Models

The main motivation for this section comes from the need for a general framework for the (re)use of reduced order models in circuit simulation.[24] Although many model order reduction methods have been developed and evolved since the 1990s (see for instance [141, 146] for an overview), it is usually less clear how to use these

[24] Section 4.5 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 [156] and [159].

methods efficiently in industrial practice, e.g., in a circuit simulator. One reason can be that the reduced order model does not satisfy certain physical properties, for instance, it may not be stable or passive while the original system is. Failing to preserve these properties is typically inherent to the reduced order method used (or its implementation). Passivity (and stability implicitly) can nowadays be preserved via several methods [142, 151, 157, 166, 169, 173, 174], but none address the practical aspect of (re)using the reduced order models with circuit simulation software (e.g., SPICE [150]). While the original system is available in *netlist* format, the reduced order model is in general only available in *numerical* format. Typically, circuit simulators are not prepared for inputs of this form and would require additional software architecture to handle them. In contrast, a reduced model in *netlist* representation could be easily coupled to bigger systems and simulated.

Synthesis is the realization step needed to map the reduced order model into a netlist consisting of electrical circuit components [154, 170]. In [148] it was shown that passive systems (with positive real transfer functions) can be synthesized with positive R,L,C elements and transformers. Later developments [147] propose a method to circumvent the introduction of transformers, however the resulting realization is non-minimal (i.e., the number of electrical components generated during synthesis is too large). Allowing for possibly negative R, L, C values, other methods have been proposed via e.g. direct stamping [163, 166] or full realization [155, 167]. These mostly model the input/output connections of the reduced model with controlled sources.

In this section we consider two synthesis methods that do not involve controlled sources: (1) *Foster synthesis* [154], where the realization is done via the system's transfer function and (2) *RLCYSN synthesis by unstamping* [176], which exploits input-output structure preservation in the reduced system matrices [provided that the original system matrices are written in *modified nodal analysis (MNA)* representation]. The focus of this section is on structure preservation and RLCSYN, especially because synthesis by unstamping is simple to implement for both SISO and MIMO systems. Strengthening the result of [176], we give a simple procedure to reduce either current- or voltage-driven circuits directly in impedance form by removing all the sources. Such an impedance-based reduction enables synthesis without controlled sources. The reduced order model is available as a netlist, making it suitable for simulation and reuse in other designs. Similar software [149] is commercially available.

The material in this section is organized as follows. A brief mathematical formulation of model order reduction is given in Sect. 4.5.1. The Foster synthesis is presented in Sect. 4.5.2. In Sect. 4.5.3 we focus on reduction and synthesis with structure (and input/output) preservation. Section 4.5.3.1 describes the procedure to convert admittance models to impedance form, so that synthesized models are easily (re)used in simulation. Based on [176], Sect. 4.5.3.2 is an outline of SPRIM/IOPOR reduction and RLCSYN synthesis. Examples follow in Sect. 4.5.4, and Sect. 4.5.5 concludes.

4.5.1 Problem Formulation

In this section the dynamical systems $\Sigma(\mathbf{A}, \mathbf{E}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ are of the form $\mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$, $\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)$, where $\mathbf{A}, \mathbf{E} \in \mathbb{R}^{n\times n}$, $\mathbf{E}$ may be singular but the pencil $(\mathbf{A}, \mathbf{E})$ is regular, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{p\times n}$, $\mathbf{x}(t) \in \mathbb{R}^{n}$, and $\mathbf{u}(t) \in \mathbb{R}^{m}$, $\mathbf{y}(t) \in \mathbb{R}^{p}$, $\mathbf{D} \in \mathbb{R}^{p\times m}$. If $m, p > 1$, the system is called multiple-input multiple-output (MIMO), otherwise it is called single-input single-output (SISO). The frequency domain transfer function is defined as: $\mathbf{H}(s) = \mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$. For systems in MNA form arising in circuit simulation see Sect. 4.5.3.

The model order reduction problem is to find, given an n-th order (descriptor) dynamical system, a k-th order system: $\tilde{\mathbf{E}}\dot{\tilde{\mathbf{x}}}(t) = \tilde{\mathbf{A}}\tilde{\mathbf{x}}(t) + \tilde{\mathbf{B}}\mathbf{u}(t)$, $\tilde{\mathbf{y}}(t) = \tilde{\mathbf{C}}\tilde{\mathbf{x}}(t) + \mathbf{D}\mathbf{u}(t)$ where $k < n$, and $\tilde{\mathbf{E}}, \tilde{\mathbf{A}} \in \mathbb{R}^{k\times k}$, $\tilde{\mathbf{B}} \in \mathbb{R}^{k\times m}$, $\tilde{\mathbf{C}} \in \mathbb{R}^{p\times k}$, $\tilde{\mathbf{x}}(t) \in \mathbb{R}^{k}$, $\mathbf{u}(t) \in \mathbb{R}^{m}$, $\tilde{\mathbf{y}}(t) \in \mathbb{R}^{p}$, and $\mathbf{D} \in \mathbb{R}^{p\times m}$. The number of inputs and outputs is the same as for the original system, and the corresponding transfer function becomes: $\tilde{\mathbf{H}}(s) = \tilde{\mathbf{C}}(s\tilde{\mathbf{E}} - \tilde{\mathbf{A}})^{-1}\tilde{\mathbf{B}} + \mathbf{D}$. For an overview of model order reduction methods, see [141, 145, 146, 172]. Projection based model order reduction methods construct a reduced order model via Petrov-Galerkin projection:

$$\tilde{\Sigma}(\tilde{\mathbf{E}}, \tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}, \mathbf{D}) \equiv (\mathbf{W}^*\mathbf{E}\mathbf{V}, \mathbf{W}^*\mathbf{A}\mathbf{V}, \mathbf{W}^*\mathbf{B}, \mathbf{V}^*\mathbf{C}, \mathbf{D}), \tag{4.61}$$

where $\mathbf{V}, \mathbf{W} \in \mathbb{R}^{n\times k}$ are matrices whose $k < n$ columns form bases for relevant subspaces of the state-space. There are several projection methods, that differ in the way the matrices $\mathbf{V}$ and $\mathbf{W}$ are chosen. These also determine which properties are preserved after reduction. Some stability preserving methods are: *modal approximation* [171], *Poor Man's TBR* [168]. Among *moment matching* [152] methods, the following preserve passivity: *PRIMA* [166], *SPRIM* [151], *spectral zero interpolation*, [142, 157, 161, 173]. From the balancing methods, *balanced truncation* [144] preserves stability, and *positive real balanced truncation* [169, 174] preserves passivity.

4.5.2 Foster Synthesis of Rational Transfer Functions

This section describes the Foster synthesis method, which was developed in the 1930s by Foster and Cauer [154] and involves realization based on the system's transfer function. The Foster approach can be used to realize any reduced order model that is computed by standard projection based model order reduction techniques. Realizations will be described in terms of SISO impedances (Z-parameters). For equivalent realizations in terms of admittances (Y-parameters), see for instance [154, 175]. Given the reduced system (4.61) consider the partial fraction expansion [162] of its transfer function:

$$\tilde{\mathbf{H}}(s) = \sum_{i=1}^{k} \frac{\tilde{r}_i}{s - \tilde{p}_i} + \mathbf{D}, \tag{4.62}$$

The residues are $\tilde{r}_i = \frac{(\tilde{\mathbf{C}}\tilde{\mathbf{x}}_i)(\tilde{\mathbf{y}}_i^*\tilde{\mathbf{B}})}{\tilde{\mathbf{y}}_i^*\tilde{\mathbf{E}}\tilde{\mathbf{x}}_i}$ and the poles are $\tilde{p}_i$. An eigentriplet $(\tilde{p}_i, \tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)$ is composed of an eigenvalue $\tilde{p}_i$ of $(\tilde{\mathbf{A}}, \tilde{\mathbf{E}})$ and the corresponding right and left eigenvectors $\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i \in \mathbb{C}^k$. The expansion (4.62) consists of basic summands of the form:

$$Z(s) = r_1 + \frac{r_2}{s - p_2} + \frac{r_3}{s} + \left(\frac{r_4}{s - p_4} + \frac{\bar{r}_4}{s - \bar{p}_4} \right) + s r_6 + \left(\frac{r_7}{s - p_7} + \frac{r_7}{s - \bar{p}_7} \right), \tag{4.63}$$

where for completeness we can assume that any kind of poles may appear, i.e., either purely real, purely imaginary, in complex conjugate pairs, at ∞ or at 0 (see also Table 4.4). The Foster realization converts each term in (4.63) into the corresponding circuit block with R, L, C components, and connects these blocks in series in the final netlist. This is shown in Fig. 4.24. Note that any reordering of the circuit blocks in the realization of (4.63) in Fig. 4.24 still is a realization of (4.63). The values for the circuit components in Fig. 4.24 are determined according to Table 4.4.

The realization in netlist form can be implemented in any language such as SPICE [150], so that it can be reused and combined with other circuits as well. The advantages of Foster synthesis are: (1) its straightforward implementation for single-input-single-output (SISO) transfer functions, via either the impedance or the admittance transfer function, (2) for purely RC or RL circuits, netlists obtained from reduction via modal approximation [171] are guaranteed to have positive

Table 4.4 Circuit element values for Fig. 4.24 for the Foster impedance realization of (4.63)

Pole	Residue	R(Ohm)	C(F)	L(H)	G(Ohm$^-1$)
$p_1 = \infty$	$r_1 \in \mathbb{R}$	r_1			
$p_2 \in \mathbb{R}$	$r_2 \in \mathbb{R}$	$-\frac{r_2}{p_2}$	$\frac{1}{r_2}$		
$p_3 = 0$	$r_3 \in \mathbb{R}$		$\frac{1}{r_3}$		
$p_4 = \sigma + i\omega \in \mathbb{C}$ $p_5 \equiv \bar{p}_4$	$r_4 = \alpha + i\beta \in \mathbb{C}$ $r_5 \equiv \bar{r}_4$	$\frac{a_0}{a_1} L$	$\frac{1}{a_1}$	$\frac{a_1^3}{a_1^2 b_0 - a_0(a_1 b_1 - a_0)}$	$\frac{a_1 b_1 - a_0}{a_1^2}$
$a_0 = -2(\alpha\sigma + \beta\omega), \quad a_1 = 2\alpha, \quad b_0 = \sigma^2 + \omega^2, \quad b_1 = -2\sigma$					
$p_6 = \infty$	$r_6 \in \mathbb{R}$			r_6	
$p_7 \in i\mathbb{R}$ $p_8 \equiv \bar{p}_7$	$r_7 \in \mathbb{R}$ $r_8 \equiv \bar{r}_7$		$\frac{1}{r_7}$	$\frac{2r_7}{p_7 \bar{p}_7}$	

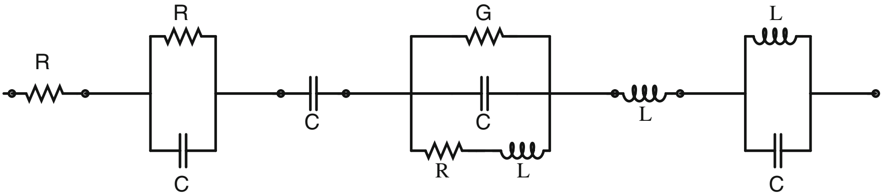

Fig. 4.24 Realization of a general impedance transfer function as a series *RLC* circuit

RC or *RL* values respectively [158]. The main disadvantage is that for multi-input-multi-output transfer functions, finding the Foster realization (see for instance [175]) is cumbersome and may also give dense reduced netlists (i.e., all nodes are interconnected). This is because the Foster synthesis of a k-dimensional reduced system with p terminals, will generally yield $O(p^2k)$ circuit elements. A method based on partitioning of an RLC circuit is found in [164]. The method produces a positive-valued, passive and stable reduced RLC circuit.

