Chapter 6

Efficient computation of transfer function dominant poles of large second-order dynamical systems

Abstract. This chapter presents a new algorithm for the computation of dominant poles of transfer functions of large-scale second-order dynamical systems: the Quadratic Dominant Pole Algorithm (QDPA). The algorithm works directly with the system matrices of the original system, so no linearization is needed. To improve global convergence, QDPA uses subspace acceleration, and deflation of found dominant poles is implemented in a very efficient way. The dominant poles and corresponding eigenvectors can be used to construct structure-preserving modal approximations, but also to improve reduced-order models computed by Krylov subspace methods, as is illustrated by numerical results. Generalizations to MIMO systems, higher-order systems and the computation of dominant zeros are also described.

Key words. poles, transfer function, quadratic eigenvalue problem, second-order dynamical systems, small-signal stability, transfer function residues, model reduction, large-scale systems, sparse eigenanalysis, modal equivalents, modal analysis, second-order Arnoldi

6.1 Introduction

A transfer function usually reflects the engineer's interest in studying a given part of a large dynamical system, and often only has a small number of dominant poles compared to the number of state variables. The dominant behavior of the system can be captured by projecting the state-space on the modes corresponding to the dominant poles. This type of model reduction is known as modal approximation, see for instance [159]. The computation of the dominant poles and modes requires specialized eigenvalue methods.

In [91] Newton's method is used to compute a dominant pole of single-input single-output (SISO) transfer function: the Dominant Pole Algorithm (DPA). In two recent publications this algorithm was improved and extended to a robust and efficient method for the computation of the dominant poles and corresponding eigenvectors of large scale SISO [123] (see Chapter 3) and multi-input multi-output (MIMO) transfer functions [122] (see Chapter 4).

In this chapter an efficient algorithm, Quadratic DPA (QDPA), for the computation of dominant poles of second-order transfer functions is presented. It is shown how DPA can be generalized to second and higher order polynomial transfer functions. Furthermore, it is described how subspace acceleration and efficient deflation can be added to obtain a more effective algorithm similar to SADPA. All algorithms presented work directly with the state-space matrices of the higher order system, i.e. no linearization is needed. Moreover, any modal equivalents that are constructed by projecting the state-space matrices on the dominant left and right eigenspaces, preserve the structure of the original system. The modal equivalents are compared to reduced-order models computed by (second-order) Arnoldi methods [12, 11] and it is shown how these models can be improved by using the dominant poles computed by QDPA.

The chapter is organized as follows. Section 6.2 gives definitions and properties of transfer functions and dominant poles, and describes the basic Quadratic DPA algorithm (QDPA). In Section 6.3 subspace acceleration and deflation are added. Numerical results are presented in Section 6.4. Generalizations to higher order polynomial transfer functions and MIMO systems, and computation of dominant zeros, are described in Section 6.5. Section 6.6 concludes.

6.2 Transfer functions and dominant poles

In this chapter, the second-order dynamical systems $(M, C, K, \mathbf{b}, \mathbf{c}, d)$ are of the form

$$\begin{cases} M\ddot{\mathbf{x}}(t) + C\dot{\mathbf{x}}(t) + K\mathbf{x}(t) &= \mathbf{b}u(t) \\ y(t) &= \mathbf{c}^*\mathbf{x}(t) + du(t), \end{cases} \tag{6.2.1}$$

where $M, C, K \in \mathbb{R}^{n\times n}$, $\mathbf{b}, \mathbf{c}, \mathbf{x}(t) \in \mathbb{R}^n$, $u(t), y(t), d \in \mathbb{R}$. In typical applications such as structural system analysis, M is the mass matrix, K is the stiffness matrix and C is the damping matrix. The vectors $\mathbf{b}$ and $\mathbf{c}$ are called the input and output vector, respectively. The transfer function $H : \mathbb{C} \longrightarrow \mathbb{C}$ of (6.2.1) is defined as

$$H(s) = \mathbf{c}^*(s^2M + sC + K)^{-1}\mathbf{b} + d. \tag{6.2.2}$$

The poles of transfer function (6.2.2) are a subset of the eigenvalues $\lambda_i \in \mathbb{C}$ of the quadratic eigenvalue problem

$$(\lambda^2 M + \lambda C + K)\mathbf{x} = 0, \qquad \mathbf{x} \neq 0.$$

Most material on quadratic eigenvalue problems in this section can be found in more detail in [10, 152]. An eigentriplet $(\lambda_i, \mathbf{x}_i, \mathbf{y}_i)$ is composed of an eigenvalue

λ_i and corresponding right and left eigenvectors $\mathbf{x}_i, \mathbf{y}_i \in \mathbb{C}^n$ (identified by their components in $\mathbf{b}$ and $\mathbf{c}$):

$$\begin{cases} (\lambda_i^2 M + \lambda_i C + K)\mathbf{x}_i &= 0, \qquad \mathbf{x}_i \neq 0, \quad (i = 1, \ldots, 2n), \\ \mathbf{y}_i^*(\lambda_i^2 M + \lambda_i C + K) &= 0, \qquad \mathbf{y}_i \neq 0, \quad (i = 1, \ldots, 2n). \end{cases} \tag{6.2.3}$$

The quadratic eigenvalue problem (6.2.3) has $2n$ eigenvalues with at most $2n$ right (and left) eigenvectors. It is clear that $2n$ eigenvectors do not form an independent set in an n-dimensional space. Still, however, it is possible to obtain a partial fraction representation of the transfer function similar to the first order case. The approach followed here is also described in [152, Section 3.5]. If K is nonsingular, the quadratic eigenvalue problem (6.2.3) can be linearized to the generalized eigenproblem

$$A\phi_i = \lambda_i B\phi_i, \quad \psi_i^* A = \lambda_i \psi_i^* B, \quad \phi_i, \psi_i \neq 0, \quad (i = 1, \ldots, 2n),$$

where

$$A = \begin{bmatrix} 0 & -K \\ -K & -C \end{bmatrix}, \text{ and } B = \begin{bmatrix} -K & 0 \\ 0 & M \end{bmatrix}, \tag{6.2.4}$$

and

$$\phi_i = \begin{bmatrix} \mathbf{x}_i \\ \lambda \mathbf{x}_i \end{bmatrix}, \text{ and } \psi_i = \begin{bmatrix} \mathbf{y}_i \\ \bar{\lambda}_i \mathbf{y}_i. \end{bmatrix}$$

The corresponding linearized dynamical system is

$$\begin{cases} B\dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + \mathbf{b}_l u(t) \\ y(t) &= \mathbf{c}_l^* \mathbf{x}(t) + du(t), \end{cases} \tag{6.2.5}$$

where $\mathbf{b}_l = [0, \mathbf{b}^T]^T$ and $\mathbf{c}_l = [\mathbf{c}^T, 0]^T$. Note that also other linearizations are possible, but that this choice preserves symmetry, if M, C, and K are symmetric, and is convenient for other reasons as well, as will become clear in the following. Denoting the matrix polynomial in (6.2.3) by $Q(\lambda) = \lambda^2 M + \lambda C + K$, it can be verified that for nonsingular K one has

$$\begin{bmatrix} Q(\lambda) & 0 \\ 0 & I \end{bmatrix} = E(\lambda)(A - \lambda B)F(\lambda), \tag{6.2.6}$$

where

$$E(\lambda) = \begin{bmatrix} (C + \lambda M)K^{-1} & -I \\ -K^{-1} & 0 \end{bmatrix}, \text{ and } F(\lambda) = \begin{bmatrix} I & 0 \\ \lambda I & I \end{bmatrix},$$

i.e. the linear matrix $A - \lambda B$ is indeed a linearization of $Q(\lambda)$ [152, Section 3.5]. Some matrix manipulation of (6.2.6) gives

$$\begin{aligned} Q(\lambda)^{-1} &= \begin{bmatrix} I & 0 \end{bmatrix} F(\lambda)^{-1}(A - \lambda B)^{-1} E(\lambda)^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix} \\ &= -\begin{bmatrix} I & 0 \end{bmatrix} (A - \lambda B)^{-1} \begin{bmatrix} 0 \\ I \end{bmatrix}. \end{aligned} \tag{6.2.7}$$

Now assume that M and K are nonsingular and that all eigenvalues are semisimple (i.e., for every eigenvalue the algebraic multiplicity is equal to the geometric multiplicity). Let $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_{2n})$ be a diagonal matrix with eigenvalues of (6.2.3), and let $X = [\mathbf{x}_1, \ldots, \mathbf{x}_{2n}]$ and $Y = [\mathbf{y}_1, \ldots, \mathbf{y}_{2n}]$ have as their columns the corresponding right and left eigenvectors, respectively. Because the eigenvalues of $Q(\lambda)$ are semisimple, the eigenvalues of (A, B) are semisimple as well and (A, B) is diagonalizable, and the corresponding matrices with right and left eigenvectors are

$$\Phi = \begin{bmatrix} X \\ X\Lambda \end{bmatrix}, \text{ and } \Psi = \begin{bmatrix} Y \\ Y\Lambda^* \end{bmatrix}.$$

If Φ and Ψ are normalized so that $\Psi^* A \Phi = \Lambda$ and $\Psi^* B \Phi = I$, and if s is not an eigenvalue of (A, B) (or (6.2.3)), then

$$(A - sB)^{-1} = \Phi(\Lambda - sI)^{-1}\Psi^*,$$

and by (6.2.7) it follows that[1]

$$Q(s)^{-1} = X(sI - \Lambda)^{-1}\Lambda Y^*.$$

Finally, the partial fraction representation becomes

$$H(s) = \mathbf{c}^* X(sI - \Lambda)^{-1}\Lambda Y^* \mathbf{b} = \sum_{i=1}^{2n} \frac{R_i}{s - \lambda_i},$$

where the residues are given by

$$R_i = (\mathbf{c}^* \mathbf{x}_i)(\mathbf{y}_i^* \mathbf{b})\lambda_i. \tag{6.2.8}$$

Note that the $\mathbf{x}_i$ and $\mathbf{y}_i$ are scaled so that

$$\begin{bmatrix} \mathbf{y}_i^* & \lambda_i \mathbf{y}_i^* \end{bmatrix} \begin{bmatrix} -K & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} \mathbf{x}_i \\ \lambda_i \mathbf{x}_i \end{bmatrix} = -\mathbf{y}_i^* K \mathbf{x}_i + \lambda_i^2 \mathbf{y}_i^* M \mathbf{x}_i = 1. \tag{6.2.9}$$

Although there are different indices of modal dominance [4, 68, 123, 159], the following will be used in this chapter.

