

# Computation of Stability Radii for Large-Scale Dissipative Hamiltonian Systems

Nicat Aliyev      Volker Mehrmann      Emre Mengi

August 9, 2018

## Abstract

A linear time-invariant dissipative Hamiltonian (DH) system  $\dot{x} = (J - R)Qx$ , with a skew-Hermitian  $J$ , an Hermitian positive semi-definite  $R$ , and an Hermitian positive definite  $Q$ , is always Lyapunov stable and under weak further conditions even asymptotically stable. In various applications there is uncertainty on the system matrices  $J, R, Q$ , and it is desirable to know whether the system remains asymptotically stable uniformly against all possible uncertainties within a given perturbation set. Such robust stability considerations motivate the concept of stability radius for DH systems, i.e., what is the maximal perturbation permissible to the coefficients  $J, R, Q$ , while preserving the asymptotic stability. We consider two stability radii, the unstructured one where  $J, R, Q$  are subject to unstructured perturbation, and the structured one where the perturbations preserve the DH structure. We employ characterizations for these radii that have been derived recently in [*SIAM J. Matrix Anal. Appl.*, 37, pp. 1625-1654, 2016] and propose new algorithms to compute these stability radii for large scale problems by tailoring subspace frameworks that are interpolatory and guaranteed to converge at a super-linear rate in theory. At every iteration, they first solve a reduced problem and then expand the subspaces in order to attain certain Hermite interpolation properties between the full and reduced problems. The reduced problems are solved by means of the adaptations of existing level-set algorithms for  $\mathcal{H}_\infty$ -norm computation in the unstructured case, while, for the structured radii, we benefit from algorithms that approximate the objective eigenvalue function with a piece-wise quadratic global underestimator. The performance of the new approaches is illustrated with several examples including a system that arises from a finite-element modeling of an industrial disk brake.

**Key words.** Linear Time-Invariant Dissipative Hamiltonian System, Port-Hamiltonian system, Robust Stability, Stability Radius, Eigenvalue Optimization, Subspace Projection, Structure Preserving Subspace Framework, Hermite Interpolation.

**AMS subject classifications.** 65F15, 93D09, 93A15, 90C26

# 1 Introduction

Linear time-invariant *Dissipative Hamiltonian (DH) systems* are dynamical systems of the form

$$\dot{x} = (J - R)Qx. \quad (1.1)$$

They arise as homogeneous part of *port-Hamiltonian (PH) systems* of the form

$$\begin{aligned} \dot{x} &= (J - R)Qx(t) + (B - P)u(t), \\ y(t) &= (B + P)^*Qx(t) + Du(t), \end{aligned} \quad (1.2)$$

when the input  $u$  is 0 and the output  $y$  is not considered. Here  $Q = Q^H \in \mathbb{C}^{n \times n}$  is an Hermitian positive definite matrix (denoted as  $Q > 0$ ),  $J \in \mathbb{C}^{n \times n}$  is a skew-Hermitian matrix associated with the energy flux of the system,  $R \in \mathbb{C}^{n \times n}$  is a Hermitian positive semi-definite (denoted by  $R \geq 0$ ) *dissipation matrix* of the system,  $B \pm P \in \mathbb{C}^{n \times m}$  are the *port matrices*, and  $D$  describes the *direct feed-through* from input to output. The function  $\mathcal{H}(x) = \frac{1}{2}x^H Qx$  (called *Hamiltonian function*) describes the total internal energy of the system. Here and elsewhere  $A^H$  denotes the conjugate transpose of a complex matrix  $A$ .

PH and DH systems play an essential role in most areas of science and engineering, see e.g. [12, 20], due to their very important structural properties; e.g., they allow modularized modeling and easy model reduction via Galerkin projection. An important structural property is that DH systems are automatically *Lyapunov stable*, i.e., all eigenvalues of  $A = (J - R)Q$  are in the closed left half of the complex plane, and those on the imaginary axis are semisimple, see [15]. However, DH systems are not necessarily *asymptotically stable*, since  $A$  may have purely imaginary eigenvalues, e.g., when the dissipation matrix  $R$  vanishes, then all eigenvalues are purely imaginary. If a DH system is Lyapunov stable but not asymptotically stable, then arbitrarily small unstructured perturbations (such as rounding errors) may cause the system to become unstable.

These issues are our motivation to analyse whether a DH system is *robustly asymptotically stable*, i.e., whether small (structured or unstructured) perturbations keep it asymptotically stable.

**Example 1.1.** Disk brake squeal is a well-known problem in mechanical engineering. It occurs due to self-excited vibrations caused by instability at the pad-rotor interface [1]. The transition from stability to instability of the brake system is generally examined by finite element (FE) analysis of the system. In [7] FE models resulting for disk brakes are derived in form of second order differential equations

$$M\ddot{x} + D(\Omega)\dot{x} + K(\Omega)x = f, \quad (1.3)$$

with large and sparse coefficient matrixes  $M$ ,  $D(\Omega)$ , and  $K(\Omega) \in \mathbb{R}^{n \times n}$ , where  $D(\Omega)$  and  $K(\Omega)$  depend on the rotational speed  $\Omega > 0$  of the disk, that have the form

$$D(\Omega) := D_M + \frac{1}{\Omega}D_R, \quad K(\Omega) := K_E + \Omega^2 K_g$$

with  $D_M, D_R$  representing material, friction-induced damping matrices, and  $K_E, K_g$  corresponding to elastic, geometric stiffness matrices, respectively. Here,

$M > 0$  and  $K(\Omega) > 0$ , whereas  $D(\Omega) \geq 0$ . (The function  $f$  represents a forcing term or control, but for the stability analysis one may assume that  $f = 0$ , which we assume in the following.) The incorporation of gyroscopic effects, modeled by the term  $G(\Omega)\dot{x}$ , with  $G(\Omega) := \Omega D_G = -\Omega D_G^H$ , and circulatory effects, modeled by an unsymmetric term  $Nx$  gives rise to a system

$$M\ddot{x} + (D(\Omega) + G(\Omega))\dot{x} + (K(\Omega) + N)x = 0, \quad (1.4)$$

or in first order representation  $\widetilde{M}\dot{z} + \widetilde{K}z = 0$ , where

$$\widetilde{M} = \begin{bmatrix} M & 0 \\ 0 & K(\Omega) \end{bmatrix}, \quad \widetilde{K} = \begin{bmatrix} D(\Omega) + G(\Omega) & K(\Omega) + N \\ -K(\Omega) & 0 \end{bmatrix}. \quad (1.5)$$

Straightforward manipulations yields a system

$$\dot{\widetilde{z}} = (J - R)Q\widetilde{z} \quad (1.6)$$

with  $\widetilde{z} = Q^{-1}z$ , where

$$J = \begin{bmatrix} -G(\Omega) & -(K(\Omega) + \frac{1}{2}N) \\ K(\Omega) + \frac{1}{2}N^H & 0 \end{bmatrix}, \quad (1.7)$$

$$R = \begin{bmatrix} D(\Omega) & \frac{1}{2}N \\ \frac{1}{2}N^H & 0 \end{bmatrix}, \quad Q = \begin{bmatrix} M & 0 \\ 0 & K(\Omega) \end{bmatrix}^{-1}.$$

In the absence of the circulatory effects, i.e., when  $N = 0$ , the system in (1.6) is a DH system and as a result it is Lyapunov stable and typically even asymptotically stable. However, small circulatory effects, i.e., perturbations by an  $N$  of small norm, may result in instability.