4.5.3 Structure Preservation and Synthesis by Unstamping

This section describes the second synthesis approach, which is based on *unstamping* the reduced matrix data into an *RLC* netlist and is denoted by RLCSYN [176]. It is suitable for obtaining netlist representations for models reduced via methods that preserve the MNA structure and the circuit terminals, such as the *input-output structure preserving* method SPRIM/IOPOR [176]. The strength of the result in [176] is that the input/output connectivity is synthesized after reduction without controlled sources, provided that the system is in *impedance form* (i.e., inputs are currents injected into the circuit terminals, and outputs are voltages measured at the terminals). Here, we interpret the input-output preservation as preserving the external nodes[25] of the original model during reduction. This way the reduced netlist can easily be coupled to other circuitry in place of the original netlist, and (re)using the reduced model in simulation becomes straightforward. The main drawback is that, when the reduced system matrices are dense and the number of terminals is large [$O(10^3)$], the netlist obtained from RLCSYN will be dense. For a k dimensional reduced network with p terminals, the RLCSYN synthesized netlist will generally have $O(p^2k^2)$ circuit elements. The density of the reduced netlist however is not a result of the synthesis procedure, but a consequence of the fact that the reduced system matrices are dense. Developments for sparsity preserving model reduction for multi-terminal circuits can be found in [160], where sparse netlists are obtained by synthesizing sparse reduced models via RLCSYN.

First, we motivate reduction and synthesis in impedance form, and show how it also applies for systems that are originally in admittance form. Then we explain model reduction via SPRIM/IOPOR, followed by RLCSYN synthesis.

4.5.3.1 A Simple Admittance to Impedance Conversion

In [176] it was shown how SPRIM/IOPOR preserves the structure of the input/output connectivity when the original model is in impedance form, allowing for

[25] A *terminal (external node)* is a node that is visible on the outside, i.e., a node in which currents can be injected. The other nodes are called *internal*.

synthesis via RLCSYN without controlled sources. The emerging question is: How to ensure synthesis without controlled sources, if the original model is in admittance form (i.e., it is voltage driven)? We show that reduction and synthesis via the input impedance transfer function is possible by removing any voltage sources from the original circuit a priori and re-inserting them in the reduced netlist if needed.

To this end, consider the modified nodal analysis (MNA) description of an input *admittance*[26] type *RLC* circuit, driven by voltage sources:

$$\underbrace{\begin{pmatrix}\mathscr{C} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathscr{L}\end{pmatrix}}_{\mathbf{E}_Y}\underbrace{\frac{d}{dt}\begin{pmatrix}\boldsymbol{v}(t)\\ \boldsymbol{i}_S(t)\\ \boldsymbol{i}_L(t)\end{pmatrix}}_{\dot{\mathbf{x}}_Y} + \underbrace{\begin{pmatrix}\mathscr{G} & \mathscr{E}_v & \mathscr{E}_l\\ -\mathscr{E}_v{}^* & \mathbf{0} & \mathbf{0}\\ -\mathscr{E}_l^* & \mathbf{0} & \mathbf{0}\end{pmatrix}}_{-\mathbf{A}_Y}\underbrace{\begin{pmatrix}\boldsymbol{v}(t)\\ \boldsymbol{i}_S(t)\\ \boldsymbol{i}_L(t)\end{pmatrix}}_{\mathbf{x}_Y} = \underbrace{\begin{pmatrix}\mathbf{0}\\ \mathscr{B}\\ \mathbf{0}\end{pmatrix}}_{\mathbf{B}_Y}\mathbf{u}(t), \quad (4.64)$$

where $\mathbf{u}(t) \in \mathbb{R}^{n_1}$ are input voltages and $\mathbf{y}(t) = \mathbf{C}_Y\mathbf{x}(t) \in \mathbb{R}^{n_1}$ are output currents, $\mathbf{C}_Y = \mathbf{B}_Y^*$. The states are $\mathbf{x}_Y(t) = [\mathbf{v}(t),\ \mathbf{i}_S(t),\ \mathbf{i}_L(t)]^T$, with $\mathbf{v}(t) \in \mathbb{R}^{n_v}$ the node voltages, $\mathbf{i}_S(t) \in \mathbb{R}^{n_1}$ the currents through the voltage sources, and $\mathbf{i}_L(t) \in \mathbb{R}^{n_l}$ the currents through the inductors, $n_v + n_1 + n_l = n$. The $n_v = n_1 + n_2$ node voltages correspond to the n_1 external nodes (i.e., the number of inputs/terminals) and the n_2 internal nodes.[27] Assuming without loss of generality that (4.64) is permuted such that the *first* n_1 nodes are the external nodes, we have: $\mathbf{v}_{1:n_1}(t) = \mathbf{u}(t)$. The dimensions of the underlying matrices are: $\mathscr{C} \in \mathbb{C}^{n_v \times n_v}$, $\mathscr{G} \in \mathbb{C}^{n_v \times n_v}$, $\mathscr{E}_v \in \mathbb{C}^{n_v \times n_1}$, $\mathscr{L} \in \mathbb{C}^{n_l \times n_l}$, $\mathscr{E}_l \in \mathbb{C}^{n_v \times n_l}$, $\mathscr{B} \in \mathbb{C}^{n_1 \times n_1}$. Recalling that $\mathbf{v}_{1:n_1}(t) = \mathbf{u}(t)$, the following holds:

$$\mathscr{E}_v = \begin{pmatrix}\mathscr{B}_v\\ \mathbf{0}_{n_2 \times n_1}\end{pmatrix} \in \mathbb{C}^{n_v \times n_1},\quad \mathscr{B}_v \in \mathbb{C}^{n_1 \times n_1},\quad \mathscr{B} = -\mathscr{B}_v. \qquad (4.65)$$

We derive the first order impedance-type system associated with (4.64). Note that by definition, $\mathbf{i}_S(t)$ flows **out of** the circuit terminals into the voltage source (i.e., from the + to the − terminal of the voltage source, see also [166, Figure 3] [158]). We can define new input currents as the currents flowing **into** the circuit terminals: $\mathbf{i}_{in}(t) = -\mathbf{i}_S(t)$. Since $\mathbf{u}(t) = \mathbf{v}_{1:n_1}(t)$ are the terminal voltages, they

[26] The subscript Y refers to quantities associated with a system in admittance form.

[27] For the pencil $(\mathbf{A}_Y, \mathbf{E}_Y)$ to be regular, in (4.64) one node must be chosen as a ground (reference) node; this is however only a numerical requirement.

describe the new output equations, and it is straightforward to rewrite (4.64) in the impedance form:

$$\left\{\begin{array}{l}\underbrace{\begin{pmatrix}\mathscr{C} & \mathbf{0}\\ \mathbf{0} & \mathscr{L}\end{pmatrix}}_{\mathbf{E}}\underbrace{\frac{d}{dt}\begin{pmatrix}\boldsymbol{v}(t)\\ \boldsymbol{i}_L(t)\end{pmatrix}}_{\dot{\mathbf{x}}}+\underbrace{\begin{pmatrix}\mathscr{G} & \mathscr{E}_l\\ -\mathscr{E}_l{}^* & \mathbf{0}\end{pmatrix}}_{-\mathbf{A}}\underbrace{\begin{pmatrix}\boldsymbol{v}(t)\\ \boldsymbol{i}_L(t)\end{pmatrix}}_{\mathbf{x}}=\underbrace{\begin{pmatrix}\mathscr{E}_v\\ \mathbf{0}\end{pmatrix}}_{\mathbf{B}}\mathbf{i}_{in}(t)\\ \underbrace{\begin{pmatrix}\mathscr{E}_v^* & \mathbf{0}\end{pmatrix}}_{\mathbf{C}}\underbrace{\begin{pmatrix}\boldsymbol{v}(t)\\ \boldsymbol{i}_L(t)\end{pmatrix}}_{\mathbf{x}}=\mathbf{y}(t) \;=\; \mathscr{B}_v\mathbf{v}_{1:n_1}(t), \quad \mathscr{E}_v^* = \begin{pmatrix}\mathscr{B}_v^* & \mathbf{0}_{n_1\times n_2}\end{pmatrix}\end{array}\right. \tag{4.66}$$

where **B** describes the new *input incidence matrix* corresponding the input currents, $\mathbf{i}_{in}$. The new *output incidence matrix* is **C**, corresponding to the voltages at the circuit terminals. We emphasize that (4.66) has fewer unknowns than (4.64), since the currents $\mathbf{i}_S$ have been eliminated. The transfer function associated to (4.66) is an input impedance: $\mathbf{H}(s) = \frac{\mathbf{y}(s)}{\mathbf{i}_{in}(s)}$. In Sect. 4.5.3.2 we explain how to obtain an impedance type reduced order model in input/output structure preserved form:

$$\left\{\begin{array}{l}\underbrace{\begin{pmatrix}\tilde{\mathscr{C}} & \mathbf{0}\\ \mathbf{0} & \tilde{\mathscr{L}}\end{pmatrix}}_{\tilde{\mathbf{E}}}\underbrace{\frac{d}{dt}\begin{pmatrix}\tilde{\boldsymbol{v}}(t)\\ \tilde{\boldsymbol{i}}_L(t)\end{pmatrix}}_{\dot{\tilde{\mathbf{x}}}}+\underbrace{\begin{pmatrix}\tilde{\mathscr{G}} & \tilde{\mathscr{E}}_l\\ -\tilde{\mathscr{E}}_l^* & \mathbf{0}\end{pmatrix}}_{-\tilde{\mathbf{A}}}\underbrace{\begin{pmatrix}\tilde{\boldsymbol{v}}(t)\\ \tilde{\boldsymbol{i}}_L(t)\end{pmatrix}}_{\tilde{\mathbf{x}}}=\underbrace{\begin{pmatrix}\tilde{\mathscr{E}}_v\\ \mathbf{0}\end{pmatrix}}_{\tilde{\mathbf{B}}}\mathbf{i}_{in}(t)\\ \underbrace{\begin{pmatrix}\tilde{\mathscr{E}}_v^* & \mathbf{0}\end{pmatrix}}_{\tilde{\mathbf{C}}}\underbrace{\begin{pmatrix}\tilde{\boldsymbol{v}}(t)\\ \tilde{\boldsymbol{i}}_L(t)\end{pmatrix}}_{\tilde{\mathbf{x}}}=\mathbf{y}(t) \;=\; \mathscr{B}_v\mathbf{v}_{1:n_1}(t), \;\; \tilde{\mathscr{E}}_v^* = \begin{pmatrix}\mathscr{B}_v^* & \mathbf{0}_{n_1\times k_2}\end{pmatrix}\end{array}\right. \tag{4.67}$$

where $\tilde{\mathscr{C}}$, $\tilde{\mathscr{L}}$, $\tilde{\mathscr{G}}$, $\tilde{\mathscr{E}}_v$ are the reduced MNA matrices, and the reduced input impedance transfer function is: $\tilde{\mathbf{H}}(s) = \frac{\tilde{\mathbf{y}}(s)}{\mathbf{i}_{in}(s)}$. Due to the input/output preservation, the circuit terminals are easily preserved in the reduced model (4.67). The simple example in Sect. 4.5.4.1 illustrates the procedure just described.

It turns out that after reduction and synthesis, the reduced model (4.67) can still be used as a voltage driven admittance block in simulation. This is shown next. We can rewrite the second equation in (4.67) as: $\begin{pmatrix}-\tilde{\mathscr{E}}_v^* & \mathbf{0} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\tilde{\mathbf{v}}(t)^T & \tilde{\mathbf{i}}_S(t)^T & \tilde{\mathbf{i}}_L(t)^T\end{pmatrix}^T = \mathscr{B}\mathbf{u}(t)$. This result together with $\mathbf{i}_{in}(t) = -\mathbf{i}_S(t)$, reveals that (4.67) can be rewritten as:

$$\underbrace{\begin{pmatrix}\tilde{\mathscr{C}} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \tilde{\mathscr{L}}\end{pmatrix}}_{\tilde{\mathbf{E}}_Y}\underbrace{\frac{d}{dt}\begin{pmatrix}\tilde{\boldsymbol{v}}(t)\\ \boldsymbol{i}_S(t)\\ \tilde{\boldsymbol{i}}_L(t)\end{pmatrix}}_{\dot{\tilde{\mathbf{x}}}_Y}+\underbrace{\begin{pmatrix}\tilde{\mathscr{G}} & \tilde{\mathscr{E}}_v & \tilde{\mathscr{E}}_l\\ -\tilde{\mathscr{E}}_v^* & \mathbf{0} & \mathbf{0}\\ -\tilde{\mathscr{E}}_l^* & \mathbf{0} & \mathbf{0}\end{pmatrix}}_{-\tilde{\mathbf{A}}_Y}\underbrace{\begin{pmatrix}\tilde{\boldsymbol{v}}(t)\\ \boldsymbol{i}_S(t)\\ \tilde{\boldsymbol{i}}_L(t)\end{pmatrix}}_{\tilde{\mathbf{x}}_Y}=\underbrace{\begin{pmatrix}\mathbf{0}\\ \mathscr{B}\\ \mathbf{0}\end{pmatrix}}_{\tilde{\mathbf{B}}_Y}\mathbf{u}(t), \tag{4.68}$$

which has the same structure as the original admittance model (4.64). Conceptually one could have reduced system (4.64) directly via the input admittance. In that case, synthesis by unstamping via RLCSYN [176] would have required controlled sources [155] to model the connections at the circuit terminals. As shown above, this is avoided by: applying the simple admittance-to-impedance conversion (4.64) to (4.66), reducing (4.66) to (4.67), and finally reinserting voltage sources after synthesis [if the input-output structure preserved admittance reduced admittance (4.68) is needed]. Being only a pre- and post-processing step, the proposed voltage-source removal and re-insertion can be applied irrespective of the model reduction algorithm used. For ease of understanding we relate it here to model reduction via SPRIM/IOPOR.

4.5.3.2 I/O Structure Preserving Reduction and RLCSYN Synthesis

The reduced input impedance model (4.67) is obtained via the input-output structure preserving SPRIM/IOPOR projection [176] as follows. Let $\mathbf{V} = \left(\mathbf{V}_1^T, \mathbf{V}_2^T, \mathbf{V}_3^T\right)^T \in \mathbb{C}^{((n_1+n_2+n_l)\times k)}$ be the projection matrix obtained with PRIMA [166], where $\mathbf{V}_1 \in \mathbb{C}^{(n_1\times k)}$, $\mathbf{V}_2 \in \mathbb{C}^{(n_2\times k)}$, $\mathbf{V}_3 \in \mathbb{C}^{(n_l\times k)}$, $k \geq n_1$, $i = 1\ldots 3$. After appropriate orthonormalization (e.g., via Modified Gram-Schmidt [171, Chapter 1]), we obtain: $\tilde{\mathbf{V}}_i = \text{orth}(\mathbf{V}_i) \in \mathbb{C}^{n_i\times k_i}$,$k_i \leq k$. The SPRIM [151] block structure preserving projection is: $\tilde{\mathbf{V}} = \text{blkdiag}\left(\tilde{\mathbf{V}}_1, \tilde{\mathbf{V}}_2, \tilde{\mathbf{V}}_3\right) \in \mathbb{C}^{n\times(k_1+k_2+k_3)}$, which does not yet preserve the structure of the input and output matrices. The input-output structure preserving SPRIM/IOPOR [176] projection is $\tilde{\mathbf{W}} = \begin{pmatrix}\mathbf{W} & \mathbf{0}\\ \mathbf{0} & \tilde{\mathbf{V}}_3\end{pmatrix} \in \mathbb{C}^{n\times(n_1+k_2+k_3)}$ where:

$$\mathbf{W} = \begin{pmatrix}\mathbf{I}_{n_1\times n_1} & \mathbf{0}\\ \mathbf{0} & \tilde{\mathbf{V}}_2\end{pmatrix} \in \mathbb{C}^{(n_1+n_2)\times(n_1+k_2)}. \tag{4.69}$$

Recalling (4.65), we obtain the reduced system matrices in (4.67): $\tilde{\mathscr{C}} = \mathbf{W}^*\mathscr{C}\mathbf{W}$, $\tilde{\mathscr{G}} = \mathbf{W}^*\mathscr{G}\mathbf{W}$, $\tilde{\mathscr{L}} = \tilde{\mathbf{V}}_3^*\mathscr{L}\tilde{\mathbf{V}}_3$, $\tilde{\mathscr{E}}_l = \mathbf{W}^*\mathscr{E}_l\tilde{\mathbf{V}}_3$, $\tilde{\mathscr{E}}_v = \mathbf{W}^*\mathscr{E}_v = \left(\mathscr{B}_v^* \; \mathbf{0}_{n_1\times k_2}\right)^*$, which compared to (4.65) clearly preserve input-output structure. Therefore a netlist representation for the reduced impedance-type model can be obtained, that is driven injected currents just as the original circuit. This is done via the RLCSYN [176] unstamping procedure. To this end, we use the Laplace transform and convert (4.67) to the second order form:

$$\begin{cases} [s\tilde{\mathscr{C}} + \tilde{\mathscr{G}} + \frac{1}{s}\tilde{\Gamma}]\tilde{\mathbf{v}}(s) = \tilde{\mathscr{E}}_v\mathbf{i}_{in}(s)\\ \tilde{\mathbf{y}}(s) = \tilde{\mathscr{E}}_v^*\tilde{\mathbf{v}}(s), \end{cases} \tag{4.70}$$

where $\tilde{\mathbf{i}}_L(s) = \frac{1}{s}\tilde{\mathscr{L}}^{-1}\left(\tilde{\mathscr{E}_l}^*\right)\tilde{\mathbf{v}}(s)$ and $\tilde{\Gamma} = \tilde{\mathscr{E}}_l\tilde{\mathscr{L}}^{-1}\tilde{\mathscr{E}}_l^*$.

The presentation of RLCSYN follows [176, Sect. 4], [158] and is only summarized here. In circuit simulation, the process of forming the $\mathscr{C}, \mathscr{G}, \mathscr{L}$ system matrices from the individual branch element values is called "stamping". The reverse operation of "unstamping" involves decomposing entry-wise the values of the reduced system matrices in (4.70) into the corresponding R, L, and C values. When applied on reduced models, the unstamping procedure may produce negative circuit elements because the reduced system matrices are no longer diagonally dominant (while the original matrices were). Obtaining positive circuit elements only is subject to further research. The resulting Rs, Ls and Cs are connected in the reduced netlist according to the MNA topology. The reduced input/output matrices of (4.70) directly reveal the input connections in the reduced model via injected currents, without any controlling elements. The prerequisites for an unstamping realization procedure therefore are:

1. The original system is in MNA impedance form (4.66). If the system is of admittance type (4.64), apply the admittance-to-impedance conversion from Sect. 4.5.3.1.
2. In (4.66), no Ls are directly connected to the input terminals so that, after reduction, diagonalization and regularization preserve the input/output structure.
3. System (4.66) is reduced with SPRIM/IOPOR [176] to (4.67) and converted to second order form (4.70). The alternative is to obtain the second order form of the original system first, and reduce it directly with SAPOR/IOPOR [143, 176].
4. The reduced system (4.70) must be diagonalized and regularized according to [176]. Diagonalization ensures that all inductors in the synthesized model are connected to ground (i.e., there are no inductor loops). Regularization eliminates spurious over-large inductors. These steps however are not needed for purely RC circuits.

4.5.4 Numerical Examples

We apply the proposed reduction and synthesis framework on several test cases. The first is a simple circuit which illustrates the complete admittance-to-impedance formulation and the RLCSYN unstampting procedure, as described in Sect. 4.5.3. The second example is a SISO transmission line model, while the third is a MIMO model of a spiral inductor.