Definition 6.2.1. *A pole λ_i of $H(s)$ with corresponding right and left eigenvectors $\mathbf{x}_i$ and $\mathbf{y}_i$ ($-\mathbf{y}_i^* K \mathbf{x}_i + \lambda_i^2 \mathbf{y}_i^* M \mathbf{x}_i = 1$) is called* the *dominant pole if*

$$\widehat{R}_i \equiv \frac{|R_i|}{Re(\lambda_i)} > \widehat{R}_j$$

for all $j \neq i$.

More generally, a pole λ_i is called dominant if $|\widehat{R}_i|$ is not very small compared to $|\widehat{R}_j|$, for all $j \neq i$. This can also be seen in the corresponding Bode-plot, which is a plot of $|H(i\omega)|$ against $\omega \in \mathbb{R}$: peaks occur at frequencies ω close to the imaginary

[1] Note that Λ is missing in equation (3.11) in [152, Section 3.5].

parts of the dominant poles of $H(s)$. An approximation of $H(s)$ that consists of $k < 2n$ terms with $|R_j|$ above some value, determines the effective transfer function behavior [144] and is also known as transfer function modal equivalent:

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

Modal equivalents and reduced-order models that are constructed by projecting the state-space matrices on the dominant left and right eigenvectors, preserve the structure of the original system: if X and Y are $n \times k$ matrices with $k \ll n$ right and left eigenvectors corresponding to dominant poles, then the reduced-order model

$$\left\{ \begin{array}{lcl} M_r \ddot{\mathbf{x}}_r(t) + C_r \dot{\mathbf{x}}_r(t) + K_r \mathbf{x}_r(t) & = & \mathbf{b}_r u(t) \\ y_r(t) & = & \mathbf{c}_r^* \mathbf{x}_r(t) + du(t), \end{array} \right. \tag{6.2.10}$$

with $M_r = Y^*MX, C_r = Y^*CX, K_r = Y^*KX \in \mathbb{C}^{k \times k}$ and $\mathbf{b}_r = Y^*\mathbf{b}, \mathbf{c}_r = X^*\mathbf{c} \in \mathbb{C}^k$ is still a second-order system. 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 eigenspaces 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 formed by the columns of X_r and Y_r for the eigenspaces are still (at most) k dimensional. The real reduced model can be formed by using X_r and Y_r in (6.2.10). Preserving the second-order structure is a desirable property from the modeling viewpoint, but also from the viewpoint of generating realizable reduced-order models (see also [55]). Note that although the k dominant poles are also poles of the original transfer function, the k-th order modal equivalent will have k other poles as well, since it is a second-order system.

The dominant poles are specific (complex) eigenvalues of the QEP (6.2.3) and usually form a small subset of the spectrum, so that rather accurate modal equivalents are possible for $k \ll n$. The dominant poles can be located anywhere in the spectrum, depending on the components of the corresponding eigenvectors in $\mathbf{b}$ and $\mathbf{c}$ [126] (see Chapter 2). Since the dominance of a pole is independent of d, $d = 0$ in the following without loss of generality.

6.2.1 The Quadratic Dominant Pole Algorithm (QDPA)

The poles of transfer function (6.2.2) are the $\lambda \in \mathbb{C}$ for which $\lim_{s \to \lambda} |H(s)| = \infty$. Consider now the function $G : \mathbb{C} \longrightarrow \mathbb{C}$

$$G(s) = \frac{1}{H(s)}.$$

For a pole λ of $H(s)$, $\lim_{s \to \lambda} G(s) = 0$. In other words, the poles are the roots of $G(s)$ and a good candidate to find these roots is Newton's method. This idea is the basis of the Dominant Pole Algorithm (DPA) [91] for first order transfer functions,

but can be generalized to transfer functions of any order (and to MIMO systems, see [93, 122]).

The derivative of $G(s)$ with respect to s is given by

$$G'(s) = -\frac{H'(s)}{H^2(s)}. \tag{6.2.11}$$

The derivative of $H(s) = \mathbf{c}^*(s^2M + sC + K)^{-1}\mathbf{b}$ with respect to s is

$$H'(s) = -\mathbf{c}^*(s^2M + sC + K)^{-1}(2sM + C)(s^2M + sC + K)^{-1}\mathbf{b}. \tag{6.2.12}$$

Equations (6.2.11) and (6.2.12) lead to the following Newton scheme:

$$\begin{aligned} s_{k+1} &= s_k - \frac{G(s_k)}{G'(s_k)} \\ &= s_k + \frac{1}{H(s_k)}\frac{H^2(s_k)}{H'(s_k)} \\ &= s_k - \frac{\mathbf{c}^*(s_k^2M + s_kC + K)^{-1}\mathbf{b}}{\mathbf{c}^*(s_k^2M + s_kC + K)^{-1}(2s_kM + C)(s_k^2M + s_kC + K)^{-1}\mathbf{b}} \\ &= s_k - \frac{\mathbf{c}^*\mathbf{v}}{\mathbf{w}^*(2s_kM + C)\mathbf{v}}. \end{aligned} \tag{6.2.13}$$

where $\mathbf{v} = (s_k^2M+s_kC+K)^{-1}\mathbf{b}$ and $\mathbf{w} = (s_k^2M+s_kC+K)^{-*}\mathbf{c}$. An implementation of this Newton scheme, called QDPA, is represented in Alg. 6.1.

Algorithm 6.1 Quadratic Dominant Pole Algorithm (QDPA)

INPUT: System $(M, C, K, \mathbf{b}, \mathbf{c})$, initial pole estimate s_0, tolerance $\epsilon \ll 1$

OUTPUT: Approximate dominant pole λ 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 $(s_k^2M + s_kC + K)\mathbf{v}_k = \mathbf{b}$

4: Solve $\mathbf{w}_k \in \mathbb{C}^n$ from $(s_k^2M + s_kC + K)^*\mathbf{w}_k = \mathbf{c}$

5: Compute the new pole estimate

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

6: The pole $\lambda = s_{k+1}$ with $\mathbf{x} = \mathbf{v}_k/\|\mathbf{v}_k\|_2$ and $\mathbf{y} = \mathbf{w}_k/\|\mathbf{w}_k\|_2$ has converged if

$$\|(s_{k+1}^2M + s_{k+1}C + K)\mathbf{x}\|_2 < \epsilon$$

7: Set $k = k + 1$

8: **end while**

The two linear systems that need to be solved in step 3 and 4 of Alg. 6.1 can be efficiently solved using one LU-factorization $LU = s_k^2 M + s_k C + K$, by noting that $U^* L^* = (s_k^2 M + s_k C + K)^*$. If an exact LU-factorization is not available, one has to use inexact Newton schemes, such as Jacobi-Davidson style methods [139, 140, 142].

Note that QDPA operates on the original $n \times n$ state-space matrices and that no linearization is needed. This is an advantage, since applying standard DPA to the linearized system (6.2.5) requires factorizations of $2n \times 2n$ matrices (although the $2n \times 2n$ linear equations can also be solved using only factorization of $n \times n$ matrices, see also Section 6.3.2).

It is well known that when computing Rayleigh quotients corresponding to $\mathbf{v}$ and $\mathbf{w}$, this leads to a second-order equation and hence to *two* Ritz values [10]. Selection of the best Ritz value can be difficult [75]. However, in QDPA the choice is automatically made via the Newton update (cf. step 5).

The matrix $(s_{k+1}^2 M + s_{k+1} C + K)$ in step 6 can be reused in step 3 and 4 of the next iteration, where it is needed anyway. In step 5, however, it is in general (depending on the sparsity of M and C), more efficient to compute $(2s_k M + C)\mathbf{v}_k$ as $(2s_k)(M\mathbf{v}_k) + C\mathbf{v}_k$.

In [126] it is shown that DPA tends to converge to more dominant poles than two-sided Rayleigh quotient iteration. The fixed right-hand sides in DPA are crucial for this behavior, and it is expected that QDPA has the same desirable convergence. In [65, 66] DPA was applied to compute dominant poles of high-order rational (polynomial) matrices $Y(s)$ (nodal admittance matrices for electrical networks), which together with the derivative $Y'(s)$ have straightforward construction laws. Other Newton based schemes (of which QDPA in fact is a specialization) for the computation of eigenvalues are presented, for instance, in [130]. In [152, Section 6] and [99] large overviews of existing methods are given. The methods presented in this work focus on the computation of dominant poles.