Asymptotic stability of a general linear dynamical system in the presence of uncertainty can only be guaranteed when the system has a reasonable *distance to instability*, i.e., to systems with purely imaginary eigenvalues. Hence, an estimation of the distance to instability, which is an optimization problem over admissible perturbations, is an important ingredient of a proper stability analysis.

In this paper we focus on the stability analysis of large scale (and typically sparse) DH systems of the form (1.1) in the presence of uncertainties in the coefficients. Considering perturbations in one of the coefficient matrices  $J$ ,  $R$ ,  $Q$  of (1.1), in [15] characterizations for several structured distances to instability were derived under restricted perturbations of the form  $B\Delta C$ , with restriction matrices  $B \in \mathbb{C}^{n \times m}$  and  $C \in \mathbb{C}^{p \times n}$  of full column rank and full row rank, respectively, allowing selected parts of the matrices  $J, R, Q$  to be unperturbed. We will use an adaptation of the subspace framework introduced in [2], based on model order reduction techniques to compute the stability radii using the characterizations in [15].

The paper is organized as follows. Section 2 provides formal definitions of the structured and unstructured stability radii, and in Section 3 we briefly recall the characterizations of these stability radii derived in [15]. Section 4 proposes

subspace frameworks for computing the unstructured stability radii problems exploiting these characterizations. The performance of the proposed frameworks for the unstructured stability radii is illustrated via the disk brake example and several synthetic examples in Section 4.3. Finally, Section 5 focuses on the structured stability radius when only  $R$  is subject to Hermitian perturbations. We first discuss how small-scale problems can be solved in Section 5.1. A new structured subspace framework is discussed in Section 5.2 followed by several numerical examples in Section 5.3.

## 2 Unstructured and Structured Stability Radii

In [15] computable formulas for DH systems of the form (1.1) are derived using several notions of unstructured and structured stability radii. In this section we briefly recall the main definitions and results from [15] for restricted perturbations in one of the following forms.

$$((J + B\Delta C) - R)Q, (J - (R + B\Delta C))Q, \text{ or } (J - R)(Q + B\Delta C). \quad (2.1)$$

In the following  $i\mathbb{R}$  denotes the imaginary axis in the complex plane,  $\Lambda(A)$  the spectrum of a matrix  $A$ , and  $\|A\|_2$  the spectral norm.

**Definition 2.1.** Consider a DH system of the form (1.1) and suppose that  $B \in \mathbb{C}^{n \times m}$  and  $C \in \mathbb{C}^{p \times n}$  are given full rank restriction matrices.

- (i) The unstructured restricted stability radius  $r(J; B, C)$  with respect to perturbations of  $J$  under the restriction matrices  $B, C$  is defined by

$$r(J; B, C) := \inf\{\|\Delta\|_2 : \Delta \in \mathbb{C}^{m \times p}, \Lambda((J + B\Delta C) - R)Q \cap i\mathbb{R} \neq \emptyset\}.$$

- (ii) The unstructured restricted stability radius  $r(R; B, C)$  with respect to perturbations of  $R$  under the restriction matrices  $B, C$  is defined by

$$r(R; B, C) := \inf\{\|\Delta\|_2 : \Delta \in \mathbb{C}^{m \times p}, \Lambda((J - (R + B\Delta C))Q \cap i\mathbb{R} \neq \emptyset\}.$$

- (iii) The unstructured restricted stability radius  $r(Q; B, C)$  with respect to perturbations of  $Q$  under the restriction matrices  $B, C$  is defined by

$$r(Q; B, C) := \inf\{\|\Delta\|_2 : \Delta \in \mathbb{C}^{m \times p}, \Lambda((J - R)(Q + B\Delta C) \cap i\mathbb{R} \neq \emptyset\}.$$

**Example 2.2.** Consider again Example 1.1. Here it is of interest to know whether (for given  $\Omega$ ) the norm of the non-symmetric matrix  $N$  is tolerable to preserve the asymptotic stability of the DH system in (1.6) without the circulatory effects. The relevant stability radius for a specified  $\Omega$  is given by

$$\inf \left\{ \|N\|_2 \mid \Lambda(\mathcal{A}(N)) \cap i\mathbb{R} \neq \emptyset \right\}, \quad (2.2)$$

where

$$\begin{aligned}\mathcal{A}(N) &:= \left( \begin{bmatrix} -G(\Omega) & -(K(\Omega) + \frac{1}{2}N) \\ K(\Omega) + \frac{1}{2}N^H & 0 \end{bmatrix} - \begin{bmatrix} D(\Omega) & \frac{1}{2}N \\ \frac{1}{2}N^H & 0 \end{bmatrix} \right) \begin{bmatrix} M & 0 \\ 0 & K(\Omega) \end{bmatrix}^{-1} \\ &= \left\{ \begin{bmatrix} -G(\Omega) & -K(\Omega) \\ K(\Omega) & 0 \end{bmatrix} - \left( \begin{bmatrix} D(\Omega) & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & N \\ 0 & 0 \end{bmatrix} \right) \right\} \begin{bmatrix} M & 0 \\ 0 & K(\Omega) \end{bmatrix}^{-1}.\end{aligned}$$

Hence, the stability radius in (2.2) corresponds to the unstructured stability radius  $r(R; B, C)$  with the restriction matrices  $B = [I \ 0]^T$  and  $C = [0 \ I]$  with  $n \times n$  blocks.

Furthermore, in the definition of  $\mathcal{A}(N)$  the skew-Hermitian perturbations are more influential on the imaginary parts of its eigenvalues, whereas the Hermitian perturbations are more effective in moving its eigenvalues towards the imaginary axis. This leads us to the consideration of the stability radius

$$\inf \left\{ \|N\|_2 \mid \Lambda(\mathcal{A}_0(N)) \cap i\mathbb{R} \neq \emptyset \right\} \quad (2.3)$$

with

$$\mathcal{A}_0(N) := \left( \begin{bmatrix} -G(\Omega) & -K(\Omega) \\ K(\Omega) & 0 \end{bmatrix} - \begin{bmatrix} D(\Omega) & \frac{1}{2}N \\ \frac{1}{2}N^H & 0 \end{bmatrix} \right) \begin{bmatrix} M & 0 \\ 0 & K(\Omega) \end{bmatrix}^{-1}.$$

Examples such as Example 2.2 motivate the following definition of the structured stability radius in [15].

**Definition 2.3.** Consider a DH system of the form (1.1) and suppose that  $B \in \mathbb{C}^{n \times m}$  is a given restriction matrix. The structured restricted stability radius with respect to Hermitian perturbations of  $R$  under the restriction  $B$  is defined by

$$\begin{aligned}r^{\text{Herm}}(R; B) &:= \inf \{ \|\Delta\|_2 \mid \Delta = \Delta^H, \\ &\quad \Lambda((J - R)Q - (B\Delta B^H)Q) \cap i\mathbb{R} \neq \emptyset \}. \end{aligned} \quad (2.4)$$

### 3 Characterizations for Stability Radii

The numerical techniques that we will derive for the computation of the unstructured and structured stability radii exploit eigenvalue or singular value optimization characterizations derived in [15].