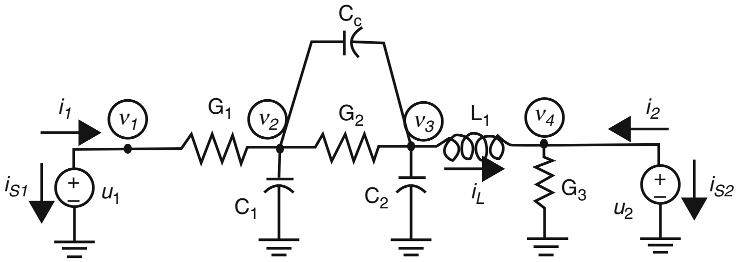

Fig. 4.25 Admittance-type circuit driven by input voltages [166]. $G_{1,2,3} = 0.1S$, $L_1 = 10^{-3}H$, $C_{1,2} = 10^{-6}$, $C_c = 10^{-4}$, $\|u_{1,2}\| = 1$

4.5.4.1 Illustrative Example

The circuit in Fig. 4.25 is voltage driven, and the MNA admittance form (4.64) is:

$$\left(\begin{array}{cccc|cc|c} 0\,0 & & 0 & 0 & 0 & 0 & 0 \\ 0\,0 & & 0 & 0 & 0 & 0 & 0 \\ 0\,0 & & C_1+C_c & -C_c & 0 & 0 & 0 \\ 0\,0 & & -C_c & C_2+C_c & 0 & 0 & 0 \\ \hline 0\,0 & & 0 & 0 & 0 & 0 & 0 \\ 0\,0 & & 0 & 0 & 0 & 0 & 0 \\ \hline 0\,0 & & 0 & 0 & 0 & 0 & L \end{array}\right) \dot{\begin{pmatrix} v_1 \\ v_4 \\ v_2 \\ v_3 \\ i_{S_1} \\ i_{S_2} \\ i_L \end{pmatrix}} + \left(\begin{array}{cccc|cc|c} G_1 & 0 & -G_1 & 0 & 1 & 0 & 0 \\ 0 & G_3 & 0 & 0 & 0 & 1 & 1 \\ -G_1 & 0 & G_1+G_2 & -G_2 & 0 & 0 & 0 \\ 0 & 0 & -G_2 & G_2 & 0 & 0 & 1 \\ \hline -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & -1 & 0 & 0 & 0 \end{array}\right) \begin{pmatrix} v_1 \\ v_4 \\ v_2 \\ v_3 \\ i_{S_1} \\ i_{S_2} \\ i_L \end{pmatrix} = \left(\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \hline -1 & 0 \\ 0 & -1 \\ \hline 0 & 0 \end{array}\right) \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \tag{4.71}$$

Notice that

$$\mathbf{i}_{in} = \begin{pmatrix} i_1 \\ i_2 \end{pmatrix} = - \begin{pmatrix} i_{S_1} \\ i_{S_2} \end{pmatrix} \tag{4.72}$$

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} v_1 \\ v_4 \end{pmatrix}, \tag{4.73}$$

thus the external nodes (input nodes/terminals) are v_1 and v_4, and the internal nodes are v_2 and v_3. As described in Sect. 4.5.3.1, (4.71) has an equivalent impedance formulation (4.66), with:

$$\mathscr{C} = \begin{pmatrix} 0\,0 & 0 & 0 \\ 0\,0 & 0 & 0 \\ 0\,0 & C_1+C_c & -C_c \\ 0\,0 & -C_c & C_2+C_c \end{pmatrix}, \ \mathscr{L} = \begin{pmatrix} L \end{pmatrix}, \ \mathscr{G} = \begin{pmatrix} G_1 & 0 & -G_1 & 0 \\ 0 & G_3 & 0 & 0 \\ -G_1 & 0 & G_1+G_2 & -G_2 \\ 0 & 0 & -G_2 & G_2 \end{pmatrix}, \ \mathscr{E}_l = \begin{pmatrix} 0 \\ -1 \\ 0 \\ 1 \end{pmatrix} \tag{4.74}$$

$$\mathscr{E}_v = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \ \mathscr{B} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}, \ \mathscr{B}_v = -\mathscr{B} \tag{4.75}$$

Matrices (4.74) and (4.75) are reduced either in first order form using SPRIM/IOPOR according to Sect. 4.5.3.2.

Here we reduce the circuit with SPRIM/IOPOR and synthesize it by unstamping via RLCSYN. Note that there is an L directly connected to the second input node v_4, thus assumption 2. from RLCSYN is not satisfied. We thus reduce and synthesize the single-input-single-output version of (4.71) only, where the second input i_2 is removed. Therefore the new incidence matrices are:

$$\mathscr{E}_{v_1} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \mathscr{B}_1 = \begin{pmatrix} -1 \end{pmatrix}, \ \mathscr{B}_{v_1} = -\mathscr{B}_1. \tag{4.76}$$

We choose an underlying PRIMA projection matrix $\mathbf{V} \in \mathbb{C}^{n \times k}$ spanning a $k = 2$-dimensional Krylov subspace (with expansion point $s_0 = 0$). According to Sect. 4.5.3.2, after splitting $\mathbf{V}$ and appropriate re-orthonormalization, the dimensions of the input-output structure preserving partitioning are:

$$n_1 = 1, \ n_2 = 3, \ n_l = 1, \ k_2 = 2, \ k_3 = 1, \tag{4.77}$$

and the SPRIM/IOPOR projection is:

$$\tilde{\mathbf{W}} = \left(\begin{array}{lll|l} 1 & 0 & 0 & 0 \\ 0 & 4.082 \cdot 10^{-1} & -4.861 \cdot 10^{-1} & 0 \\ 0 & 8.164 \cdot 10^{-1} & 5.729 \cdot 10^{-1} & 0 \\ 0 & 4.082 \cdot 10^{-1} & -6.597 \cdot 10^{-1} & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \in \mathbb{C}^{5 \times 4}, \ \text{with } \mathbf{W} \in \mathbb{C}^{4 \times 3}. \tag{4.78}$$

After diagonalization and regularization, the SPRIM/IOPOR reduced system matrices in (4.70) are:

$$\tilde{\mathscr{C}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1.749 \cdot 10^{-5} & -5.052 \cdot 10^{-5} \\ 0 & -5.052 \cdot 10^{-5} & 1.527 \cdot 10^{-4} \end{pmatrix}, \ \tilde{\mathscr{G}} = \begin{pmatrix} 1 & 8.165 \cdot 10^{-2} & -5.729 \cdot 10^{-2} \\ 8.165 \cdot 10^{-2} & 9.999 \cdot 10^{-2} & -7.726 \cdot 10^{-2} \\ -5.7295 \cdot 10^{-2} & -7.7265 \cdot 10^{-2} & 2.084 \cdot 10^{-1} \end{pmatrix}$$
$$\tilde{\Gamma} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 30.14 \end{pmatrix}, \ \tilde{\mathscr{E}}_{v_1} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \tag{4.79}$$

Reduced matrices (4.79) are now unstamped individually using RLCSYN. The reduced system dimension in second order form is thus $N = 3$, indicating that the reduced netlist will have 3 nodes and an additional ground node. In the following, we denote by $M_{i,j}$ $i = 1 \dots N$, $j = 0 \dots N - 1$ a circuit element connected between nodes (i, j) in the resulting netlist. M represents a circuit element of the type: R,L,C or current source J.

By unstamping $\tilde{\mathscr{G}}$, we obtain the following R values (for simplicity only 4 figures behind the period are shown here, nevertheless in implementation they are computed with machine precision $\epsilon = 10^{-16}$):

$$R_{1,0} = \left[\sum_{k=1}^{3} \tilde{\mathscr{G}}_{(1,k)}\right]^{-1} = 8.0417\ \Omega,\ \ R_{1,2} = -\left[\tilde{\mathscr{G}}_{(1,2)}\right]^{-1} = -12.247\ \Omega,\ \ R_{1,3} = -\left[\tilde{\mathscr{G}}_{(1,3)}\right]^{-1} = 17.452\ \Omega,$$

$$R_{2,0} = \left[\sum_{k=1}^{3} \tilde{\mathscr{G}}_{(2,k)}\right]^{-1} = 9.5798\ \Omega,\ \ R_{2,3} = -\left[\tilde{\mathscr{G}}_{(2,3)}\right]^{-1} = 12.942\ \Omega,\ \ R_{3,0} = \left[\sum_{k=1}^{3} \tilde{\mathscr{G}}_{(3,k)}\right]^{-1} = 13.535\ \Omega.$$

By unstamping $\tilde{\mathscr{C}}$, we obtain the following C values:

$$C_{2,0} = \sum_{k=1}^{3} \tilde{\mathscr{C}}_{(2,k)} = -3.3026 \cdot 10^{-5}\ F,\ \ C_{2,3} = -\tilde{\mathscr{C}}_{(2,3)} = 5.0526 \cdot 10^{-5}\ F,$$

$$C_{3,0} = \left[\sum_{k=1}^{3} \tilde{\mathscr{C}}_{(3,k)}\right]^{-1} = 1.0221 \cdot 10^{-4}\ F.$$

By unstamping $\tilde{\Gamma}$, we obtain the following L values:

$$L_{3,0} = \left[\sum_{k=1}^{3} \tilde{\Gamma}_{(3,k)}\right]^{-1} = 3.317 \cdot 10^{-2}\ H.$$

By unstamping $\tilde{\mathscr{E}}_{v_1}$, we obtain the current source $J_{1,0}$ of amplitude 1 A.

The Pstar [165] equivalent netlist is shown below:.

```
circuit;
     r r_1_0 (1, 0) 8.0417250765565598e+000;
     r r_1_2 (1, 2) -1.2247448713915894e+001;
     r r_1_3 (1, 3) 1.7452546181796258e+001;
     r r_2_0 (2, 0) 9.5798755840972589e+000;
     r r_2_3 (2, 3) 1.2942609947762115e+001;
     r r_3_0 (3, 0) 1.3535652691596653e+001;
     l l_3_0 (3, 0) 3.3170000000000033e-002;
     c c_2_0 (2, 0) -3.3026513336014821e-005;
     c c_2_3 (2, 3) 5.0526513336014765e-005;
     c c_3_0 (3, 0) 1.0221180442099465e-004;
     j j_1 (1, 0) sw(1, 0);
     c: Set node 1 as output: vn(1);
     c: Resistors 6;
     c: Capacitors 3;
     c: Inductors 1;
end;
```

Table 4.5 summarizes the reduction and synthesis results. Even though the number of internal variables (states) generated by the simulator is smaller for the SPRIM/IOPOR model than for the original, the number of circuit elements generated by RLCSYN is larger in the reduced model than in the original. Figure 4.26 shows that approximation with SPRIM/IOPOR is more accurate than with PRIMA. The Pstar simulation of the RLCSYN synthesized model also matches the MATLAB simulation of the reduced transfer function.