6.3 Subspace acceleration, selection and deflation

QDPA can be extended with subspace acceleration and a selection strategy to improve global convergence to the most dominant poles, and deflation to avoid recomputation of already found poles. Although the ingredients are the same as for SADPA [123] and SAMDP [122], they are repeated here because especially the deflation procedure is more complicated for the quadratic eigenvalue problem [95].

6.3.1 Subspace acceleration and selection

Instead of discarding the intermediate approximations $\mathbf{v}_k$ and $\mathbf{w}_k$ of the right and left eigenvectors in step 3 and 4, they are kept in (bi)orthogonal search spaces V and W. Following the Petrov-Galerkin approach, the projected quadratic eigenvalue

problem becomes

$$(\tilde{\lambda}^2\widetilde{M} + \tilde{\lambda}\widetilde{C} + \widetilde{K})\tilde{\mathbf{x}} = 0, \text{ and } \tilde{\mathbf{y}}^*(\tilde{\lambda}^2\widetilde{M} + \tilde{\lambda}\widetilde{C} + \widetilde{K}) = 0, \tag{6.3.1}$$

where the $k \times k$ matrices $\widetilde{M}$, $\widetilde{C}$, $\widetilde{K}$ are given by

$$\widetilde{M} = W^*MV, \quad \widetilde{C} = W^*CV, \text{ and } \widetilde{K} = W^*KV.$$

In the k-th iteration this projected quadratic eigenvalue problem is of small size $k \ll n$ and hence can be solved via linearization and the QZ method. With the eigentriplets $(\tilde{\lambda}_i, \tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)$ of (6.3.1), approximate eigentriplets of the original problem can be constructed as

$$(\hat{\lambda}_i = \tilde{\lambda}_i, \hat{\mathbf{x}}_i = V\tilde{\mathbf{x}}_i, \hat{\mathbf{y}}_i = W\tilde{\mathbf{y}}_i), \quad (i = 1, \ldots, k). \tag{6.3.2}$$

From these approximate triplets the most dominant on is selected. To determine the most dominant one, first the approximate left and right eigenvectors need to be normalized so that (cf. (6.2.9))

$$-\hat{\mathbf{y}}_i^*K\hat{\mathbf{x}}_i + \hat{\lambda}_i^2\hat{\mathbf{y}}_i^*M\hat{\mathbf{x}}_i = 1.$$

The approximate residues $\widehat{R}_i$ are given by

$$\widehat{R}_i = (\mathbf{c}^*\hat{\mathbf{x}}_i)(\hat{\mathbf{y}}_i^*\mathbf{b})\hat{\lambda}_i,$$

and the approximate triplets are sorted in decreasing $|\widehat{R}_i|/|\mathrm{Re}(\hat{\lambda}_i)|$ order. The shift in the next iteration becomes $s_{k+1} = \hat{\lambda}_1$.

Note that it is not necessary to compute the approximate eigentriplets (6.3.2) explicitly, since the approximate residues $\widehat{R}_i$ can be computed as

$$\widehat{R}_i = ((\mathbf{c}^*V)\tilde{\mathbf{x}}_i)(\tilde{\mathbf{y}}_i^*(W^*\mathbf{b}))\tilde{\lambda}_i \quad (= (\mathbf{c}^*\hat{\mathbf{x}}_i)(\hat{\mathbf{y}}_i^*\mathbf{b})\hat{\lambda}_i),$$

provided the $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{y}}_i$ are scaled so that

$$1 = -\tilde{\mathbf{y}}_i^*\widehat{K}\tilde{\mathbf{x}}_i + \tilde{\lambda}_i^2\tilde{\mathbf{y}}_i^*\widehat{M}\tilde{\mathbf{x}}_i \quad (= -\hat{\mathbf{y}}_i^*K\hat{\mathbf{x}}_i + \hat{\lambda}_i^2\hat{\mathbf{y}}_i^*M\hat{\mathbf{x}}_i).$$

Numerically, however, it is often more robust to normalize the approximate eigenvectors so that

$$\hat{\mathbf{x}}_i^*\hat{\mathbf{x}}_i = \hat{\mathbf{y}}_i^*\hat{\mathbf{y}}_i = 1,$$

since then the angles $\angle(\hat{\mathbf{x}}_i, \mathbf{c})$ and $\angle(\hat{\mathbf{y}}_i, \mathbf{b})$ are exploited [72, 126]. These scalings can still be performed on the vectors $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{y}}_i$.

The use of subspace acceleration has several advantages. Firstly, it may improve global convergence since the most dominant approximation is selected every iteration, and secondly, after convergence of a pole triplet already good approximations of other dominant pole triplets may be available. The price one has to pay are the costs for keeping the search spaces V and W orthogonal.

6.3.2 Deflation

In contrast to ordinary and generalized eigenproblems, the at most $2n$ eigenvectors of a quadratic eigenproblem obviously are not independent. Hence deflation by restriction cannot be applied directly (without linearization). Instead, an approach inspired by the deflation described in [95] is used here. The idea is to implement deflation via the linearized eigenproblem, since the eigenvectors of the linearized eigenproblem are independent (assuming all eigenvalues are nondegenerate). This can be organized in two non-equivalent ways, as will be discussed next.

In the following, suppose that the $(n \times k)$ matrices X and Y have as their columns the found right and left eigenvectors $\mathbf{x}_i$ and $\mathbf{y}_i$ $(i = 1, \ldots, k)$ of the QEP, respectively, and let Λ be a diagonal $(k \times k)$ matrix with the corresponding eigenvalues on its diagonal. The corresponding eigenvector matrices for the linearized problem (A, B) are

$$\Phi = \begin{bmatrix} X \\ X\Lambda \end{bmatrix}, \text{ and } \Psi = \begin{bmatrix} Y \\ Y\Lambda^* \end{bmatrix}.$$

Furthermore, let the eigenvectors in X and Y be normalized so that $\Psi^* A \Phi = \Lambda$ and $\Psi^* B \Phi = I$. Then it follows that the pencil

$$((I - B\Phi\Psi^*)A(I - \Phi\Psi^* B), (I - B\Phi\Psi^*)B(I - \Phi\Psi^* B))$$

has the same eigenvalues and eigenvectors as (A, B), but with the found eigenvalues transformed to zero.

Approach I: Orthogonalizing against found eigenvectors

A way to implement deflation for the expansion vectors $\mathbf{v}_k$ and $\mathbf{w}_k$ of the search spaces V and W of quadratic eigenproblem is:

1. Construct approximate eigenvectors for (A, B) via

$$\phi_k = \begin{bmatrix} \mathbf{v}_k \\ \sigma \mathbf{v}_k \end{bmatrix}, \quad \psi_k = \begin{bmatrix} \mathbf{w}_k \\ \bar{\sigma} \mathbf{w}_k \end{bmatrix},$$

   where σ is the corresponding approximate eigenvalue, computed via the Newton update (cf. (6.2.13)).

2. Project these approximations via

$$\begin{bmatrix} \phi^1 \\ \phi^2 \end{bmatrix} = (I - \Phi\Psi^* B)\phi_k, \text{ and } \begin{bmatrix} \psi^1 \\ \psi^2 \end{bmatrix} = (I - \Psi\Phi^* B^*)\psi_k.$$

3. Expand V and W (bi)orthogonally with ϕ^1 and ψ^1, respectively.

Approach II: Deflation by restriction of linearized input and output vectors

A second way is to take advantage of the following efficient deflation for first-order transfer functions. It can be verified ($\Psi^* B \Phi = I$) that with the deflated input and output vectors of the linearized problem

$$\mathbf{b}_d = \begin{bmatrix} \mathbf{b}_d^1 \\ \mathbf{b}_d^2 \end{bmatrix} = (I - B\Phi\Psi^*) \begin{bmatrix} 0 \\ \mathbf{b} \end{bmatrix}, \text{ and } \mathbf{c}_d = \begin{bmatrix} \mathbf{c}_d^1 \\ \mathbf{c}_d^2 \end{bmatrix} = (I - B^*\Psi\Phi^*) \begin{bmatrix} \mathbf{c} \\ 0 \end{bmatrix},$$

the residues of the deflated found poles are transformed to zero. Secondly, a straightforward manipulation shows that

$$(s_k B - A)^{-1} \mathbf{b}_d \perp B^*\Psi, \text{ and } (s_k B - A)^{-*} \mathbf{c}_d \perp B\Phi,$$

so that in principle no explicit B-orthogonalizations of expansion vectors against found eigenvectors are needed. This property makes deflation very cheap, since per found pole only once two projections are needed to compute $\mathbf{b}_d$ and $\mathbf{c}_d$.