**Theorem 3.1.** For an asymptotically stable DH system of the form (1.1) and restriction matrices  $B \in \mathbb{C}^{n \times m}$ ,  $C \in \mathbb{C}^{p \times n}$  the following assertions hold:

- (i) The unstructured stability radius  $r(R; B, C)$  is finite if and only if  $G_R(\omega) := CQ(i\omega I_n - (J - R)Q)^{-1}B$  is not identically zero if and only if  $r(J; B, C)$  is finite. If  $r(R; B, C)$  is finite, then we have

$$r(R; B, C) = r(J; B, C) = \inf_{\omega \in \mathbb{R}} \frac{1}{\|G_R(\omega)\|_2}. \quad (3.1)$$

(ii) The unstructured stability radius  $r(Q; B, C)$  is finite if and only if  $G_Q(\omega) := C(i\omega I_n - (J - R)Q)^{-1}(J - R)B$  is not identically zero for all  $\omega \in \mathbb{R}$ . If  $r(Q; B, C)$  is finite, then we have

$$r(Q; B, C) = \inf_{\omega \in \mathbb{R}} \frac{1}{\|G_Q(\omega)\|_2}. \quad (3.2)$$

For the structured stability radius and Hermitian perturbations of  $R$  the following result is obtained in [15].

**Theorem 3.2.** For an asymptotically stable DH system of the form (1.1), and a restriction matrix  $B \in \mathbb{C}^{n \times m}$  of full column rank, let

1.  $W(\lambda) := (J - R)Q - \lambda I$  for a given  $\lambda \in \mathbb{C}$  such that  $W(\lambda)$  is invertible,
2.  $L(\lambda)$  be a lower triangular Cholesky factor of

$$\tilde{H}_0(\lambda) := B^H W(\lambda)^{-H} Q B B^H Q W(\lambda)^{-1} B,$$

i.e.,  $L(\lambda)$  is a lower triangular matrix satisfying  $\tilde{H}_0(\lambda) = L(\lambda)L(\lambda)^H$ ,

3.  $H_0(\lambda) := L(\lambda)^{-1} L(\lambda)^{-H}$ ,
4.  $H_1(\lambda) := i(\tilde{H}_1(\lambda) - \tilde{H}_1(\lambda)^H)$ , where  $\tilde{H}_1(\lambda) := L(\lambda)^{-1} B^H W(\lambda)^{-H} Q B L(\lambda)^{-H}$ .

Then  $r^{\text{Herm}}(R; B)$  is finite, and given by

$$r^{\text{Herm}}(R; B) = \left\{ \inf_{\omega \in \mathbb{R}} \sup_{t \in \mathbb{R}} \lambda_{\min}(H_0(i\omega) + t H_1(i\omega)) \right\}^{1/2},$$

where the inner supremum is attained if and only if  $H_1(i\omega)$  is indefinite.

The characterization in [15] is presented in a slightly different form. In particular, it is stated in terms of an orthonormal basis  $U(\lambda)$  for the kernel of  $((I - BB^+)W(\lambda))$ . It turns out that  $U(\lambda)$  does not have to be orthonormal, rather the theorem can be stated in terms of any basis for the kernel of  $((I - BB^+)W(\lambda))$ . In Theorem 3.2, we have employed a particular basis that simplifies the formulas and facilitates the computation.

## 4 Computation of the Unstructured Stability Radii for Large-Scale Problems

In this section we study the computation of unstructured stability radii for large-scale DH systems using the characterizations of  $r(R; B, C)$ ,  $r(Q; B, C)$ ,  $r(J; B, C)$  given in Theorem 3.1. One easily observes that

$$\begin{aligned} G_R(i\omega) &:= CQ(i\omega I_n - (J - R)Q)^{-1}B, \\ G_Q(i\omega) &:= C(i\omega I_n - (J - R)Q)^{-1}(J - R)B \end{aligned}$$

can be viewed as restrictions of transfer functions of control systems to the imaginary axis. To be precise, setting  $\tilde{A} = (J - R)Q$ ,  $\tilde{B} = B$  and  $\tilde{C} = CQ$ , the matrix-valued function  $G_R(i\omega) := CQ(i\omega I_n - (J - R)Q)^{-1}B$  becomes

$$\tilde{G}_R(i\omega) := \tilde{C}(i\omega I_n - \tilde{A})^{-1}\tilde{B} \quad (4.1)$$

which can be considered as the transfer function of the system

$$\dot{x} = \tilde{A}x + \tilde{B}u, \quad y = \tilde{C}x \quad (4.2)$$

on the imaginary axis. Theorem 3.1 suggests that if  $\tilde{G}_R(i\omega) := \tilde{C}(i\omega I_n - \tilde{A})^{-1}\tilde{B}$  is not identically zero, then  $r(R; B, C)$  and  $r(J; B, C)$  are finite, and characterized by

$$\begin{aligned} r(R; B, C) &= r(J; B, C) = \inf_{\omega \in \mathbb{R}} \frac{1}{\|\tilde{G}_R(i\omega)\|_2} \\ &= \frac{1}{\sup_{\omega \in \mathbb{R}} \|\tilde{G}_R(i\omega)\|_2} = \frac{1}{\|\tilde{G}_R\|_{\mathcal{H}_\infty}}, \end{aligned} \quad (4.3)$$

where  $\|\tilde{G}_R\|_{\mathcal{H}_\infty} := \sup_{\omega \in \mathbb{R}} \sigma_{\max}(\tilde{G}_R(i\omega))$  denotes the  $\mathcal{H}_\infty$ -norm of  $\tilde{G}_R$ , and  $\sigma_{\max}(\cdot)$  denotes the maximal singular value.

For the stability radius  $r(Q; B, C)$ , consideration of  $G_Q(i\omega) := C(i\omega I_n - (J - R)Q)^{-1}(J - R)B$ , by setting  $\tilde{A} = (J - R)Q$ ,  $\tilde{B} = (J - R)B$  and  $\tilde{C} = C$ , leads us to a similar characterization.

## 4.1 A Subspace Framework

Recently, in [2], a subspace framework for the computation of the  $\mathcal{H}_\infty$ -norm of a large-scale system has been proposed, which is inspired from model order reduction techniques, and has made the computation of  $\mathcal{H}_\infty$ -norms feasible for very large control systems. We will now discuss how to use these techniques for the computation of the unstructured stability radii  $r(R; B, C)$ ,  $r(J; B, C)$ ,  $r(Q; B, C)$  in the large-scale setting.

To briefly summarize the iterative procedure in the subspace framework of [2], let us assume that in iteration  $k$ , two subspaces  $\mathcal{V}_k$  and  $\mathcal{W}_k$  of equal dimension have been determined, as well as matrices  $V_k$  and  $W_k$  whose columns span orthonormal bases for these subspaces. Applying a Petrov-Galerkin projection to system (4.2), restricts the state  $x$  to  $\mathcal{V}_k$ , i.e., in (4.2) we replace  $x$  by  $V_k x_k$ , and imposes that the residual after this restriction is orthogonal to  $\mathcal{W}_k$ . This projection gives rise to a reduced order system

$$\dot{x}_k = \tilde{A}_k x_k + \tilde{B}_k u, \quad y_k = \tilde{C}_k x_k, \quad (4.4)$$

with

$$\tilde{A}_k := W_k^H \tilde{A} V_k, \quad \tilde{B}_k := W_k^H \tilde{B}, \quad \tilde{C}_k = \tilde{C} V_k. \quad (4.5)$$