Table 4.5 Input impedance reduction (SPRIM/IOPOR) and synthesis (RLCSYN)

System	Dimension	*R*	*C*	*L*	States	Inputs/Outputs
Original	5	3	3	1	5	1
SPRIM/IOPOR	4	6	3	1	4	1

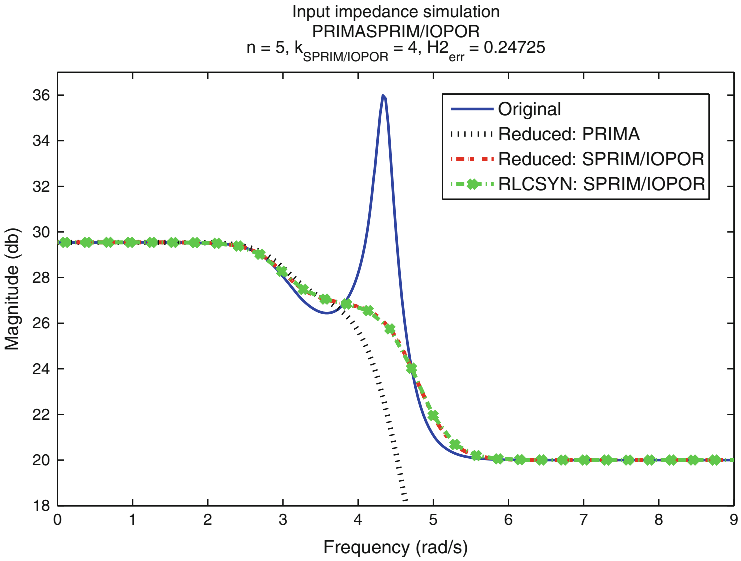

Fig. 4.26 Original, reduced and synthesized systems: PRIMA, SPRIM/IOPOR. The reduced and synthesized systems match but miss the peak around 4.5 rad/s

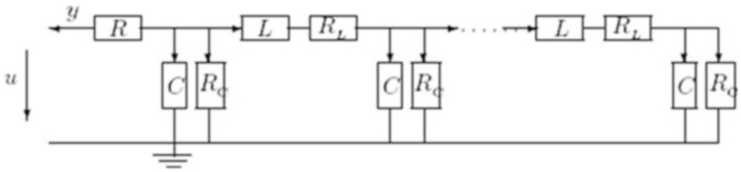

Fig. 4.27 Transmission line from Sect. 4.5.4.2

4.5.4.2 SISO *RLC* Network

We reduce the SISO *RLC* transmission line in Fig. 4.27. Note that the circuit is driven by the voltage **u**, thus it is of admittance type (4.64). The admittance simulation of the model reduced with the *dominant spectral zero method (Dominant SZM)* [157, 161], synthesized with the Foster approach, is shown in Fig. 4.28. The behavior of the original model is well approximated for the entire frequency range, and can also reproduce oscillations at dominant frequency points.

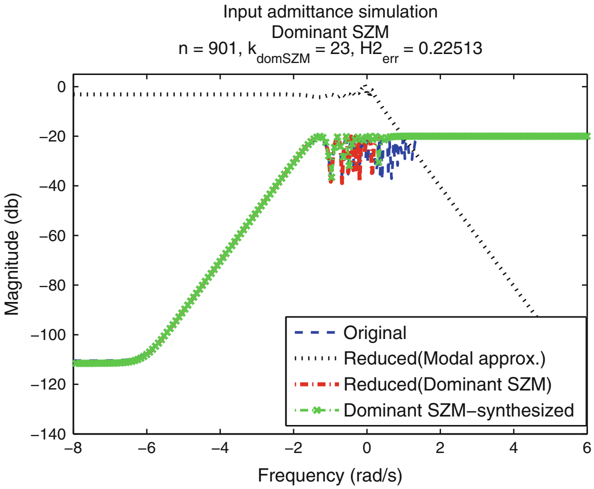

Fig. 4.28 Input admittance transfer function: original, reduced with Dominant SZM in admittance form and synthesized with Foster admittance

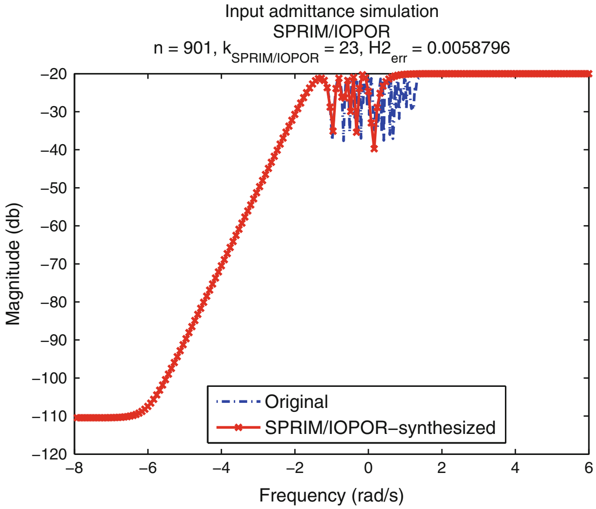

Fig. 4.29 Input admittance transfer function: original and synthesized SPRIM/IOPOR model (via impedance), after reconnecting the voltage source at the input terminal

In Fig. 4.29 the benefit of the admittance-to-impedance transformation, described in Sect. 4.5.3.1, is seen. By reducing the system in impedance form with SPRIM/-IOPOR and synthesizing (4.67) [using the second order form (4.70)] with RLCSYN [176], we are able to recover the reduced admittance (4.68) as well. The approximation is good for the entire frequency range.

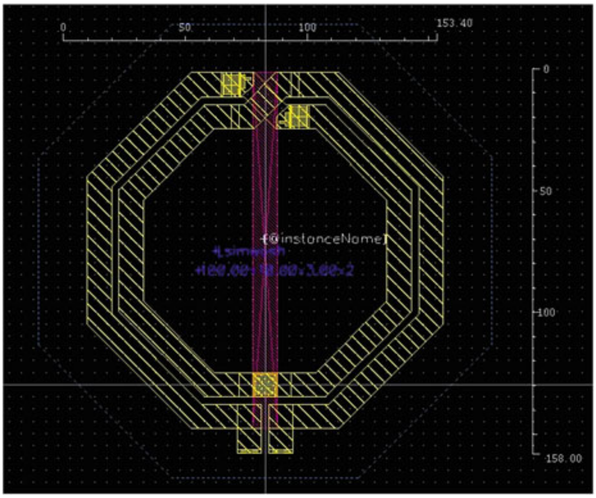

Fig. 4.30 Coil structure from Sect. 4.5.4.3

4.5.4.3 MIMO *RLC* Network

We reduce the MIMO *RLC* netlist resulting from the parasitic extraction [153] of the coil structure in Fig. 4.30. The model has 4 pins (external nodes). Pin 4 is connected to other circuit nodes only via C's, which causes the original model (4.66) to have a pole at 0. The example shows that the SPRIM/IOPOR model preserves the terminals and is synthesizable with RLCSYN without controlled sources.

Figure 4.31, shows the simulation of the transfer function from input 4 to output 4. SPRIM/IOPOR is more accurate around DC than PRIMA. Another alternative is to ground pin 4 prior to reduction. As seen from Fig. 4.32, SPRIM/IOPOR applied on the remaining 3-terminal system gives better approximation than PRIMA for the entire frequency range. With pin 4 grounded however, we loose the ability to (re)connect the synthesized model in simulation via all the terminals.

4.5.5 Conclusions and Outlook

A framework for realizing reduced mathematical models into *RLC* netlists was developed. Model reduction by projection for *RLC* circuits was described and associated with two synthesis approaches: Foster realization (for SISO transfer functions) and RLCSYN [176] synthesis by unstamping (for MIMO systems). An admittance-to-impedance conversion was prosed as a pre-model reduction step and shown to enable synthesis without controlled sources. The approaches were

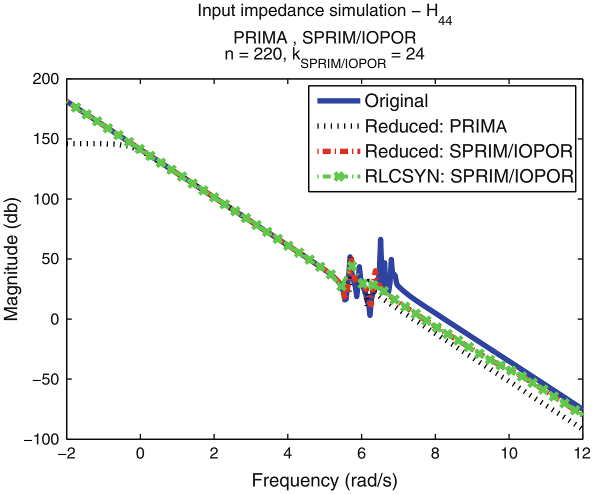

Fig. 4.31 Input impedance transfer function with "v_4" kept: $\mathbf{H}_{44}$ for PRIMA, SPRIM/IOPOR and RLCSYN realization

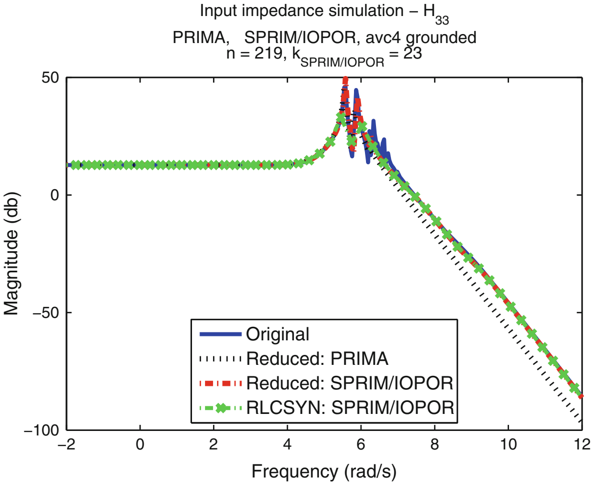

Fig. 4.32 Input impedance transfer function with "v_4" grounded: $\mathbf{H}_{33}$ for PRIMA, SPRIM/IOPOR and RLCSYN realization

tested on several examples. Future research will investigate reduction and synthesis methods for *RCLK* circuits with many terminals, while developments on sparsity-preserving model reduction for multi-terminal *RC* circuits can be found in [160].