The linear systems for DPA applied to the linearized system (6.2.4) are

$$(s_k B - A)\mathbf{v}_g = \mathbf{b}_d, \text{ and } (s_k B - A)^* \mathbf{w}_g = \mathbf{c}_d,$$

with A and B as in (6.2.4). These systems can be solved without forming A and B: with $\mathbf{v}_g^2$ and $\mathbf{w}_g^2$ solutions of

$$(s_k^2 M + s_k C + K)\mathbf{v}_g^2 = s_k \mathbf{b}_d^2 + \mathbf{b}_d^1, \text{ and } (s_k^2 M + s_k C + K)^* \mathbf{w}_g^2 = \bar{s}_k \mathbf{c}_d^2 + \mathbf{c}_d^1,$$

it follows that for $s_k \neq 0$

$$\mathbf{v}_g = \begin{bmatrix} (-K^{-1}\mathbf{b}_d^1 + \mathbf{v}_g^2)/s_k \\ \mathbf{v}_g^2 \end{bmatrix}, \text{ and } \mathbf{w}_g = \begin{bmatrix} (-K^{-*}\mathbf{c}_d^1 + \mathbf{w}_g^2)/\bar{s}_k \\ \mathbf{w}_g^2 \end{bmatrix}.$$

The search spaces V and W are expanded with $\mathbf{v}_g^1 = (-K^{-1}\mathbf{b}_d^1 + \mathbf{v}_g^2)/s_k$ and $\mathbf{w}_g^1 = (-K^{-*}\mathbf{c}_d^1 + \mathbf{w}_g^2)/\bar{s}_k$, respectively. Due to rounding errors, however, components in the direction of already found eigenvectors may enter the search spaces again. To avoid convergence to found eigentriplets it is therefore needed to compute the approximate residues via the deflated input and output vectors $\mathbf{b}_d$ and $\mathbf{c}_d$, so that in fact deflation takes place implicitly. If $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ are approximate normalized right and left eigenvectors corresponding to $\hat{\lambda}$ of the QEP, then the approximate linearized residue is computed as

$$\widehat{R} = (\mathbf{c}_d^* \begin{bmatrix} \hat{\mathbf{x}} \\ \hat{\lambda}\hat{\mathbf{x}} \end{bmatrix})(\begin{bmatrix} \hat{\mathbf{y}}^* & \hat{\lambda}\hat{\mathbf{y}}^* \end{bmatrix} \mathbf{b}_d).$$

Because the residues of already found poles are transformed to zero, there is practically no risk of selecting approximations of already found poles, so that expansion of the search spaces with already found components is limited.

The two approaches described in this section are not equivalent, i.e. given the same shift s_k, they do not compute the same expansion vectors, nor is it possible to

decide which is the best on pure theoretical grounds. An advantage of approach I is that the original Newton equations are solved, but on the other hand the Newton update needs to be computed in step 1, and moreover, orthogonalization must be carried out every iteration. The big advantage of approach II is that deflation needs to be carried out only once per found pole (for $\mathbf{b}$ and $\mathbf{c}$), while it is required that K is nonsingular, since one solve with K and K^* per found pole is needed to compute $\mathbf{b}_d$ and $\mathbf{c}_d$. Numerical experiments, reported in Section 6.4, show that approach II is more efficient and effective in practice.

6.3.3 Improving local convergence

As soon as an approximate pole triplet is accurate enough, for instance when $\|(s_{k+1}^2 M + s_{k+1}C + K)\mathbf{v}_k\|_2 < \epsilon_r$ with $\epsilon < \epsilon_r < 10^{-4}$, then convergence can be accelerated by using one or two iterations of quadratic two-sided Rayleigh Quotient Iteration, described in Alg. 6.2 (see [139] for a Jacobi-Davidson equivalent). The equation for the Rayleigh quotients, given left and right eigenvector approximations $\mathbf{w}_{k+1}$ and $\mathbf{v}_{k+1}$, is

$$(\tilde{\lambda}^2 \mathbf{w}_{k+1}^* M \mathbf{v}_{k+1} + \tilde{\lambda}\mathbf{w}_{k+1}^* C \mathbf{v}_{k+1} + \mathbf{w}_{k+1}^* K \mathbf{v}_{k+1})\tilde{\mathbf{x}} = 0,$$

which is a 1×1 QEP, and hence provides *two* Ritz values, so a selection must be made [75]. The best choice here would be to select the Ritz value closest to s_{k+1}. For the goal of improving an already rather accurate eigentriplet, however, it is very likely that the approximate eigenvalue is already of the desired accuracy, while the corresponding eigenvector approximations need to be improved. To avoid the selection procedure, the new eigenvalue is computed via the Newton update (cf. step 5 in Alg. 6.1). Essentially, the QRQI presented in Alg. 6.2 is the same as QDPA, but the right-hand sides are updated every iteration.

6.4 Numerical results

In this section, subspace accelerated QDPA is applied to two large-scale examples. The constructed modal equivalents are compared to reduced-order models computed by a second-order Arnoldi method [11], and it is shown how the dominant poles computed by QDPA can be used to improve such reduced-order models. All experiments reported here are executed in Matlab 7.3 on a SUN Ultra 20 (AMD Opteron 2.8GHz, 2GB RAM). During the selection procedure of the most dominant eigentriplet, the approximate eigenvectors are scaled so that (cf. Section 6.3.1)

$$\hat{\mathbf{x}}_i^* \hat{\mathbf{x}}_i + |\hat{\lambda}_i|^2 \hat{\mathbf{x}}_i^* \hat{\mathbf{x}}_i = \hat{\mathbf{y}}_i^* \hat{\mathbf{y}}_i + |\hat{\lambda}_i|^2 \hat{\mathbf{y}}_i^* \hat{\mathbf{y}}_i = 1.$$

6.4.1 The Butterfly gyro

The Butterfly gyro, developed at the Imego Institute with Saab Bofors Dynamics AB, is a vibrating micro-mechanical gyro that is used in inertial navigation

Algorithm 6.2 Quadratic Rayleigh Quotient Iteration (QRQI)

INPUT: System $(M, C, K, \mathbf{v}_0, \mathbf{w}_0)$, initial estimate s_0, tolerance $\epsilon \ll 1$
OUTPUT: Approximate eigenvalue λ and corresponding right and left eigenvectors $\mathbf{x}$ and $\mathbf{y}$

1: Set $k = 0$
2: **while** not converged **do**
3: Solve $\mathbf{v}_{k+1} \in \mathbb{C}^n$ from $(s_k^2 M + s_k C + K)\mathbf{v}_{k+1} = (2s_k M + C)\mathbf{v}_k$
4: Solve $\mathbf{w}_{k+1} \in \mathbb{C}^n$ from $(s_k^2 M + s_k C + K)^* \mathbf{w}_{k+1} = (2s_k M + C)^* \mathbf{w}_k$
5: Compute the new eigenvalue estimate

$$s_{k+1} = s_k - \frac{\mathbf{w}_k^*(2s_k M + C)^* \mathbf{v}_{k+1}}{\mathbf{w}_{k+1}^*(2s_k M + C)\mathbf{v}_{k+1}}$$

6: Set $\mathbf{v}_{k+1} = \mathbf{v}_{k+1}/\|\mathbf{v}_{k+1}\|_2$ and $\mathbf{w}_{k+1} = \mathbf{w}_{k+1}/\|\mathbf{w}_{k+1}\|_2$
7: The eigenvalue $\lambda = s_{k+1}$ with $\mathbf{x} = \mathbf{v}_{k+1}$ and $\mathbf{y} = \mathbf{w}_{k+1}$ has converged if

$$\|(s_{k+1}^2 M + s_{k+1} C + K)\mathbf{v}_{k+1}\|_2 < \epsilon$$

8: Set $k = k + 1$
9: **end while**

applications, for instance for the detection of (unwanted) forces in (unwanted) directions (see [2, 88] for more details). For future improvements of the Butterfly gyro, efficiency and accuracy of gyro simulations is of great importance. Model order reduction not only reduces the simulation times significantly, it also provides a state-space equivalent formulation of the original finite element representation, which is helpful in developing and testing signal processing algorithms for the gyro.

The system matrices of the second-order system $(M, C, K, \mathbf{b}, \mathbf{c})$ can be found in the Oberwolfach benchmark collection [2]. The full system has 17361 degrees of freedom, one input and 12 outputs. For the experiments here $\mathbf{b}$ was the input vector B, $\mathbf{c}$ was taken to be the first column of the 17361×12 selector (output) matrix L, and $C = \beta K$ with $\beta = 10^{-7}$ (equal to the settings in [88]).

The QDPA algorithm with subspace acceleration was used to compute 5, 10, 20, 30, and 35 dominant poles, using both of the deflation strategies described in Section 6.3.2. Since the frequency range of interest is from 10^4 Hz to 10^6 Hz, the initial shift was $s_0 = 2\pi i(1.5 \cdot 10^5)$. The process was restarted with search spaces containing the $k_{\min} = 4$ most dominant approximations at dimension $k_{\max} = 10$. Note that all linear system solves were computed directly, that is using the backslash operator in Matlab, and not using LU-factorizations of $Q(s_k) = s_k^2 M + s_k C + K$, since this appeared to be significantly faster (and requiring less memory). The running times shown in Table 6.1 show that deflation approach II (implicitly solving the linearized system and deflating $\mathbf{b}$ and $\mathbf{c}$, see Section 6.3.2) is more efficient than deflation approach I (explicit orthogonalization against found eigenvectors). The

Table 6.1: Computational times for computing dominant poles and left and right eigenvectors of the Butterfly Gyro ($n = 17361$). Deflation type I refers to explicit orthogonalization against found eigenvectors, type II refers to using implicit linearization and deflated input and output vectors. Every iteration a solve with $Q(s_k) = s_k^2 M + s_k C + K$ and $(Q(s_k))^*$ is needed.

	Deflation I		Deflation II	
#poles	Time (s)	#iterations	Time (s)	#iterations
5	147	28	111	16
10	541	111	228	33
20	1157	235	576	91
30	1947	389	1100	191
35	2560	508	1345	233

frequency response Bode plots, based on the modal equivalents computed using the second deflation strategy, are shown in Figure 6.1. Since qualitatively there is no difference between the two deflation strategies (not shown here), deflation strategy II is the method of choice.