Then the  $\mathcal{H}_\infty$ -norm of a transfer function  $G(s) := \tilde{C}(sI_n - \tilde{A})^{-1}\tilde{B}$  in (4.2) can be approximated by computing the  $\mathcal{H}_\infty$ -norm of

$$G_k(s) := \tilde{C}_k(sI_k - \tilde{A}_k)^{-1}\tilde{B}_k$$

for instance by employing the method in [4] or [5], in particular, by computing  $\omega_{k+1} := \arg \max_{\omega \in \mathbb{R}} \sigma_{\max}(G_k(i\omega))$ . This is computationally cheap if the dimensions of  $\mathcal{V}_k, \mathcal{W}_k$  are small. Once  $\omega_{k+1}$  has been computed, then the subspaces  $\mathcal{V}_k$  and  $\mathcal{W}_k$  are expanded into larger subspaces  $\mathcal{V}_{k+1}$  and  $\mathcal{W}_{k+1}$  in such a way that the corresponding reduced transfer function  $G_{k+1}(s)$  satisfies the *Hermite interpolation conditions*

$$\begin{aligned}\sigma_{\max}(G(i\omega_{k+1})) &= \sigma_{\max}(G_{k+1}(i\omega_{k+1})), \\ \sigma'_{\max}(G(i\omega_{k+1})) &= \sigma'_{\max}(G_{k+1}(i\omega_{k+1})),\end{aligned}\tag{4.6}$$

where  $\sigma'_{\max}(G(i\omega))$  denotes the derivative of  $\sigma_{\max}(G(i\omega))$  with respect to  $\omega$ . Denoting the image space of a matrix by  $A$  by  $\text{Im}(A)$ , it is shown in [2] that

$$\begin{aligned}\mathcal{V}_{k+1} &:= \mathcal{V}_k \oplus \text{Im}((i\omega_{k+1}I_n - \tilde{A})^{-1}\tilde{B}), \\ \mathcal{W}_{k+1} &:= \mathcal{W}_k \oplus \text{Im}((\tilde{C}(i\omega_{k+1}I_n - \tilde{A})^{-1})^H),\end{aligned}$$

more specifically the inclusions

$$\text{Im}((i\omega_{k+1}I_n - \tilde{A})^{-1}\tilde{B}) \subseteq \mathcal{V}_{k+1}, \quad \text{Im}((\tilde{C}(i\omega_{k+1}I_n - \tilde{A})^{-1})^H) \subseteq \mathcal{W}_{k+1},$$

ensure that the Hermite interpolation conditions (4.6) are satisfied. The procedure is then repeated with the expanded subspaces  $\mathcal{V}_{k+1}, \mathcal{W}_{k+1}$ . In [2], it is shown that the sequence  $\{\omega_k\}$  converges at a super-linear rate and satisfies

$$\begin{aligned}\sigma_{\max}(G(i\omega_j)) &= \sigma_{\max}(G_k(i\omega_j)), \\ \sigma'_{\max}(G(i\omega_j)) &= \sigma'_{\max}(G_k(i\omega_j))\end{aligned}$$

for  $j = 1, \dots, k$ .

A disadvantage of this general approach is that even if  $\tilde{A} = (J - R)Q$  has DH structure, this is not necessarily true for  $\tilde{A}_k$ , so it cannot be guaranteed from the structure that the reduced system is stable. In the next section we modify the procedure of [2] to preserve the DH structure.

## 4.2 A Structure Preserving Subspace Framework

In this subsection we derive an interpolating, DH structure preserving version of the robust subspace projection framework. Structure preserving subspace projection methods in the context of model order reduction of large-scale PH and DH systems have been proposed in [10, 11, 17, 18, 21, 22]. Our approach is inspired by [10] and uses a general interpolation result from [6].

**Theorem 4.1.** *Let  $G(s)$  be the transfer function for a full order system as in (4.2), and let  $G_k(s)$  be the transfer function for the reduced system defined by (4.4), (4.5).*

(i) (Right Tangential Interpolation) *For given  $\hat{s} \in \mathbb{C}$  and  $\hat{b} \in \mathbb{C}^m$ , if*

$$\left[ (\hat{s}I - \tilde{A})^{-1} \right]^\ell \tilde{B}\hat{b} \in \mathcal{V}_k \quad \text{for } \ell = 1, \dots, N, \tag{4.7}$$

and  $\mathcal{W}_k$  is such that  $W_k^H V_k = I$ , then we have

$$G^{(\ell)}(\hat{s})\hat{b} = G_k^{(\ell)}(\hat{s})\hat{b} \quad \text{for } \ell = 0, \dots, N-1 \quad (4.8)$$

provided that both  $\hat{s}I - \tilde{A}$  and  $\hat{s}I - \tilde{A}_k$  are invertible.

(ii) (Left Tangential Interpolation) For a given  $\hat{s} \in \mathbb{C}$  and  $\hat{c} \in \mathbb{C}^p$ , if

$$\left( \hat{c}^H \tilde{C} \left[ (\hat{s}I - \tilde{A})^{-1} \right]^\ell \right)^H \in \mathcal{W}_k \quad \text{for } \ell = 1, \dots, N, \quad (4.9)$$

and  $\mathcal{V}_k$  is such that  $W_k^H V_k = I$ , then we have

$$\hat{c}^H G^{(\ell)}(\hat{s}) = \hat{c}^H G_k^{(\ell)}(\hat{s}) \quad \text{for } \ell = 0, \dots, N-1 \quad (4.10)$$

provided that both  $\hat{s}I - \tilde{A}$  and  $\hat{s}I - \tilde{A}_k$  are invertible.

#### 4.2.1 Computation of $r(R; B, C)$ and $r(J; B, C)$

The computation of  $r(R; B, C) = r(J; B, C)$  involves the maximization of the largest singular value of the transfer function  $G(s) = CQ(sI - (J - R)Q)^{-1}B$  associated with the system

$$\begin{aligned} \dot{x} &= (J - R)Qx + Bu, \\ y &= CQx. \end{aligned} \quad (4.11)$$

on the imaginary axis. We make use of Theorem 4.1 to obtain a reduced order system satisfying the interpolation conditions (4.8) while retaining the structure in (4.11). We, in particular, employ right tangential interpolation for a given  $\hat{s} \in \mathbb{C}$  and  $\hat{b} \in \mathbb{C}^m$ , and choose  $\mathcal{V}_k$  as any subspace satisfying (5.8). Let us also define  $W_k := QV_k(V_k^H QV_k)^{-1}$ ,  $\mathcal{W}_k := \text{Im}(W_k)$ , so that

$$W_k^H V_k = I_k \quad \text{and} \quad (W_k V_k^H)^2 = W_k V_k^H,$$

i.e.,  $W_k V_k^H$  is an oblique projector onto  $\text{Im}(QV_k)$ .

The matrices  $\tilde{A}_k, \tilde{B}_k, \tilde{C}_k$  of the reduced system (4.4), (4.5) for these choices of  $V_k$  and  $W_k$  then satisfy

$$\begin{aligned} \tilde{A}_k &= W_k^H \tilde{A} V_k = W_k^H (J - R)QV_k = W_k^H (J - R)W_k V_k^H QV_k \\ &= (J_k - R_k)Q_k, \end{aligned} \quad (4.12)$$

where  $J_k := W_k^H J W_k = -J_k^H$ ,  $R_k := W_k^H R W_k = R_k^H \geq 0$ , and  $Q_k := V_k^H Q V_k = Q_k^H > 0$ . Additionally,  $\tilde{C}_k = \tilde{C} V_k = C Q V_k = C W_k V_k^H Q V_k = C_k Q_k$ , where  $C_k := C W_k$ , and  $\tilde{B}_k = W_k^H B =: B_k$ .

This construction leads to the following result of [10].