References

References for Section 4.1

1. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems. SIAM, Philadelphia (2005)
2. Bai, Z.: Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. Appl. Numer. Math. **43**, 9–44 (2002). doi:10.1016/S0168-9274(02)00116-2. http://dl.acm.org/citation.cfm?id=765737.765740
3. Bechtold, T., Verhoeven, A., ter Maten, E.J.W., Voss, T.: Model order reduction: an advanced, efficient and automated computational tool for microsystems. In: Cutello, V., Fotia, G., Puccio, L. (eds.) Applied and Industrial Mathematics in Italy II: Selected Contributions from the 8th SIMAI Conference, Baia Samuele. Series on Advances in Mathematics for Applied Sciences, vol. 75, pp. 113–124. World Scientific, Singapore (2007)
4. Benner, P., Quintana-Ortí, E.: Solving stable generalized Lyapunov equations with the matrix sign function. Numer. Algorithms **20**, 75–100 (1999). http://dx.doi.org/10.1023/A:1019191431273. 10.1023/A:1019191431273
5. Benner, P., Sorensen, D.C., Mehrmann, V. (eds.): Dimension Reduction of Large-Scale Systems: Proceedings Workshop Oberwolfach, Germany, 2003. Lecture Notes in Computational Science and Engineering, vol. 45, pp. 19–25. Springer, Berlin/Heidelberg (2005)
6. Benner, P., Schneider, A.: On stability, passivity and reciprocity preservation of ESVDMOR. In: Benner, P., Hinze, M., ter Maten, E.J.W. (eds.) Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74, pp. 277–287. Springer, Dordrecht (2011)
7. 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)
8. Deschrijver, D.: Broadband macromodeling of linear systems by vector fitting. Ph.D. thesis, Universiteit Antwerpen (2007)
9. Feldmann, P., Freund, R.W.: Efficient linear circuit analysis by Padé approximation via the Lanczos process. In: Proceedings of the Conference on European Design Automation (EURO-DAC'94), Grenoble, pp. 170–175. IEEE Computer Society Press, Los Alamitos (1994). http://dl.acm.org/citation.cfm?id=198174.198236
10. Ferranti, F.: Parameterized macromodeling and model order reduction of high-speed interconnects. Ph.D. thesis Ghent University (2011)
11. Ferranti, F., Deschrijver, D., Knockaert, L., Dhaene, T.: Data-driven parameterized model order reduction using z-domain multivariate orthonormal vector fitting technique. In: Benner, P., Hinze, M., ter Maten, E.J.W. (eds.) Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74, pp. 141–148. Springer, Dordrecht (2011)
12. Freund, R.W.: Krylov-subspace methods for reduced-order modeling in circuit simulation. J. Comput. Appl. Math. **123**(1–2), 395–421 (2000). http://linkinghub.elsevier.com/retrieve/pii/S0377042700003964
13. Freund, R.W.: Model reduction methods based on Krylov subspaces. Acta Numer. **12**, 267–319 (2003)
14. Freund, R.W.: SPRIM: structure-preserving reduced-order interconnect macromodeling. In: Proceedings of the 2004 International Conference on Computer-Aided Design (ICCAD'04), San Jose, 7–11 Nov 2004, pp. 80–87. IEEE Computer Society/ACM (2004). doi:http://doi.acm.org/10.1145/1112239.1112258
15. Freund, R.W.: Structure-preserving moder order reduction of RLC circuit equations. In: 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, pp. 49–74. Springer, Berlin (2008)
16. Freund, R.W.: The SPRIM algorithm for structure-preservation order reduction of general RCL circuits. In: Benner, P., Hinze, M., ter Maten, E.J.W. (eds.) Model Reduction for Circuit

Simulation. Lecture Notes in Electrical Engineering, vol. 74, pp. 25–52. Springer, Dordrecht (2011)

17. Freund, R.W.: Recent advances in structure-preserving model order reduction. In: Li, P., Silveira, L.M., Feldmann, P. (eds.) Simulation and Verification of Electronic and Biological Systems, pp. 43–70. Springer, Dordrecht/ Heidelberg/London/New York (2011)
18. Freund, R.W., Feldmann, P.: Reduced-order modeling of large passive linear circuits by means of the SyPVL algorithm. In: Proceedings of the 1996 IEEE/ACM International Conference on Computer-Aided Design (ICCAD'96), San Jose, pp. 280–287. IEEE Computer Society, Washington, DC (1996). http://dl.acm.org/citation.cfm?id=244522.244571
19. Gallivan, K., Grimme, E., Van Dooren, P.: Asymptotic waveform evaluation via a Lanczos method. Appl. Math. Lett. **7**(5), 75–80 (1994). doi:10.1016/0893-9659(94)90077-9. http://www.sciencedirect.com/science/article/pii/0893965994900779
20. Green, M., Limebeer, D.: Linear Robust Control. Prentice-Hall Information and System Sciences Series. Prentice Hall, Englewood Cliffs (1995). http://books.google.de/books?id=8NdSAAAAMAAJ
21. Grimme, E.J.: Krylov projection methods for model reduction. Ph.D. thesis, University of Illinois at Urbana-Champaign (1997)
22. Günther, M., Feldmann, U., ter Maten, E.J.W.: Modelling and discretization of circuit problems. In: W.H.A. Schilders, E.J.W. ter Maten (eds.) Handbook of Numerical Analysis. Special Volume Numerical Analysis of Electromagnetism, vol. XIII, pp. 523–659. Elsevier/North Holland, Amsterdam (2005). doi:10.1016/s1570-8659(04)13006-8
23. Gugercin, S., Antoulas, A.C., Beattie, C.: H2 model reduction for large-scale linear dynamical systems. SIAM J. Matrix Anal. Appl. **30**(2), 609–638 (2008)
24. Hairer, E., Wanner, G.: Solving Ordinary Differential Equations II – Stiff and Differential-Algebraic Problems, 2nd rev. edn. Springer, Berlin (1996)
25. Honkala, M.: Building blocks for fast circuit simulation. Ph.D. thesis, Aalto University (2013)
26. Ionescu, T.C., Astolfi, A.: Families of moment matching based, structure preserving approximations for linear port Hamiltonian systems. Automatica **49**(8), 2424–2434 (2013)
27. Ionutiu, R., Rommes, J.: "A framework for synthesis of reduced order models". In: Coupled Multiscale Simulation and Optimization in Nanoelectronics, Ch. 3, Section 4.5, this volume (2015)
28. Kailath, T.: Linear Systems. Prentice-Hall Information and System Sciences Series. Prentice Hall International, Singapore (1998). http://books.google.de/books?id=MAfiOgAACAAJ
29. Kunkel, P., Mehrmann, V.: Differential-Algebraic Equations: Analysis and Numerical Solution. EMS Textbooks in Mathematics. European Mathematical Society, Zürich (2006)
30. Li, J., White, J.: Low rank solution of Lyapunov equations. SIAM J. Matrix Anal. Appl. **24**(1), 260–280 (2002). doi:10.1137/S0895479801384937. http://epubs.siam.org/doi/abs/10.1137/S0895479801384937
31. Martins, N., Lima, L., Pinto, H.: Computing dominant poles of power system transfer functions. IEEE Trans. Power Syst. **11**(1), 162–170 (1996)
32. Miettinen, P., Honkala, M., Roos, J.: PARTMOR: partitioning-based RL-in-RL-out MOR method. In: Roos, J., Costa, L.R.G. (eds.) Scientific Computing in Electrical Engineering (SCEE 2008), Espoo. Mathematics in Industry, vol. 14, pp. 547–594. Springer (2010)
33. Mohaghegh, K.: Linear and nonlinear model order reduction for numerical simulation of electric circuits. Ph.D. thesis, Bergische Universität Wuppertal. Logos Verlag, Berlin (2010)
34. 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), Espoo. Mathematics in Industry, vol. 14, pp. 563–570. Springer (2010)
35. Moore, B.: Principal component analysis in linear systems: controllability, observability, and model reduction. IEEE Trans. Autom. Control **26**(1), 17–32 (1981)
36. Odabasioglu, A., Celik, M., Pileggi, L.T.: PRIMA: passive reduced-order interconnect macromodeling algorithm. In: Proceedings of the 1997 IEEE/ACM international conference

on Computer-Aided Design (ICCAD'97), San Jose, pp. 58–65. IEEE Computer Society, Washington, DC (1997). http://dl.acm.org/citation.cfm?id=266388.266423

37. Phillips, J.R., Silveira, L.M.: Poor Man's TBR: a simple model reduction scheme. IEEE Trans. Comput.-Aided Des. Circuits Syst. **24**(1), 43–55 (2005)
38. Pillage, L., Rohrer, R.: Asymptotic waveform evaluation for timing analysis. IEEE Trans. Comput.-Aided Des. Circuits Syst. **9**(4), 352–366 (1990). doi:10.1109/43.45867
39. Pinnau, R.: Model reduction via proper orthogonal decomposition. In: 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, pp. 95–109. Springer, Berlin (2008)
40. Polyuga, R.V.: Model reduction of port-Hamiltonian systems. Ph.D. thesis, Rijksuniversiteit Groningen (2010). http://dissertations.ub.rug.nl/faculties/science/2010/r.v.polyuga/
41. Polyuga, R.V., van der Schaft, A.J.: Structure preserving port-Hamiltonian model reduction of electrical circuits. In: Benner, P., Hinze, M., ter Maten, E.J.W. (eds.) Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74, pp. 241–260. Springer, Dordrecht (2011)
42. Pulch, R., ter Maten, E.J.W., Augustin, F.: Sensitivity analysis and model order reduction for random linear dynamical systems. CASA-Report 2013–15, TU Eindhoven, http://www.win.tue.nl/analysis/reports/rana13-15.pdf. IMACM-Report 2013–7, Bergische Universität Wuppertal (2013)
43. Rathinam, M., Petzold, L.R.: A new look at proper orthogonal decomposition. SIAM J. Numer. Anal. **41**, 1893–1925 (2003)
44. Reis, T.: Circuit synthesis of passive descriptor systems—a modified nodal approach. Int. J. Circuit Theory Appl. **38**(1), 44–68 (2010). doi:10.1002/cta.532. http://dx.doi.org/10.1002/cta.532
45. Rommes, J.: Modal approximation and computation of dominant poles. In: 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, pp. 177–193. Springer, Berlin (2008)
46. Rommes, J., Martins, N.: Efficient computation of multivariable transfer function dominant poles using subspace acceleration. IEEE Trans. Power Syst. **21**(4), 1471–1483 (2006)
47. Roos, J., Honkala, M., Miettinen, P.: GABOR: Global-approximation-based order reduction. In: In: Roos, J., Costa, L.R.G. (eds.) Scientific Computing in Electrical Engineering (SCEE 2008), Espoo. Mathematics in Industry, vol. 14, pp. 507–514. Springer (2010)
48. Saadvandi, M.: Passivity preserving model reduction and selection of spectral zeros. MSc-thesis, Royal Institute of Technology KTH, Stockholm. Also published as Technical Note NXP-TN-2008/00276, Unclassified Report, NXP Semiconductors, Eindhoven (2008)
49. Saak, J., Benner, P.: Efficient solution of large scale Lyapunov and Riccati equations arising in model order reduction problems. Proc. Appl. Math. Mech. (PAMM) **8**(1), 10085–10088 (2008). doi:10.1002/pamm.200810085. http://dx.doi.org/10.1002/pamm.200810085
50. Schilders, W.H.A.: Introduction to Model Order Reduction. In: 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, pp. 3–32. Springer, Berlin (2008)
51. 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)
52. Schilders, W.H.A.: The need for novel model order reduction techniques in the electronics industry. In: Benner, P., Hinze, M., ter Maten, E.J.W. (eds.) Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74, pp. 3–24. Springer, Dordrecht (2011)
53. Sorensen, D.: Passivity preserving model reduction via interpolation of spectral zeros. Syst. Control Lett. **54**(4), 347–360 (2005)
54. Stykel, T.: Gramian-based model reduction for descriptor systems. Math. Control Signals Syst. (MCSS) **16**(4), 297–319 (2004)
55. Vollebregt, A.J., Bechtold, T., Verhoeven, A., ter Maten, E.J.W.: Model order reduction of large ODE systems: MOR for Ansys versus ROM Workbench. In: Ciuprina, G., Ioan, D. (eds.)