Typically, the 20 most dominant poles, causing the highest peaks in the Bode plot, were found relatively quickly compared to the total running time for computing 35 dominant pole triplets. As more dominant poles are found, only less distinguishing, more or less equally dominant poles remain and QDPA has difficulties in finding these less dominant poles. Comparing the 20th, 30th and 35th order solutions (note that the *real orthogonal* bases for the left and right eigenspaces were equal to the number of poles found for these models, see also Section 6.2), it can also be observed that qualitatively the improvement is smaller. For practical purposes, especially a good match in the $10^4 - 10^5$ Hz frequency range is of importance.

In [88] a 40th order reduced model of the Butterfly gyro was presented. This model was created by computing an orthonormal basis for the Krylov space $\mathcal{K}^{40}(K^{-1}M, K^{-1}\mathbf{b})$ (note that C was neglected and in fact PRIMA [105] was applied, see [128] for more details). If the columns of X form an orthonormal basis for this Krylov space, then the reduced system becomes

$$(X^*MX, X^*CX, X^*KX, X^*\mathbf{b}, X^*\mathbf{c}).$$

In Figure 6.2 the Bode plot of this 40th order model is plotted together with the 30th and 35th order QDPA models. In the eye norm it appears as if the 35th and 40th order model are qualitatively the same, except near 10^6 Hz, where the 35th order QDPA model is more accurate. Although the latter is confirmed by the relative error plots in Figure 6.3, it can also be observed that the 40th order model is more accurate in the lower frequency range. Both the 35th order QDPA and 40th order PRIMA model, however, are of acceptable accuracy. The accuracy of the QDPA model is almost constant over the range 10^4 Hz to 10^6 Hz, while the accuracy of the PRIMA model decreases after 10^5 Hz. This can be explained by the fact that the expansion point for PRIMA was $\sigma = 0$, so that the reduced model is more accurate in the lower frequency range. The QDPA models, on the other hand, are accurate in the neighborhood of frequencies near the imaginary parts of

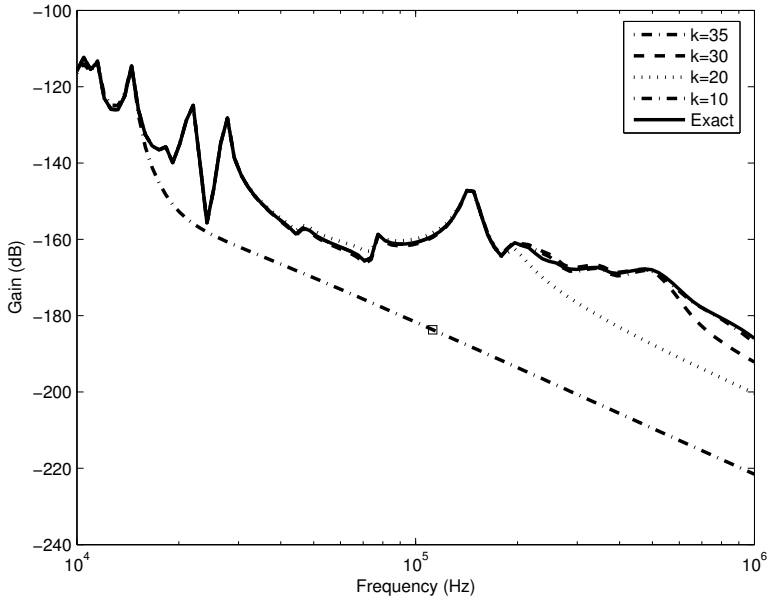

Figure 6.1: Exact transfer function (solid) and 35th (dash-dot), 30th (dash), 20th (dot), and 10th (dash-dot-square) order modal equivalents (based on eigenspaces of 35, 30, 20 and 10 dominant poles, respectively).

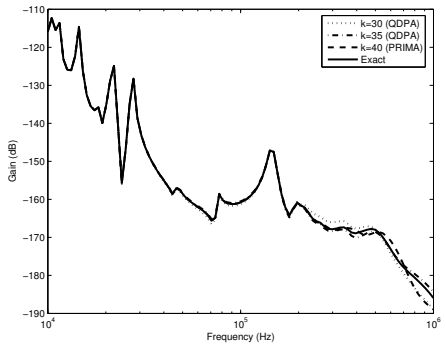

Figure 6.2: Exact response (solid), 40th order PRIMA model (dash), and 35th (dash-dot) and 30th (dot) order modal equivalents.

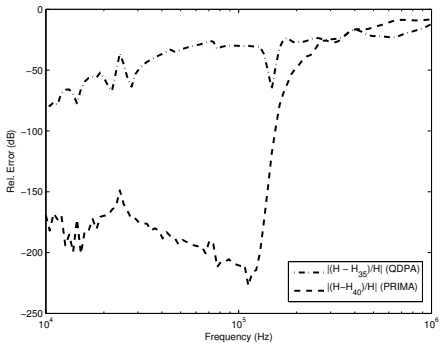

Figure 6.3: Relative error for 40th order PRIMA model (dash) and 35th order QDPA modal equivalent (dash-dot).

the dominant poles.

Although the PRIMA model can be computed in considerably less time (about $20s$), it will not always produce more accurate reduced models. If the imaginary parts of the dominant poles vary largely in magnitude and hence the Bode plot shows peaks over a larger frequency range, then a single expansion point may not be sufficient to produce an acceptable reduced model (rational Krylov methods may be used to handle this, see [58] and the next example), while subspace accelerated QDPA finds the dominant poles automatically.

6.4.2 The breathing sphere

The breathing sphere, taken from [82], is a three-dimensional vibrating body and has its origin in sound radiation analysis (acoustics). A finite element discretization leads to a second-order model of order $n = 17611$, with transfer function

$$H(s) = \mathbf{c}^*(sM^2 + sC + K)\mathbf{b}.$$

The input vector $\mathbf{b}$ contains the normal-velocities of each element, the state $\mathbf{x}$ represents the acoustic pressure on the corresponding elements, and the output vector $\mathbf{c}$ selects state variables. In this experiment all elements of $\mathbf{c}$ are zero, except for $\mathbf{c}_{173} = 1$[2]. The mass matrix M is symmetric.

The QDPA algorithm with subspace acceleration was used to compute 10, 20, 40, and 60 dominant poles, using deflation strategy II described in Section 6.3.2. The initial shift was $s_0 = 2i$. The process was restarted with search spaces containing the $k_{\min} = 4$ most dominant approximations at dimension $k_{\max} = 10$. All linear system solves were computed directly using the backslash operator in Matlab. The running times are shown in Table 6.2. The frequency response

[2]Communicated by J. Lampe [82].

Bode plots of the modal equivalents of order $k = 2 \times 40$ and $k = 2 \times 60$ are shown in Figure 6.4, together with a 40th order rational second-order Krylov model (Rational Krylov via Arnoldi (RKA)). The latter model was constructed as $(W^*MV, W^*CV, W^*KV, W^*\mathbf{b}, V^*\mathbf{c})$, where the columns of V and W are orthonormal bases for the second-order rational Krylov subspaces

$$\sum_{j=1}^{4} \mathcal{G}^{10}(-\widetilde{K}_j^{-1}\widetilde{C}_j, -\widetilde{K}_j^{-1}M, \widetilde{K}_j^{-1}\mathbf{b}),$$

and

$$\sum_{j=1}^{4} \mathcal{G}^{10}(-\widetilde{K}_j^{-*}\widetilde{C}_j^{*}, -\widetilde{K}_j^{-*}M^{*}, \widetilde{K}_j^{-*}\mathbf{c}),$$

respectively, with $\widetilde{K}_j = \sigma_j^2 M + \sigma_j C + K$, $\widetilde{C}_j = 2\sigma_j M + C$, and interpolation points $\sigma_1 = 0.1$, $\sigma_2 = 0.5$, $\sigma_3 = 1$, $\sigma_4 = 5$. A second-order Krylov subspace is defined as

$$\mathcal{G}^{k+1}(A, B, \mathbf{v}_0) = \text{span}(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_k),$$

with

$$\begin{aligned} \mathbf{v}_1 &= A\mathbf{v}_1 \\ \mathbf{v}_i &= A\mathbf{v}_{i-1} + B\mathbf{v}_{i-2}, \quad (i = 2, \ldots, k). \end{aligned}$$

See [11] for the complete second-order Arnoldi algorithm (SOAR) and more details.

The 40th order SOAR model captures the response for low frequencies, but fails to match the peaks around 0.6 rad/s and higher frequencies. The QDPA modal equivalents, on the other hand, are more accurate in the details. The CPU time to produce the SOAR model was only 506 seconds, but it is difficult to choose the interpolation points in such a way that the model captures more detail. QDPA computes the dominant poles automatically and consequently, the modal equivalents capture the details (the peaks) in the frequency response.