**Theorem 4.2.** *Consider a linear system of the form (4.11) with transfer function  $G(s) := CQ(sI_n - (J - R)Q)^{-1}B$ . Furthermore, for a given point  $\hat{s} \in \mathbb{C}$  and*

a given tangent direction  $\hat{b} \in \mathbb{C}^m$ , suppose that  $V_k$  is a matrix with orthonormal columns such that

$$(\hat{s}I_n - (J - R)Q)^{-(\ell-1)}(\hat{s}I_n - (J - R)Q)^{-1}B\hat{b} \in \text{Im}(V_k) \quad \text{for } \ell = 1, \dots, N.$$

Define  $W_k := QV_k(V_k^H QV_k)^{-1}$  and set

$$\begin{aligned} J_k &:= W_k^H J W_k, \quad Q_k := V_k^H Q V_k, \quad R_k := W_k^H R W_k \\ B_k &:= W_k^H B, \quad C_k := C W_k. \end{aligned} \quad (4.13)$$

Then the resulting reduced order model

$$\begin{aligned} \dot{x}_k &= (J_k - R_k)Q_k x_k + B_k u, \\ y_k &= C_k Q_k x_k \end{aligned} \quad (4.14)$$

is a DH system with transfer function

$$G_k(s) := C_k Q_k (sI_k - (J_k - R_k)Q_k)^{-1} B_k \quad (4.15)$$

that satisfies

$$G^{(j)}(\hat{s})\hat{b} = G_k^{(j)}(\hat{s})\hat{b} \quad \text{for } j = 0 \dots, N-1, \quad (4.16)$$

where  $G^{(j)}(\hat{s})$  denotes the  $j$ -th derivative of  $G(s)$  at the point  $\hat{s}$ .

Based on Theorem 4.2 we obtain the following algorithm for the computation of the stability radii for DH systems. According to Theorem 4.2, for a given

---

**Algorithm 1** DH structure preserving subspace method for the computation of the stability radii  $r(R; B, C)$  and  $r(J; B, C)$  for large-scale systems.

---

**Input:** Matrices  $B \in \mathbb{C}^{n \times m}$ ,  $C \in \mathbb{C}^{p \times n}$ ,  $J, R, Q \in \mathbb{C}^{n \times n}$ .

**Output:** The sequence of frequencies  $\{\omega_k\}$ .

- 1: Choose initial interpolation points  $\omega_1, \dots, \omega_q \in \mathbb{R}$ .
  - 2:  $V_q \leftarrow \text{orth} \begin{bmatrix} D(i\omega_1)^{-1}B & D(i\omega_1)^{-2}B & \dots & D(i\omega_q)^{-1}B & D(i\omega_q)^{-2}B \end{bmatrix}$   
and  $W_q \leftarrow QV_q(V_q^H QV_q)^{-1}$ .
  - 3: **for**  $k = q, q+1, \dots$  **do**
  - 4:   Form  $G_k$  as in (4.15) for the choices of  $J_k, R_k, Q_k, B_k, C_k$  in (4.13).
  - 5:    $\omega_{k+1} \leftarrow \arg \max_{\omega \in \mathbb{R}} \sigma_{\max}(G_k(i\omega))$ .
  - 6:    $\hat{V}_{k+1} \leftarrow \begin{bmatrix} D(i\omega_{k+1})^{-1}B & D(i\omega_{k+1})^{-2}B \end{bmatrix}$
  - 7:    $V_{k+1} \leftarrow \text{orth} \left( \begin{bmatrix} V_k & \hat{V}_{k+1} \end{bmatrix} \right)$  and  $W_{k+1} \leftarrow QV_{k+1}(V_{k+1}^H QV_{k+1})^{-1}$ .
  - 8: **end for**
- 

$\hat{s} \in \mathbb{C}$ , setting

$$V_k := \begin{bmatrix} D(\hat{s})^{-1}B & D(\hat{s})^{-2}B \end{bmatrix}, \quad W_k := QV_k(V_k^H QV_k)^{-1}$$

where

$$D(\hat{s}) := (\hat{s}I_n - (J - R)Q), \quad (4.17)$$

we obtain  $G(\hat{s}) = G_k(\hat{s})$  and  $G'(\hat{s}) = G'_k(\hat{s})$  and thus the Hermite interpolation conditions

$$\sigma_{\max}(G(\hat{s})) = \sigma_{\max}(G_k(\hat{s})), \quad \sigma'_{\max}(G(\hat{s})) = \sigma'_{\max}(G_k(\hat{s})) \quad (4.18)$$

are satisfied, which suggest the use of the reduced system in the greedy subspace framework outlined in Algorithm 1.

In line 5 of every iteration, the subspace framework computes the  $\mathcal{H}_\infty$ -norm of a reduced system, in particular it computes the point  $i\omega_*$  on the imaginary axis where this  $\mathcal{H}_\infty$ -norm is attained. Then the current left and right subspaces are expanded in a way so that the resulting reduced system still has DH structure and its transfer function Hermite interpolates the original transfer function at  $i\omega_*$ . Since the Hermite interpolation conditions (4.18) are satisfied in  $\hat{s} = i\omega_1, \dots, i\omega_k$  at the end of iteration  $k$ , the rate-of-convergence analysis in [2] applies to deduce a superlinear rate-of-convergence for the sequence  $\{\omega_k\}$ .

The computationally most expensive part of Algorithm 1 is in lines 2 and 6, where many linear systems with possibly many right hand sides have to be solved. If this is done with a direct solver, then for each value  $\hat{\omega} \in \mathbb{R}$  one  $LU$  factorization of the matrix  $D(i\hat{\omega})$  has to be performed. For large values of  $n$ , the computation time is usually dominated by these  $LU$  factorizations. In contrast to this, the solution of the reduced problem in line 5 can be achieved by means of the (for small systems) efficient algorithm in [4, 5].

#### 4.2.2 Computation of $r(Q; B, C)$

To compute the stability radius  $r(Q; B, C)$  in the large-scale setting, we employ left tangential interpolations (i.e., part (ii) of Theorem 4.1).

In this case  $r(Q; B, C)$  is the reciprocal of the  $\mathcal{H}_\infty$ -norm of the transfer function  $G(s) := C(sI - (J - R)Q)^{-1}(J - R)B$  corresponding to the system

$$\dot{x} = (J - R)Qx + (J - R)Bu, \quad y(t) = Cx. \quad (4.19)$$

To obtain a reduced system which has the same structure as (4.19) and has a transfer function  $G_k(s)$  that satisfies  $\hat{c}^H G(\hat{s}) = \hat{c}^H G_k(\hat{s})$  for a given point  $\hat{s} \in \mathbb{C}$  and a direction  $\hat{c} \in \mathbb{C}^p$ , let us choose  $W_k$  so as to satisfy the condition in (4.9) for  $\tilde{A} := (J - R)Q$ , and  $\tilde{C} := C$ . Furthermore, we set

$$V_k := (J - R)^H W_k (W_k^H (J - R)^H W_k)^{-1},$$

as well as  $\mathcal{W}_k := \text{Im}(W_k)$ ,  $\mathcal{V}_k := \text{Im}(V_k)$ . The matrix  $V_k$  is chosen to satisfy

$$V_k^H W_k = I_k \quad \text{and} \quad (V_k W_k^H)^2 = V_k W_k^H,$$

so that  $V_k W_k^H$  is an oblique projector onto  $\text{Im}((J - R)^H W_k)$ .