Scientific Computing in Electrical Engineering. Mathematics in Industry, vol. 11, pp. 175–182. Springer, Berlin/New York (2007)

56. Willcox, K., Peraire, J., White, J.: An Arnoldi approach for generation of reduced-order models for turbomachinery. Comput. Fluids **31**(3), 369–389 (2002)

References for Section 4.2

57. Aguirre, L.A.: Quantitative measure of modal dominance for continuous systems. In: Proceedings of the 32nd Conference on Decision and Control, San Antonio, pp. 2405–2410 (1993)
58. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems. SIAM, Philadelphia (2005)
59. Bai, Z., Su, Y.: SOAR: a second-order Arnoldi method for the solution of the quadratic eigenvalue problem. SIAM J. Matrix Anal. Appl. **26**(3), 640–659 (2005)
60. 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)
61. 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)
62. Fokkema, D.R., Sleijpen, G.L.G., van der Vorst, H.A.: Jacobi-Davidson style QR and QZ algorithms for the reduction of matrix pencils. SIAM J. Sci. Comput. **20**(1), 94–125 (1998)
63. Golub, G.H., van Loan, C.F.: Matrix Computations, 3rd edn. John Hopkins University Press, Baltimore (1996)
64. Green, M., Limebeer, D.J.N.: Linear Robust Control. Prentice-Hall, Englewood Cliffs (1995)
65. Grimme, E.J.: Krylov projection methods for model reduction. Ph.D. thesis, University of Illinois (1997)
66. Haley, S.B.: The generalized eigenproblem: pole-zero computation. Proc. IEEE **76**(2), 103–120 (1988)
67. Hochstenbach, M.E., Sleijpen, G.L.G.: Two-sided and alternating Jacobi-Davidson. Linear Algebra Appl. **358**(1–3), 145–172 (2003)
68. Kailath, T.: Linear Systems. Prentice-Hall, Englewood Cliffs (1980)
69. Knyazev, A.V.: Preconditioned eigensolvers. In: Bai, Z., Demmel, J., Dongarra, J., Ruhe, A., van der Vorst, H. (eds.) Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, pp. 337–368. SIAM, Philadelphia (2000)
70. Knyazev, A.V.: Hard and soft locking in iterative methods for symmetric eigenvalue problems. In: Copper Mountain Conference, Copper Mountain (2004). http://math.cudenver.edu/~aknyazev/research/conf/cm04.htm
71. Lampe, J., Voss, H.: Second order Arnoldi reduction: application to some engineering problems. Report 93, Hamburg University of Technology (2006)
72. Martins, N., Lima, L.T.G., Pinto, H.J.C.P.: Computing dominant poles of power system transfer functions. IEEE Trans. Power Syst. **11**(1), 162–170 (1996)
73. Martins, N., Pellanda, P.C., Rommes, J.: Computation of transfer function dominant zeros with applications to oscillation damping control of large power systems. IEEE Trans. Power Syst. **22**(4), 1218–1226 (2007)
74. Martins, N., Quintão, P.E.M.: Computing dominant poles of power system multivariable transfer functions. IEEE Trans. Power Syst. **18**(1), 152–159 (2003)
75. Murthy, D.V., Haftka, R.T.: Derivatives of eigenvalues and eigenvectors of a general complex matrix. Int. J. Numer. Methods Eng. **26**, 293–311 (1988)
76. Ostrowski, A.M.: On the convergence of the Rayleigh quotient iteration for the computation of the characteristic roots and vectors. I. Arch. Ration. Mech. Anal. **1**, 233–241 (1958)
77. Parlett, B.N.: The Rayleigh quotient iteration and some generalizations for nonnormal matrices. Math. Comput. **28**(127), 679–693 (1974)

78. Parlett, B.N.: The Symmetric Eigenvalue Problem. Classics in Applied Mathematics. SIAM, Philadelphia (1998)
79. Peters, G., Wilkinson, J.H.: Inverse iteration, ill-conditioned equations and Newton's method. SIAM Rev. **21**(3), 339–360 (1979)
80. 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
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., Martins, N.: Efficient computation of transfer function dominant poles using subspace acceleration. IEEE Trans. Power Syst. **21**(3), 1218–1226 (2006)
83. Rommes, J., Martins, N.: Computing large-scale system eigenvalues most sensitive to parameter changes, with applications to power system small-signal stability. IEEE Trans. Power Syst. **23**(4), 434–442 (2008)
84. Rommes, J., Martins, N.: Efficient computation of transfer function dominant poles of large second-order dynamical systems. SIAM J. Sci. Comput. **30**(4), 2137–2157 (2008)
85. Rommes, J., Sleijpen, G.L.G.: Convergence of the dominant pole algorithm and Rayleigh quotient iteration. SIAM J. Matrix Anal. Appl. **30**(1), 346–363 (2008)
86. Ruhe, A.: Rational Krylov: a practical algorithm for large sparse nonsymmetric matrix pencils. SIAM J. Sci. Comput. **19**(5), 1535–1551 (1998)
87. Saad, Y.: Numerical Methods for Large Eigenvalue Problems: Theory and Algorithms. Manchester University Press, Manchester (1992)
88. 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)
89. Sleijpen, G.L.G., van der Vorst, H.A.: A Jacobi-Davidson iteration method for linear eigenvalue problems. SIAM J. Matrix Anal. Appl. **17**(2), 401–425 (1996)
90. Smith, J.R., Hauer, J.F., Trudnowski, D.J., Fatehi, F., Woods, C.S.: Transfer function identification in power system application. IEEE Trans. Power Syst. **8**(3), 1282–1290 (1993)
91. Stathopoulos, A.: A case for a biorthogonal Jacobi-Davidson method: restarting and correction equation. SIAM J. Matrix Anal. Appl. **24**(1), 238–259 (2002)
92. Matlab, available from The MathWorks, Inc., Natick, MA, 01760-2098 USA, http://www.mathworks.com.
93. van der Vorst, H.A.: Computational Methods for Large Eigenvalue Problems. In: Ciarlet, P.G., Lions, J.L. (eds.) Handbook of Numerical Analysis, vol. VIII, pp. 3–179. North-Holland (Elsevier), Amsterdam (2001)
94. Varga, A.: Enhanced modal approach for model reduction. Math. Model. Syst. **1**, 91–105 (1995)
95. Wang, L., Howell, F., Kundur, P., Chung, C.Y., Xu, W.: A tool for small-signal security assessment of power systems. In: Proceedings of the IEEE International Conference on Power Industry Computer Applications PICA 2001, Sydney, Australia, pp. 246–252 (2001)

References for Section 4.3

96. Antoulas, A.C.: A new result on passivity preserving model reduction. Syst. Control Lett. **54**(4), 361–374 (2005)
97. Bai, Z., Freund, R.: Eigenvalue-based characterization and test for positive realness of scalar transfer functions. IEEE Trans. Autom. Control **AC-45**, 2396–2402 (2000)
98. Bai, Z., Feldmann, P., Freund, R.: Stable and passive reduced-order model based on partial Padè approximation via the Lanczos process. Bell Laboratories, Lucent Technologies Numerical Analysis Manuscript 97/3-10 (1997)

99. Bai, Z., Feldmann, P., Freund, R.: How to make theoretically passive reduced-order models passive in practice. In: Proceeding of the IEEE 1998 Custom Integrated Circuits Conference, Santa Clatra, pp. 207–210 (1998)
100. Feldman, P., Freund, R.W.: Efficient linear circuit analysis by Padè approximation via a Lanczos method. IEEE Trans. Comput.-Aided Des. **14**, 639–649 (1995)
101. Freund, R.: Passive reduced-order models for interconnect simulation and their computation via Krylov-subspace algorithms. In: Proceedings of the ACM DAC'99, New Orleans (1999)
102. Gohberg, I., Lancaster, P., Rodman, L.: Invariant subspace of matrices with applications. SIAM, Philadelphia (1986)
103. Grimme, E.J.: Krylov projection methods for model reduction. Ph.D. thesis, University of Illinois, Urbana-Champaign (1997)
104. Gugercin, S., Antoulas, A.C.: On balancing related model reduction methods and the corresponding error. Int. J. Control **77**(8), 748–766 (2004)
105. Nong, Hung Dinh: Passivity preserving model reduction via interpolation of spectral zeros: selection criteria and implementation. Master thesis, Rice University (2007)
106. Ober, R.: Balanced parametrization of classes of linear systems. SIAM J. Control Optim. **29**, 1251–1287 (1991)
107. Odabasioglu, A., Celik, M., Pileggi, L.T.: PRIMA: passive reduced-order interconnect macromodeling algorithm. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **17**, 645–654 (1998)
108. Saadvandi, M.: Passivity preserving model reduction and selection of spectral zeros. MSc-thesis, Royal Institute of Technology KTH, Stockholm. Also published as Technical Note NXP-TN-2008/00276, Unclassified Report, NXP Semiconductors, Eindhoven (2008)
109. Saadvandi, M.: Nonlinear and parametric model order reduction for second order dynamical systems by the dominant pole algorithm. Ph.D. thesis, KU Leuven (2013)
110. Sorensen, D.C.: Passivity preserving model reduction via interpolation of spectral zeros. Syst. Control Lett. **54**(4), 361–374 (2005)
111. De Villemagne, C., Skelton, R.: Model reduction using a projection formulation. Int. J. Control **40**, 2142–2169 (1987)
112. Zhou, K., Doyle, J.C., Glover, K.: Robust and Optimal Control. Prentice Hall, Upper Saddle River (1996)