These observations lead to the idea of filling in the missing details of the rational SOAR model by expanding the Krylov bases V and W with the right and left eigenvectors X and Y of dominant poles computed by QDPA: $\widetilde{V} = [V, X]$ and $\widetilde{W} = [W, Y]$, respectively. The columns of $\widetilde{V}$ and $\widetilde{W}$ are kept orthogonal, and the new reduced-order model is constructed as $(\widetilde{W}^*M\widetilde{V}, \widetilde{W}^*C\widetilde{V}, \widetilde{W}^*K\widetilde{V}, \widetilde{W}^*\mathbf{b}, \widetilde{V}^*\mathbf{c})$. Note that the number of matched moments (thanks to SOAR [11]) is the same for this hybrid model, and also the dominant poles computed by QDPA are poles of this model. Figure 6.5 shows the frequency responses of a 10th order modal equivalent computed by QDPA with $s_0 = 0.6i$ (25 iterations, 691 seconds), the 40th order SOAR RKA model (506 seconds) and the hybrid model, and the exact response. The 10th order modal equivalent captures the peaks near $\omega = 0.6$ rad/s that the SOAR model misses. The hybrid model combines the best of the two models, as can also be observed in the error plot in Fig. 6.6: the relative error near $\omega = 0.6$ rad/s drops from $O(1)$ to $O(10^{-2})$. Adding more dominant poles leads to improvements in other frequency ranges as well.

Table 6.2: Computational times for computing dominant poles and left and right eigenvectors of the breathing sphere ($n = 17611$). Every iteration a solve with $Q(s_k) = s_k^2 M + s_k C + K$ and $(Q(s_k))^*$ is needed.

#poles	Time (s)	#iterations
10	2800	108
20	4280	156
40	13901	546
60	29880	1223

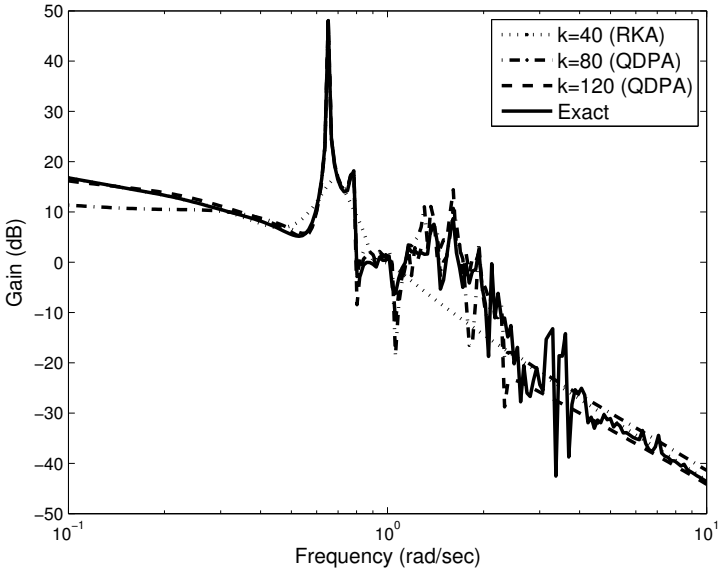

Figure 6.4: Exact transfer function (solid), 40th order SOAR RKA model (dot), and 80th (dash-dot) and 120th (dash) order modal equivalents.

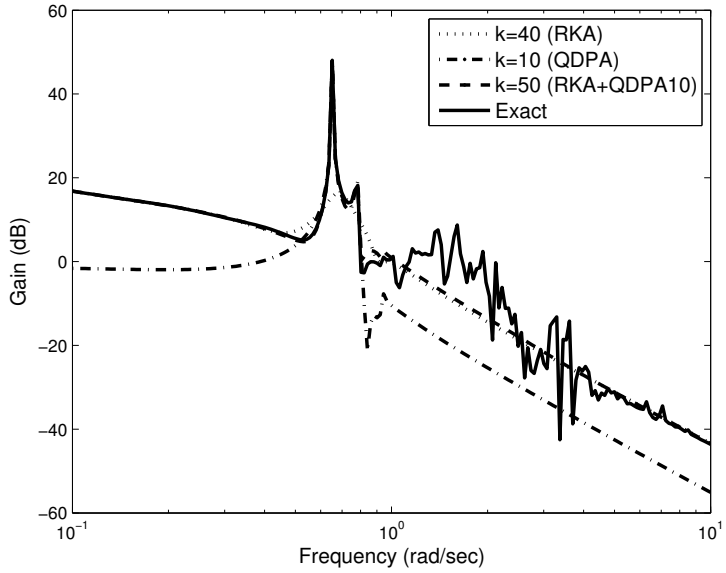

Figure 6.5: Exact transfer function (solid), 40th order SOAR RKA model (dot), 10th (dash-dot) order modal equivalent, and 50th order hybrid RKA+QDPA (dash).

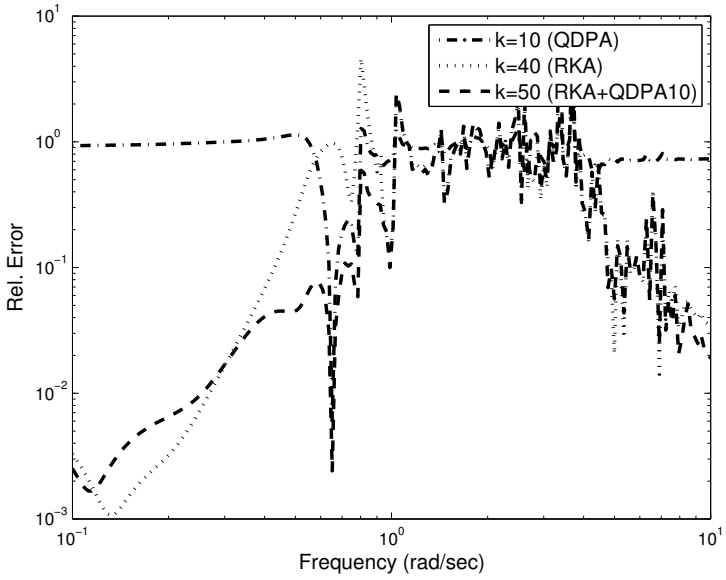

Figure 6.6: Relative errors for 10th order modal equivalent (dash-dot), 40th order SOAR RKA model (dot), and 50th order hybrid RKA+QDPA (dash).

The other way around, the imaginary parts of the dominant poles computed by QDPA can also be used as interpolation points for the rational SOAR models, to get better detail in the corresponding frequency range. Given a dominant pole $\lambda = \alpha + \beta i$, one can use either $\sigma = \beta$ (real) or $\sigma = \beta i$ as interpolation point. According to [70, Chapter 6] real shifts have a more global effect, while purely imaginary shifts focus more on detail, and hence this is advised when trying to capture the peaks. Figure 6.7 shows the frequency response of a 70th order SOAR model that was computed using interpolation points $\sigma_1 = 0.65i$, $\sigma_2 = 0.78i$, $\sigma_3 = 0.93i$, and $\sigma_4 = 0.1$. The imaginary shifts σ_1, σ_2, and σ_3 correspond to the imaginary parts of the dominant poles that cause peaks between $\omega = 0.6$ rad/s and $\omega = 1$ rad/s. These poles were selected from 5 poles computed by QDPA with $s_0 = 0.6i$ (691 seconds, the same run as in the previous paragraph). For each interpolation point a 10-dimensional dual second-order Krylov subspace was computed using SOAR (1373 seconds), and because real bases were used in the projection, the dimension of the reduced-order model is $k = 3 \times 20 + 10 = 70$. This reduced model is more accurate than the previous reduced-order models, up to $\omega = 1$ rad/s. Although the dominant poles near the imaginary shifts are present in this reduced-order model, this is in general not guaranteed. Nevertheless, this approach appears to be more robust than adding dominant poles (and corresponding states) to the reduced-order model like in the previous paragraph, albeit computationally more expensive since complex interpolation points are needed. Combined with strategies that determine the dimensions of the rational Krylov spaces dynamically, as described in [70, Chapter 6] and [58], more efficient schemes can be designed.

6.5 Higher order systems, zeros, and MIMO systems

6.5.1 Higher order polynomial systems

The algorithms and techniques developed in the previous sections can be applied to higher order polynomial transfer functions. Consider the p-th order polynomial transfer function

$$H(s) = \mathbf{c}^*(s^p A_p + s^{p-1}A_{p-1} + \cdots + sA_1 + A_0)^{-1}\mathbf{b},$$

where $A_i \in \mathbb{R}^{n\times n}$ and $\mathbf{b}, \mathbf{c} \in \mathbb{R}^n$. Then again Newton's method can be applied to compute the poles of $H(s)$ via the zeros of $1/H(s)$, leading to

$$s_{k+1} = s_k - \frac{\mathbf{c}^*(s_k^p A_p + s_k^{p-1}A_{p-1} + \cdots + s_k A_1 + A_0)^{-1}\mathbf{b}}{\mathbf{w}^*(ps_k^{p-1}A_p + (p-1)s_k^{p-2}A_{p-1} + \cdots + s_k A_2 + A_1)\mathbf{v}},$$

where $\mathbf{v} = (s_k^p A_p + s_k^{p-1}A_{p-1} + \cdots + s_k A_1 + A_0)^{-1}\mathbf{b}$ and $\mathbf{w} = (s_k^p A_p + s_k^{p-1}A_{p-1} + \cdots + s_k A_1 + A_0)^{-*}\mathbf{c}$. Since the poles are eigenvalues of the polynomial eigenproblem (PEP)

$$\begin{array}{rcll} (\lambda_i^p A_p + \lambda_i^{p-1}A_{p-1} + \cdots + \lambda_i A_1 + A_0)\mathbf{x}_i & = & 0, & \mathbf{x}_i \neq 0, \\ \mathbf{y}_i^*(\lambda_i^p A_p + \lambda_i^{p-1}A_{p-1} + \cdots + \lambda_i A_1 + A_0) & = & 0, & \mathbf{y}_i \neq 0, \end{array} \tag{6.5.1}$$