In (4.19), setting  $\tilde{A} := (J - R)Q$ ,  $\tilde{B} := (J - R)B$ ,  $\tilde{C} = C$ , let us investigate the matrices  $\tilde{A}_k, \tilde{B}_k, \tilde{C}_k$  of the corresponding reduced system defined by (4.4), (4.5). Specifically, we have that

$$\tilde{A}_k = W_k^H \tilde{A} V_k = W_k^H (J - R)Q V_k = W_k^H (J - R)W_k V_k^H Q V_k,$$

where, in the third equality, we employ  $V_k W_k^H (J - R)^H W_k = (J - R)^H W_k$ , or equivalently  $W_k^H (J - R) W_k V_k^H = W_k^H (J - R)$ . Hence, defining  $J_k := W_k^H J W_k = -J_k^H$ ,  $R_k := W_k^H R W_k = R_k^H \geq 0$ ,  $Q_k := V_k^H Q V_k = Q_k^H > 0$ , we obtain a DH system with  $\tilde{A}_k = (J_k - R_k)Q_k$ . We also have

$$\tilde{B}_k = W_k^H \tilde{B} = W_k^H (J - R)B = W_k^H (J - R)W_k V_k^H B = (J_k - R_k)B_k$$

with  $B_k := V_k^H B$ , and  $\tilde{C}_k = C V_k =: C_k$ . These constructions lead to the following analogue of Theorem 4.2.

**Theorem 4.3.** *Consider a linear system of the form (4.19) with the transfer function  $G(s) := C(sI_n - (J - R)Q)^{-1}(J - R)B$ . For a given point  $\hat{s} \in \mathbb{C}$  and a direction  $\hat{c} \in \mathbb{C}^m$ , suppose that  $W_k$  is a matrix with orthonormal columns such that*

$$\left( \hat{c}^H C (\hat{s}I_n - (J - R)Q)^{-1} (\hat{s}I_n - (J - R)Q)^{-(\ell-1)} \right)^H \in \text{Im}(W_k)$$

for  $\ell = 1, \dots, N$ . Letting  $V_k := (J - R)^H W_k (W_k^H (J - R)^H W_k)^{-1}$ , and

$$\begin{aligned} J_k &:= W_k^H J W_k, \quad Q_k := V_k^H Q V_k, \quad R_k := W_k^H R W_k \\ B_k &:= V_k^H B, \quad C_k := C V_k, \end{aligned} \tag{4.20}$$

the resulting reduced order system

$$\dot{x}_k = (J_k - R_k)Q_k x_k + (J_k - R_k)B_k u, \quad y_k = C_k x_k \tag{4.21}$$

is such that  $\dot{x}_k = (J_k - R_k)Q_k x_k$  is dissipative Hamiltonian. Furthermore, the transfer function

$$G_k(s) := C_k(sI_k - (J_k - R_k)Q_k)^{-1}(J_k - R_k)B_k \tag{4.22}$$

of (4.21) satisfies

$$\hat{c}^H G^{(\ell)}(\hat{s}) \hat{b} = \hat{c}^H G_k^{(\ell)}(\hat{s}) \quad \text{for } \ell = 0 \dots, N-1. \tag{4.23}$$

Theorem 4.3 shows that at a given  $\hat{s} \in \mathbb{C}$ , the Hermite interpolation properties  $G(\hat{s}) = G_k(\hat{s})$  and  $G'(\hat{s}) = G'_k(\hat{s})$  and, in particular,  $\sigma_{\max}(G(\hat{s})) = \sigma_{\max}(G_k(\hat{s}))$  and  $\sigma'_{\max}(G(\hat{s})) = \sigma'_{\max}(G_k(\hat{s}))$  can be achieved, while preserving the structure, with the choices

$$\begin{aligned} W_k &= \begin{bmatrix} (CD(\hat{s})^{-1})^H & (CD(\hat{s})^{-2})^H \end{bmatrix}, \\ V_k &= (J - R)^H W_k (W_k^H (J - R)^H W_k)^{-1}, \end{aligned}$$

where  $D(\hat{s})$  is as in (4.17). This in turn gives rise to Algorithm 2.

At every iteration of this algorithm, the  $\mathcal{H}_\infty$ -norm is computed for a reduced problem of the form (4.21), in particular, the optimal frequency where this  $\mathcal{H}_\infty$ -norm is attained is retrieved. Then the subspaces are updated so that the Hermite interpolation properties hold also at this optimal frequency at the largest singular values of the full and reduced problem, respectively. Once again the sequence  $\{\omega_k\}$  by Algorithm 2 is guaranteed to converge at a super-linear rate, which can be attributed to the Hermite interpolation properties holding between the largest singular values of the full and reduced transfer functions.

---

**Algorithm 2** DH structure preserving subspace method for the computation of the stability radius  $r(Q; B, C)$  of a large scale DH system.

---

**Input:** Matrices  $B \in \mathbb{C}^{n \times m}$ ,  $C \in \mathbb{C}^{p \times n}$ ,  $J, R, Q \in \mathbb{C}^{n \times n}$ .

**Output:** The sequence  $\{\omega_k\}$ .

- 1: Choose initial interpolation points  $\omega_1, \dots, \omega_j \in \mathbb{R}$ .
  - 2:  $W_j \leftarrow \text{orth} \begin{bmatrix} (CD(i\omega_1)^{-1})^H & (CD(i\omega_1)^{-2})^H & \dots \\ & (CD(i\omega_j)^{-1})^H & (CD(i\omega_j)^{-2})^H \end{bmatrix}$   
 $V_j \leftarrow (J - R)^H W_j (W_j^H (J - R)^H W_j)^{-1}$ .
  - 3: **for**  $k = j, j + 1, \dots$  **do**
  - 4:   Form  $G_k$  as in (4.22) for the choices of  $J_k, R_k, Q_k, B_k, C_k$  in (4.20).
  - 5:    $\omega_{k+1} \leftarrow \arg \max_{\omega \in \mathbb{R}} \sigma_{\max}(G_k(i\omega))$ .
  - 6:    $\widehat{W}_{k+1} \leftarrow \begin{bmatrix} (CD(i\omega_{k+1})^{-1})^H & (CD(i\omega_{k+1})^{-2})^H \end{bmatrix}$ .
  - 7:    $W_{k+1} \leftarrow \text{orth} \left( \begin{bmatrix} W_k & \widehat{W}_{k+1} \end{bmatrix} \right)$  and  
 $V_{k+1} \leftarrow (J - R)^H W_{k+1} (W_{k+1}^H (J - R)^H W_{k+1})^{-1}$ .
  - 8: **end for**
- 

### 4.3 Numerical Experiments

In this subsection we illustrate the performance of MATLAB implementations of Algorithms 1 and 2 via some numerical examples. We first discuss some implementation details and then present numerical results on two sets of random synthetic examples in Section 4.3.2, and data from a FE model of a brake disk in Section 4.3.3.

#### 4.3.1 Implementation Details and Test Setup

Algorithms 1 and 2 are terminated when at least one of the following three conditions is fulfilled:

1. The relative distance between  $\omega_k$  and  $\omega_{k-1}$  is less than a prescribed tolerance for some  $k > j$ , i.e.,

$$|\omega_k - \omega_{k-1}| < \varepsilon \cdot \frac{1}{2}(\omega_k + \omega_{k-1}).$$

2. Letting  $f_k := \max_{\omega \in \mathbb{R} \cup \infty} \sigma_{\max}(G_k(i\omega))$ , two consecutive iterates  $f_k, f_{k-1}$  are close enough in a relative sense, i.e.,

$$|f_k - f_{k-1}| < \varepsilon \cdot \frac{1}{2}(f_k + f_{k-1}).$$

3. The number of iterations exceeds a specified integer, i.e.,  $k > k_{\max}$ .

In all numerical examples that we present, we set  $\varepsilon = 10^{-6}$  and  $k_{\max} = 100$ .

In general, Algorithms 1 and 2 converge only locally. The choice of the initial interpolation points affects the maximizers that the subspace frameworks

converge to, in particular, whether convergence to a global maximizer occurs. The initial interpolation points are chosen based on the following procedure.