References for Section 4.4

113. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems. SIAM, Philadelphia (2005)
114. Antoulas, A.C.: A new result on passivity preserving model reduction. Syst. Control Lett. **54**, 361–374 (2005)
115. Bai, Z., Li, R., Su, Y.: A unified Krylov projection framework for structure-preserving model reduction. In: Schilders, W., van der Vorst, H., Rommes, J. (eds.) Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 13, pp. 75–93. Springer, Berlin (2008)
116. 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)
117. Freund, R.: SPRIM: Structure-preserving reduced-order interconnect macromodeling. In: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD'04), San Jose, pp. 80–87. Los Alamitos (2004)
118. Freund, R.: Structure preserving model order reduction of RCL circuit equations. In: Schilders, W., van der Vorst, H., Rommes, J. (eds.) Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 13, pp. 49–73. Springer, Berlin (2008)
119. Guillemin, E.A.: Synthesis of Passive Networks, 2nd edn. Wiley, New York/London (1959)

120. Ionutiu, R.: Passivity preserving model reduction in the context of spectral zero interpolation. Master's thesis, William Marsh Rice University, Houston (2008)
121. 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
122. Ionutiu, R., Rommes, J.: Circuit synthesis of reduced order models. NXP-TN 2008/00316, NXP Semiconductors (2008)
123. Ionutiu, R., Rommes, J.: "A framework for synthesis of reduced order models". In: Coupled Multiscale Simulation and Optimization in Nanoelectronics, Ch. 3, Section 4.5, this volume (2013)
124. Ionutiu, R., Rommes, J.: "Model order reduction for multi-terminal circuits". In: Coupled Multiscale Simulation and Optimization in Nanoelectronics, Ch. 5, Section 6.2, this volume (2013)
125. Ionutiu, R., Rommes, J., Antoulas, A.: Passivity preserving model reduction using dominant spectral zero interpolation. IEEE Trans. Comput.-Aided Des. Circuits Syst. **27**(12), 2250–2263 (2008)
126. Kailath, T.: Linear Systems. Prentice-Hall, Englewood Cliffs (1980)
127. Kressner, D.: Numerical methods for general and structured eigenvalue problems. Lecture Notes in Computational Science and Engineering, vol. 46. Springer, Berlin (2005)
128. Odabasioglu, A., Celik, M., Pillegi, L.: Prima: Passive reduced-order interconnect macromodelling algorithm. IEEE Trans. Comput.-Aided Des. Circuits Syst. **17**, 645–654 (1998)
129. Phillips, J., Daniel, L., Silveira, L.: Guaranteed passive balancing transformations for model order reduction. IEEE Trans. Comput.-Aided Des. Circuits Syst. **22**(8), 1027–1041 (2003)
130. Rommes, J.: Methods for eigenvalue problems with applications in model order reduction. Ph.D. thesis, Utrecht University, Utrecht (2007). http://sites.google.com/site/rommes
131. Rommes, J., Martins, N.: Efficient computation of transfer function dominant poles using subspace acceleration. IEEE Trans. Power Syst. **21**(3), 1218–1226 (2006)
132. Rommes, J., Martins, N.: Efficient computation of multivariable transfer function dominant poles using subspace acceleration IEEE Trans. Power Syst. **21**(4), 1471–1483 (2006)
133. Sheldon, X.D.T., He, L.: Advanced Model Order Reduction Techniques in VLSI design. Cambridge University Press, Cambridge (2007)
134. Sorensen, D.: Passivity preserving model reduction via interpolation of spectral zeros. Syst. Control Lett. **54**, 347–360 (2005)
135. Varga, A.: Enhanced modal approach for model reduction. Math. Model. Syst. **1**, 91–105 (1995)
136. Verbeek, M.E.: Partial element equivalent circuit (PEEC) models for on-chip passives and interconnects. Int. J. Numer. Model. Electron. Netw. Devices Fields **17**(1), 61–84 (2004)
137. Watkins, D.S.: The Matrix Eigenvalue Problem: GR and Krylov Subspace Methods. SIAM, Philadelphia (2007)
138. 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)
139. You, H., He, L.: A sparsified vector potential equivalent circuit model for massively coupled interconnects. In: IEEE International Symposium on Circuits and Systems (ISCAS), Kobe, Japan, vol. 1, pp. 105–108 (2005)
140. Zheng, H., Pileggi, L.: Robust and passive model order reduction for circuits containing susceptance elements. In: Proceedings of the IEEE/ACM International Conference on Computer Aided Design (ICCAD'02), San Jose, pp. 761–766 (2002)

References for Section 4.5

141. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems. SIAM, Philadelphia (2005)
142. Antoulas, A.C.: A new result on passivity preserving model reduction. Syst. Control Lett. **54**, 361–374 (2005)
143. Bai, Z., Li, R., Su, Y.: A unified Krylov projection framework for structure-preserving model reduction. In: Schilders, W., van der Vorst, H., Rommes, J. (eds.) Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 13, pp. 75–93. Springer, Berlin (2008)
144. Benner, P.: Advances in balancing-related model reduction for circuit simulation. In: Roos, J., Costa, L.R.G. (eds.) Scientific Computing in Electrical Engineering (SCEE 2008), Espoo. Mathematics in Industry, vol. 14, pp. 469–482. Springer (2010)
145. Benner, P., Mehrmann, V., Sorensen, D. (eds.): Dimension Reduction of Large-Scale Systems. Lecture Notes in Computational Science and Engineering, vol. 45. Springer, Berlin (2005)
146. Benner, P., Hinze, M., ter Maten, E.J.W. (eds.): Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74. Springer, Berlin (2011)
147. Bott, R., Duffin, R.: Impedance synthesis without the use of transformers. J. Appl. Phys. **20**, 816 (1949)
148. Brune, O.: Synthesis of a finite two-terminal network whose driving point impedance is a prescribed function of frequency. J. Math. Phys. **10**, 191–236 (1931)
149. Jivaro, available at EdXact SA, Voiron, France, http://www.edxact.com
150. The Spice Home Page, http://bwrcs.eecs.berkeley.edu/Classes/IcBook/SPICE/, EECS Dept., Univ. of California at Berkeley, CA 94720-1770, USA.
151. Freund, R.: SPRIM: structure-preserving reduced-order interconnect macromodeling. In: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD’04), San Jose, pp. 80–87. Los Alamitos (2004)
152. Grimme, E.J.: Krylov projection methods for model reduction. Ph.D. thesis, University of Illinois (1997)
153. Guille, A., Hanssen, M., Niehof, J.: Comparison between RLCk extraction and EM simulation on RF circuits. Tech. rep., NXP Semiconductors (2008)
154. Guillemin, E.A.: Synthesis of Passive Networks, 2nd edn. Wiley, New York/London (1959)
155. Heres, P.J.: Robust and efficient Krylov subspace methods for model order reduction. Ph.D. thesis, Eindhoven University of Technology (2005)
156. 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.
157. Ionutiu, R., Rommes, J., Antoulas, A.: Passivity preserving model reduction using dominant spectral zero interpolation. IEEE Trans. Comput.-Aided Des. Circuits Syst. **27**(12), 2250–2263 (2008)
158. Ionutiu, R., Rommes, J.: Circuit synthesis of reduced order models. Technical Note 2008/00316, NXP Semiconductors (2009)
159. Ionutiu, R., Rommes, J.: On synthesis of reduced order models. In: Benner, P., Hinze, M., ter Maten, E.J.W. (eds.) Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74, pp. 207–224. Springer, Berlin (2011)
160. Ionutiu, R., Rommes, J.: “Model order reduction for multi-terminal circuits”. In: Coupled Multiscale Simulation and Optimization in Nanoelectronics, Ch. 5, Section 6.2, this volume (2013)
161. 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. 3, Section 4.4, this volume (2013)
162. Kailath, T.: Linear Systems. Prentice-Hall, Englewood Cliffs (1980)
163. 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. **17**(7), 582–591 (1998)

164. Miettinen, P., Honkala, M., Roos, J., Valtonen, M.: PARTMOR: Partitioning-based realizable model-order reduction method for RLC circuits. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **30**(3), 374–387 (2011)
165. Pstar, in-house industrial circuit simulator, NXP Semiconductors, Eindhoven, The Netherlands, http://www.nxp.com
166. Odabasioglu, A., Celik, M., Pileggi, T.: PRIMA: passive reduced-order interconnect macromodeling algorithm. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **17**(8), 645–654 (1998)
167. Palenius, T., Roos, J.: Comparison of reduced-order interconnect macromodels for time-domain simulation. IEEE Trans. Microw. Theory Tech. **52**(9), 191–236 (2004)
168. Phillips, J.R., Silveira, L.M.: Poor Man's TBR: a simple model reduction scheme. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **24**(1), 283–288 (2005)
169. Phillips, J.R., Daniel, L., Silveira, L.: Guaranteed passive balancing transformations for model order reduction. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **22**(8), 1027–1041 (2003). doi:10.1109/TCAD.2003.814949
170. Reis, T.: Circuit synthesis of passive descriptor systems – a modified nodal approach. Int. J. Circuit Theory Appl. **38**(1), 44–68 (2010)
171. Rommes, J.: Methods for eigenvalue problems with applications in model order reduction. Ph.D. thesis, Utrecht University (2007)
172. Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.): Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 11. Springer, Berlin (2008)
173. Sorensen, D.: Passivity preserving model reduction via interpolation of spectral zeros. Syst. Control Lett. **54**, 347–360 (2005)
174. Stykel, T., Reis, T.: Passivity-preserving model reduction of differential-algebraic equations in circuit simulation. In: Proceedings in Applied Mathematics and Mechanics (ICIAM'07), Zurich, pp. 1021601–1021602 (2007)
175. Tan, S.X.D., He, L.: Advanced model order reduction techniques in VLSI design. Cambridge University Press, Cambridge (2007)
176. 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)