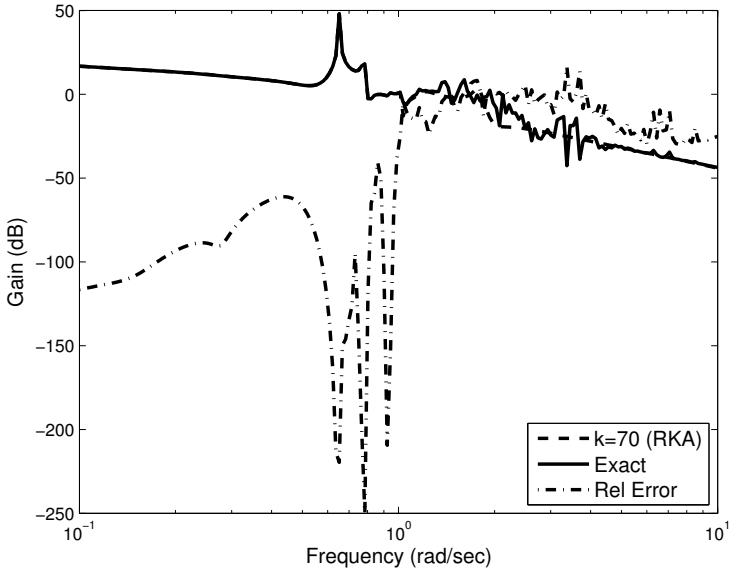

Figure 6.7: Exact transfer function (solid), 70th order SOAR RKA model (dash) using interpolation points based on dominant poles, and relative error (dash-dot).

for $i = 1, \ldots, np$, a stopping criterion is

$$||(s_{k+1}^p A_p + s_{k+1}^{p-1} A_{p-1} + \cdots + s_{k+1} A_1 + A_0)\mathbf{v}||_2 < \epsilon, \quad \epsilon \ll 1.$$

The vectors $\mathbf{x} = \mathbf{v}$ and $\mathbf{y} = \mathbf{w}$ are the corresponding approximate right and left eigenvectors of the PEP.

A corresponding linearized system can be of the form

$$\begin{cases} B\dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + \mathbf{b}_l u(t) \\ y(t) &= \mathbf{c}_l^* \mathbf{x}(t) + du(t), \end{cases}$$

where

$$A = \begin{bmatrix} 0 & N_0 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & N_1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & N_{p-2} \\ -A_0 & -A_1 & \cdots & \cdots & -A_{p-2} & -A_{p-1} \end{bmatrix},$$

$B = \mathrm{diag}(N_0, N_1 \ldots, N_{p-2}, A_p)$, the N_i $(i = 0, \ldots, p-2)$ are nonsingular $n \times n$ matrices , $\mathbf{b}_l = [0, \ldots, 0, \mathbf{b}^T]^T \in \mathbb{R}^{np \times 1}$ and $\mathbf{c}_l = [\mathbf{c}^T, 0, \ldots, 0]^T \in \mathbb{R}^{np \times 1}$. Note that with the choice $N_0 = A_0$ (if A_0 is nonsingular), one obtains for $p = 2$ the QEP as described in Section 6.2. Another possible choice is $N_i = I$ for all $i = 0, \ldots, p-2$.

The right and left eigenvectors of the linearized eigenproblem

$$A\phi_i = \lambda_i B\phi_i, \quad \psi_i^* A = \lambda_i \psi_i^* B, \quad \phi_i, \psi_i \neq 0,$$

can be verified to be

$$\phi_i = \begin{bmatrix} \mathbf{x}_i \\ \lambda_i \mathbf{x}_i \\ \lambda_i^2 \mathbf{x}_i \\ \vdots \\ \lambda_i^{p-2} \mathbf{x}_i \\ \lambda_i^{p-1} \mathbf{x}_i \end{bmatrix}, \text{ and } \psi_i = \begin{bmatrix} -N_0^{-*}(\lambda_i^{p-2} A_0)^* \mathbf{y}_i \\ -N_1^{-*}(\lambda_i^{p-3} A_0 + \lambda_i^{p-2} A_1)^* \mathbf{y}_i \\ \vdots \\ \vdots \\ -N_{p-2}^{-*}(A_0 + \lambda_i A_1 + \ldots + \lambda_i^{p-2} A_{p-2})^* \mathbf{y}_i \\ \bar{\lambda}_i^{p-1} \mathbf{y}_i \end{bmatrix}.$$

Denoting the matrix polynomial of the PEP (6.5.1) by $Q(\lambda) = s^p A_p + s^{p-1} A_{p-1} + \cdots + sA_1 + A_0$, it can be verified that

$$\begin{bmatrix} Q(\lambda) & 0 \\ 0 & I \end{bmatrix} = E(\lambda)(A - \lambda B)F(\lambda),$$

where

$$E(\lambda) = \begin{bmatrix} E_{11} & E_{12} & \cdots & \cdots & E_{1,(p-1)} & I \\ -N_0^{-1} & 0 & 0 & \cdots & \cdots & 0 \\ 0 & -N_1^{-1} & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -N_{p-3}^{-1} & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & -N_{p-2}^{-1} & 0 \end{bmatrix},$$

and

$$F(\lambda) = \begin{bmatrix} I & 0 & \cdots & \cdots & \cdots & 0 \\ \lambda I & I & 0 & \cdots & \cdots & 0 \\ \lambda^2 I & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \lambda^{p-1} & \cdots & \cdots & \lambda^2 I & \lambda I & I \end{bmatrix},$$

whit

$$\begin{aligned} E_{11} &= (A_1 + \lambda A_2 + \lambda^2 A_3 + \cdots + \lambda^{p-1} A_p) N_0^{-1}, \\ E_{12} &= (A_2 + \lambda A_3 + \cdots + \lambda^{p-2} A_p) N_0^{-1}, \\ \vdots \quad &\vdots \qquad \vdots \\ E_{1,(p-1)} &= (A_p + \lambda A_{p-1}) N_{p-2}^{-1}. \end{aligned}$$

Indeed, the linear matrix $A - \lambda B$ is a linearization of $Q(\lambda)$.

Using a similar approach to that in Section 6.2, it follows that

$$Q(s)^{-1} = X(sI - \Lambda)^{-1}\Lambda^{p-1}Y^*.$$

The partial fraction representation becomes

$$H(s) = \mathbf{c}^*X(sI - \Lambda)^{-1}\Lambda^{p-1}Y^*\mathbf{b} = \sum_{i=1}^{pn} \frac{R_i}{s - \lambda_i},$$

where the residues are given by (cf. (6.2.8))

$$R_i = (\mathbf{c}^*\mathbf{x}_i)(\mathbf{y}_i^*\mathbf{b})\lambda_i^{p-1}.$$

Note that the $\mathbf{x}_i$ and $\mathbf{y}_i$ are scaled so that

$$\psi_i^* B \phi_i = 1.$$

The techniques for subspace acceleration, selection and deflation, and local convergence improvement, described in Section 6.3, can be generalized to polynomial transfer functions. The modal equivalents and reduced-order models preserve the structure of the original systems.

6.5.2 Transfer function zeros

In [125] it is shown how the (SA)DPA algorithms can be used for the computation of dominant zeros via the dominant poles of the inverse transfer function. In the single input single output (SISO) case, $z_0 \in \mathbb{C}$ is called a transmission zero [135] if $H(z_0) = 0$, where

$$H(s) = \mathbf{c}^*(s^pA_p + s^{p-1}A_{p-1} + \cdots + sA_1 + A_0)^{-1}\mathbf{b} + d.$$

The following two theorems are straightforward generalizations of Theorem 3.1 and Theorem 3.2 in [125] (Chapter 5) and show how inverse systems $\Sigma^z = (A_p^z, A_{p-1}^z, \ldots, A_0^z, \mathbf{b}^z, \mathbf{c}^z, d^z)$ can be defined so that $H^z(s)$ is the inverse of $H(s)$, for $d = 0$ and $d \neq 0$, respectively.

Theorem 6.5.1. *Let*

$$y(s) = H(s)u(s) = \left[\mathbf{c}^*(s^pA_p + s^{p-1}A_{p-1} + \cdots + sA_1 + A_0)^{-1}\mathbf{b}\right] u(s)$$

be the transfer function of the descriptor realization

$$\Sigma = (A_p, A_{p-1}, \ldots, A_0, \mathbf{b}, \mathbf{c}, d = 0),$$

and let

$$y^z(s) = H^z(s)u^z(s) = \left[\mathbf{c}^{z*}(s^pA_p^z + s^{p-1}A_{p-1}^z + \cdots + sA_1^z + A_0^z)^{-1}\mathbf{b}^z\right] u^z(s)$$

be the transfer function of the augmented descriptor realization

$$\Sigma^z = (A_p^z, A_{p-1}^z, \ldots, A_0^z, \mathbf{b}^z, \mathbf{c}^z, d^z = 0),$$

where

$$A_0^z = \begin{bmatrix} A_0 & \mathbf{b} \\ -\mathbf{c}^* & 0 \end{bmatrix}, \quad A_i^z = \begin{bmatrix} A_i & 0 \\ 0 & 0 \end{bmatrix} \quad (i = 1, \dots p),$$
$$\mathbf{b}^z = \begin{bmatrix} \mathbf{b} \\ 1 \end{bmatrix}, \qquad \mathbf{c}^z = \begin{bmatrix} \mathbf{c} \\ 1 \end{bmatrix}, \qquad d^z = 0.$$

Then $H^z(s) = H^{-1}(s)$.