First, we discretize the interval  $[\lambda_{\min}^{\Im}, \lambda_{\max}^{\Im}]$  into  $\rho$  equally spaced points, say  $\omega_{0,1}, \dots, \omega_{0,\rho}$ , including the end-points  $\lambda_{\min}^{\Im}, \lambda_{\max}^{\Im}$ , where  $\rho$  is specified by the user and  $\lambda_{\min}^{\Im}, \lambda_{\max}^{\Im}$  denote the imaginary parts of the eigenvalues of  $(J - R)Q$  with the smallest and largest imaginary part, respectively. Then we approximate the eigenvalues  $z_1, \dots, z_\rho$  of  $(J - R)Q$  closest to  $i\omega_{0,1}, \dots, i\omega_{0,\rho}$ , and permute them into  $z_{j_1}, \dots, z_{j_\rho}$  where  $\{j_1, \dots, j_\rho\} = \{1, \dots, \rho\}$  so that  $\sigma_{\max}(G(i\Im z_{j_1})) \geq \dots \geq \sigma_{\max}(G(i\Im z_{j_\rho}))$ . The initial interpolation points  $\omega_1, \dots, \omega_\ell$  are then chosen as the imaginary parts of  $z_{j_1}, \dots, z_{j_\ell}$ , where again  $\ell \leq \rho$  is specified by the user.

#### 4.3.2 Results on Synthetic Examples

We now present results for two families of linear DH systems with random coefficient matrices; the first family consists of dense systems of order 800, whereas the second family consists of sparse systems of order 5000.

**Dense Random examples.** In the dense family the coefficient matrices  $J$ ,  $Q$ ,  $R$  are formed by the MATLAB commands

```
>> J = randn(800); J = (J - J')/2;
>> Q = randn(800); Q = (Q + Q')/2; mineig = min(eig(Q));
>> if (mineig < 10^-4) Q = Q + (-mineig + 5*rand)*eye(n); end
>> p = round(80*rand);
>> Rp = randn(p); Rp = (Rp + Rp')/2; mineig = min(eig(Rp));
>> if (mineig < 10^-4) Rp = Rp + (-mineig + 5*rand)*eye(p); end
>> R = [Rp zeros(p,800-p); zeros(800-p,p) zeros(800-p,800-p)];
>> X = randn(800); [U,~] = qr(X); R = U'*R*U;
```

The restriction matrices  $B$  and  $C$  are chosen as  $800 \times 2$  and  $2 \times 800$  random matrices created by the MATLAB command `randn`. To compute  $r(R; B, C) = r(J; B, C)$ , as well as  $r(Q; B, C)$ , we ran

- (1) the Boyd-Balakrishnan (BB) algorithm [4],
- (2) the subspace framework that does not preserve the DH structure [2, Algorithm 1] described in Subsection 4.1, and
- (3) the subspace frameworks that preserve structure, i.e., Algorithms 1 and 2, introduced in Subection 4.2

on 100 such random examples. The spectrum of a typical  $(J - R)Q$  of size 800 generated in this way is depicted in Figure 1 on the left.

The progress of Algorithm 2, as well as Algorithm 1 in [2], to compute  $r(Q; B, C)$  for this example is presented in Figure 2, which includes on the top left a plot of  $f(\omega) := \sigma_{\max}(C(i\omega I - (J - R)Q)^{-1}(J - R)B)$  for  $\omega \in [-2000, 0]$  along with the converged maximizers by the respective Algorithms. Algorithm 2 converges to the global maximizer  $\omega_{*,1} = -731.9774$  with  $f(\omega_{*,1}) = 32.321399$ , while Algorithm 1 in [2] converges to the local maximizer  $\omega_{*,2} = -1602.1187$

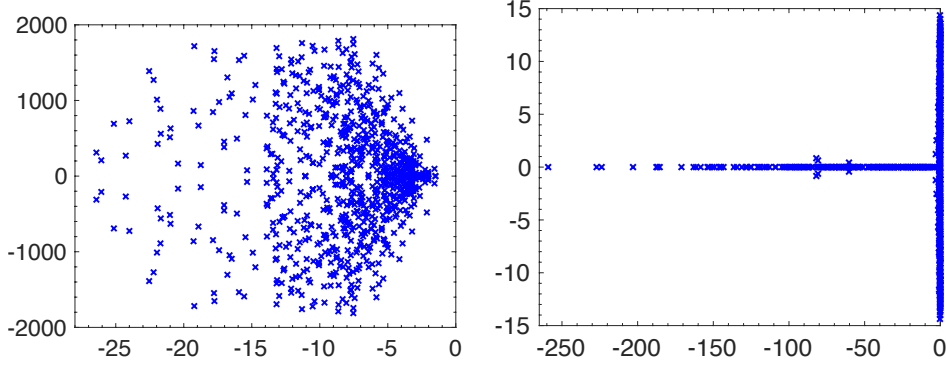


Figure 1: The spectra of  $A = (J - R)Q$  for a dense random  $J, R, Q \in \mathbb{R}^{800 \times 800}$  (left), and a sparse random  $J, R, Q \in \mathbb{R}^{5000 \times 5000}$  (right). The MATLAB commands yielding these  $J, R, Q$  are specified in Section 4.3.2.

with  $f(\omega_{*,2}) = 29.028197$ . The globally optimal peak  $(\omega_{*,1}, f(\omega_{*,1}))$  and the locally optimal peak  $(\omega_{*,2}, f(\omega_{*,2}))$  are marked in the plot with a square and a circle, respectively.

The remaining five plots in Figure 2 illustrate the progress of Algorithm 2. In each one of these plots, the black curve is a plot of the reduced function  $f_k(\omega) := \max_{\omega \in \mathbb{R} \cup \infty} \sigma_{\max}(C_k(i\omega I - (J_k - R_k)Q_k)^{-1}(J_k - R_k)B_k)$  with respect to  $\omega$ , and the circle marks the global maximizer of this reduced function. The top right shows the initial reduced function in black interpolating the full function at ten points, and the other four show the reduced function after iterations 1-4 from middle-left to bottom-right. Observe that, at every iteration, the refined reduced function interpolates the full function at the maximizer of the previous reduced function in addition to the earlier interpolation points. We also list the iterates of Algorithm 2 in Table 1 indicating a quick converge. The algorithm terminates after performing six subspace iterations.

The results of Algorithms 1 and 2 for the first 10 random examples are presented in Tables 2 and 3, respectively. Results from [2, Algorithm 1] and the BB Algorithm [4] are also included in these tables for comparison purposes. For the computation of  $r(J; B, C) = r(R; B, C)$ , the new structure-preserving Algorithm 1 and [2, Algorithm 1] perform equally well on these first 10 examples. They both return the globally optimal solutions in 9 out of 10 examples, perform similar number of subspace iterations and require similar amount of cpu-time.