Proof. This proof is a generalization of the proof of [125, Thm. 3.1]. Let $u \equiv u(s)$, $y \equiv y(s) = \mathbf{c}^*\mathbf{x}(s)$, $\mathbf{x} \equiv \mathbf{x}(s) = (s^pA_p + s^{p-1}A_{p-1} + \cdots + sA_1 + A_0)^{-1}\mathbf{b}u(s)$, $u^z \equiv u^z(s)$, $y^z \equiv y^z(s) = \mathbf{c}^{z*}\mathbf{x}^z(s)$, $\mathbf{x}^z \equiv \mathbf{x}^z(s) = (s^pA_p^z + s^{p-1}A_{p-1}^z + \cdots + sA_1^z + A_0^z)^{-1}\mathbf{b}^z u^z(s)$, and the Rosenbrock system matrix [127] equation for Σ

$$\begin{bmatrix} s^pA_p + s^{p-1}A_{p-1} + \cdots + sA_1 + A_0 & -\mathbf{b} \\ \mathbf{c}^* & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}.$$

By adding $(\mathbf{b}y - \mathbf{b}y)$ to the left side of the first equation and defining the error $(u - y)$ as a new algebraic variable, one obtains

$$\left[\begin{array}{cc:c} s^pA_p + s^{p-1}A_{p-1} + \cdots + sA_1 + A_0 & -\mathbf{b} & -\mathbf{b} \\ \mathbf{c}^* & 0 & -1 \\ \hdashline \mathbf{c}^* & 1 & 0 \end{array}\right] \left[\begin{array}{c} \mathbf{x} \\ u - y \\ \hdashline y \end{array}\right] = \left[\begin{array}{c} 0 \\ 0 \\ \hdashline u \end{array}\right]$$

or

$$\begin{bmatrix} s^pA_p^z + s^{p-1}A_{p-1}^z + \cdots + sA_1^z + A_0^z & -\mathbf{b}^z \\ \mathbf{c}^{z*} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}^z \\ u^z \end{bmatrix} = \begin{bmatrix} 0 \\ y^z \end{bmatrix},$$

which is the Rosenbrock system matrix [135] equation for Σ^z. Hence $u^z(s) = y(s)$, $y^z(s) = u(s)$ and $H^z(s) = H^{-1}(s)$. □

A similar result can be obtained for the case $d \neq 0$, see for example for the first order case [78, Ex. 2.2-20] and [79, 137].

Theorem 6.5.2. *Let*

$$y(s) = H(s)u(s) = \left[\mathbf{c}^*(s^pA_p + s^{p-1}A_{p-1} + \cdots + sA_1 + A_0)^{-1}\mathbf{b} + d\right] u(s)$$

be the transfer function of the descriptor realization

$$\Sigma = (A_p, A_{p-1}, \dots, A_0, \mathbf{b}, \mathbf{c}, d \neq 0),$$

and let

$$y^z(s) = H^z(s)u^z(s) = \left[\mathbf{c}^{z*}(s^pA_p^z + s^{p-1}A_{p-1}^z + \cdots + sA_1^z + A_0^z)^{-1}\mathbf{b}^z + d^z\right] u^z(s)$$

be the transfer function of the descriptor realization

$$\Sigma_z = (A_p^z, A_{p-1}^z, \dots, A_0^z, \mathbf{b}^z, \mathbf{c}^z, d^z \neq 0),$$

where

$$A_0^z = A_0 - d^{-1}\mathbf{b}\mathbf{c}^*, \qquad A_i^z = A_i \qquad (i = 1, \dots, p),$$
$$\mathbf{b}^z = d^{-1}\mathbf{b}, \quad \mathbf{c}^z = -d^{-1}\mathbf{c}, \quad d^z = d^{-1}.$$

Then $H^z(s) = H^{-1}(s)$.

Proof. This proof is a generalization of the proof of [125, Thm. 3.1]. Let $u \equiv u(s)$, $y \equiv y(s) = \mathbf{c}^*\mathbf{x}(s) + du(s)$, $\mathbf{x} \equiv \mathbf{x}(s) = (s^pA_p + s^{p-1}A_{p-1} + \cdots + sA_1 + A_0)^{-1}\mathbf{b}u(s)$, $u^z \equiv u^z(s)$, $y^z \equiv y^z(s) = \mathbf{c}^{z*}\mathbf{x}^z(s) + d^zu^z(s)$, $\mathbf{x}^z \equiv \mathbf{x}^z(s) = (s^pA_p^z + s^{p-1}A_{p-1}^z + \cdots + sA_1^z + A_0^z)^{-1}\mathbf{b}_z u_z(s)$, and the Rosenbrock system matrix equation for Σ

$$\begin{bmatrix} s^pA_p + s^{p-1}A_{p-1} + \cdots + sA_1 + A_0 & -\mathbf{b} \\ \mathbf{c}^* & d \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ u \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}. \tag{6.5.2}$$

The expression $u = -d^{-1}\mathbf{c}^*\mathbf{x} + d^{-1}y$ is obtained from the second equation of (6.5.2), which is substituted into the first equation of (6.5.2) to yield

$$\begin{bmatrix} s^pA_p + s^{p-1}A_{p-1} + \cdots + sA_1 + (A_0 - d^{-1}\mathbf{b}\mathbf{c}^*) & -d^{-1}\mathbf{b} \\ -d^{-1}\mathbf{c}^* & d^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{x}_z \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ u \end{bmatrix},$$

or

$$\begin{bmatrix} s^pA_p^z + s^{p-1}A_{p-1}^z + \cdots + sA_1^z + A_0^z & -\mathbf{b}_z \\ \mathbf{c}^{z*} & d^z \end{bmatrix} \begin{bmatrix} \mathbf{x}^z \\ u^z \end{bmatrix} = \begin{bmatrix} 0 \\ y^z \end{bmatrix},$$

which is the Rosenbrock system matrix equation for Σ^z. Hence $u^z(s) = y(s)$, $y^z(s) = u(s)$ and $H^z(s) = H^{-1}(s)$. □

Consequently, the QDPA algorithm with subspace acceleration can be used to compute the dominant zeros of higher order polynomial transfer functions via the dominant poles of the inverse transfer function. In practice, a single implementation of the QDPA algorithm can be used to compute both dominant poles and dominant zeros. For more details on the computation of dominant zeros, see [125].

6.5.3 MIMO transfer functions

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

$$\begin{cases} A_p\mathbf{x}^{(p)}(t) + A_{p-1}\mathbf{x}^{(p-1)}(t) + \cdots + A_1\dot{\mathbf{x}}(t) + A_0\mathbf{x}(t) &= B\mathbf{u}(t) \\ \mathbf{y}(t) &= C^*\mathbf{x}(t) + D\mathbf{u}(t), \end{cases}$$

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

$$H(s) = C^*(s^pA_p + s^{p-1}A_{p-1} + \cdots + sA_1 + A_0)^{-1}B + D. \tag{6.5.3}$$

The dominant poles of (6.5.3) are those $s \in \mathbb{C}$ for which $\sigma_{\max}(H(s)) \to \infty$. For square transfer functions ($m = q$), there is an equivalent criterion: the dominant poles are those $s \in \mathbb{C}$ for which $\lambda_{\min}(H^{-1}(s)) \to 0$. This leads, for square transfer functions, to the following Newton scheme (cf. (6.2.13)):

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

where $(\mu_{\min}, \mathbf{u}, \mathbf{v})$ is the eigentriplet of $H^{-1}(s_k)$ corresponding to $\lambda_{\min}(H^{-1}(s_k))$, and

$$Q(\lambda) = \lambda^pA_p + \lambda^{p-1}A_{p-1} + \cdots + \lambda A_1 + A_0.$$

An algorithm for computing the dominant poles of a MIMO transfer function can readily be derived from Alg. 6.1. The reader is referred to [93] for the initial MIMO DPA algorithm and to [122] for an algorithm similar to SADPA, generalizations to non-square MIMO systems and more details.

6.6 Conclusions

The Quadratic Dominant Pole Algorithm (QDPA) presented in this chapter is an efficient and effective method for the computation of dominant poles of transfer functions of large-scale second-order dynamical systems. Subspace acceleration improves the global convergence, while the inexpensive deflation strategy makes QDPA able to compute more than one dominant pole automatically, without the risk of computing already found poles again. Another advantage of QDPA is that no linearization of the system is needed, since QDPA operates on the original system matrices.

The dominant poles and corresponding left and right eigenvectors can be used to construct structure-preserving modal equivalents. The dominant eigenspaces can be combined with (second-order) Krylov subspaces (SOAR) to produce reduced-order models of better quality than computed by both methods independently. Furthermore, interpolation points for rational second-order Krylov methods can be based on the (imaginary part) of the dominant poles. Numerical experiments confirmed that accurate reduced-order models can be computed this way.

QDPA can be generalized to MIMO systems and higher-order systems, and can be used for the computation of dominant zeros as well.

Addendum

This chapter is also available as [124]

> Joost Rommes and Nelson Martins, *Efficient computation of transfer function dominant poles of large second-order dynamical systems*, Preprint 1360, Utrecht University, 2007,

and has been submitted for publication.