A more decisive conclusion can be drawn when we consider all of the 100 random examples. The left-hand columns in Figure 3 depict the ratios  $(f_{BB} - f_{SF})/((f_{BB} + f_{SF})/2)$ , where  $f_{BB}$  is the globally maximal value of  $\sigma_{\max}(CQ(i\omega I - (J - R)Q)^{-1}B)$  over  $\omega$  returned by the BB algorithm and  $f_{SF}$  are the values returned by the subspace frameworks, specifically by Algorithm 1 on the top and by [2, Algorithm 1] at the bottom. The results by Algorithm 1 match with the ones by the BB algorithm 81 times out of 100, while the results by Algorithm [2, Algorithm 1] match with the ones by the BB algorithm 67 times out of 100. (In

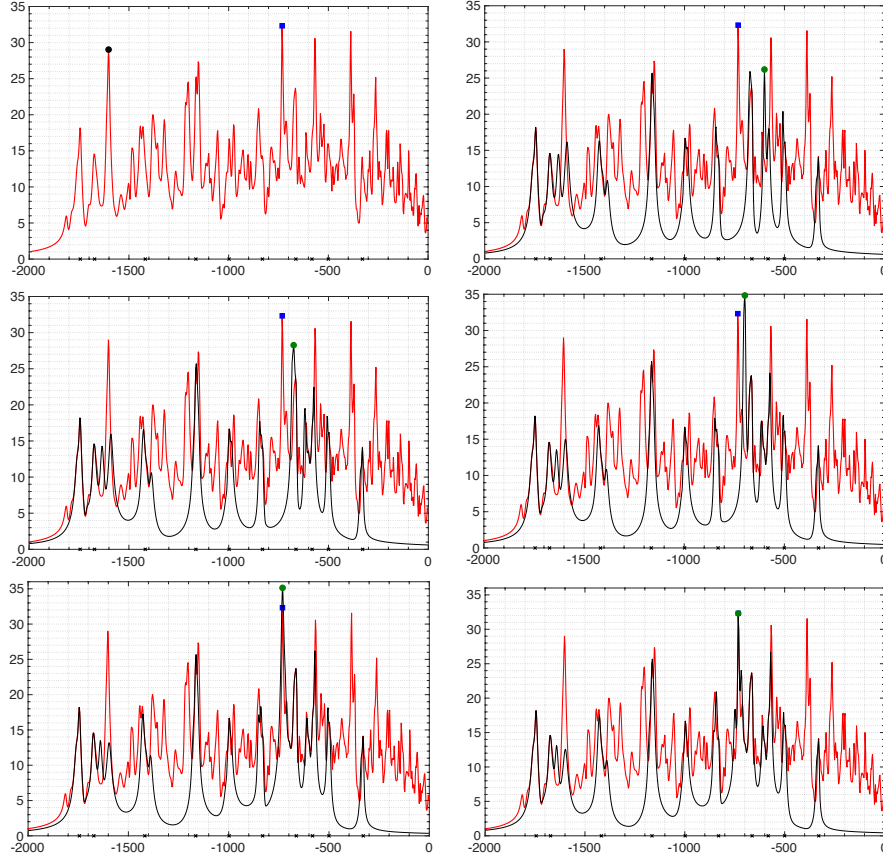


Figure 2: **(Top Left)** The plot of  $\sigma_{\max}(C(i\omega - (J - R)Q)^{-1}(J - R)B)$  as a function of  $\omega \in [-2000, 0]$  along with the maxima computed by Algorithm 2 and [2, Algorithm 1] marked with the square and circle, respectively, for a dense random example of order 800. **(Top Right)** The black curve is the initial reduced function for Algorithm 2 interpolating the full function at 10 points, whereas the circle is the global maximum of this reduced function. **(Middle Left - Bottom Right)** Plots of the reduced functions after iterations 1-4 of Algorithm 2 displayed with black curves along with the maximizers of the reduced functions marked with circles.

$k$	$\omega_{k+1}$	$\sigma_{\max}(G_k(\omega_{k+1}))$
10	-600.705819	26.182525
11	-674.769938	28.262865
12	-697.139310	34.834307
13	-731.573363	35.133647
14	-731.942586	32.309246
15	-731.977386	32.321399
16	-731.977385	32.321399

Table 1: Iterates of Algorithm 2 to compute  $r(Q; B, C)$  on a DH system with dense random  $J, R, Q \in \mathbb{R}^{800 \times 800}$  and random restriction matrices  $B \in \mathbb{R}^{800 \times 2}, C \in \mathbb{R}^{2 \times 800}$ . The algorithm is initiated with 10 interpolation points and terminates after 6 iterations with 32 dimensional subspaces.

the examples where the results by the subspace frameworks differ from those by the BB algorithm, the subspace frameworks converge to local maximizers that are not global maximizers.)

On these 100 random examples Algorithm 1 performs slightly fewer iterations, on average 17.6, whereas [2, Algorithm 1] on average performs 20.3 iterations. On the other hand, the total run-time on average is better for [2, Algorithm 1] compared with Algorithm 1 here, 21.3 s vs 30.9 s. We observe this behavior on various other DH systems; Algorithm 1 seems to be more robust for the computation of  $r(J; B, C) = r(R; B, C)$  in converging to the globally maximal value of  $\sigma_{\max}(CQ(i\omega I - (J - R)Q)^{-1}B)$  compared with [2, Algorithm 1], however, this is at the expense of slightly more computation time.

On the other hand, for the computation of  $r(Q; B, C)$ , Table 3 indicates that Algorithm 2 returns exactly the same globally maximal values (up to tolerances) as the BB algorithm for all of the first 10 examples except one, whereas application of [2, Algorithm 1] results in locally maximal solutions that are not globally maximal 4 times. Fewer number of subspace iterations in favor of Algorithm 2 are also apparent from the table. Once again, the plots of the ratios  $(f_{\text{BB}} - f_{\text{SF}})/((f_{\text{BB}} + f_{\text{SF}})/2)$  are shown in Figure 3 on the right-hand column for all 100 examples with  $f_{\text{SF}}$  now representing the value returned by Algorithm 2 on the top and by [2, Algorithm 1] at the bottom. Algorithm 2 and [2, Algorithm 1] return locally optimal solutions that are not globally optimal 21 and 27 times, respectively. In this case the difference between the number of subspace iterations for these 100 examples is more pronounced in favor of Algorithm 2; indeed the number of subspace iterations is on average 7.2 for Algorithm 2 and 17.0 for [2, Algorithm 1]. This difference in the number of iterations is also reflected in the average run-times which are 13.2 s and 19 s for Algorithm 2 and [2, Algorithm 1], respectively.

**Sparse Random Examples.** The  $5000 \times 5000$  sparse matrices  $J, Q, R$  are constrained to be banded with bandwidth 10. The matrix  $J$  is generated as in

Ex.	$\max_{\omega} \sigma_{\max}(CQ(i\omega I - (J - R)Q)^{-1}B)$			# iterations		run-time	
	Alg. 1	[2, Alg. 1]	BB Alg. [4]	Alg. 1	[2]	Alg. 1	[2]
1	32.559659	32.559659	32.559659	9	9	14	13
2	46.703932	46.703932	46.703932	15	12	16	12.4
3	26.227029	<b>24.023572</b>	26.227029	7	12	12.9	15.1
4	<b>62.748090</b>	108.030409	108.030409	17	41	17.8	27.2
5	35.974956	35.974957	35.974956	9	14	13.4	13.9
6	53.522033	53.522033	53.522033	6	3	11.4	10.2
7	31.739000	31.739000	31.739000	4	12	11.6	13.8
8	76.958658	76.958658	76.958658	35	8	43.2	11
9	37.007241	37.007241	37.007241	6	2	13	11.5
10	155.642871	155.642871	155.642871	2	2	8.6	8.5

Table 2: Run-time (in  $s$ ) comparison of Algorithm 1 and [2, Algorithm 1] to compute  $r(R; B, C) = r(J; B, C)$  for 10 dense random examples of order 800. The third column refers to the number of subspace iterations.

#	$\max \sigma_{\max}(C(i\omega I - (J - R)Q)^{-1}(J - R)B)$			# iterations		run-time	
	Alg. 1	[2, Alg. 1]	BB Alg. [4]	Alg. 1	[2]	Alg. 1	[2]
1	9.809182	9.809182	9.809182	3	26	11.1	19.5
2	22.386670	22.386670	22.386670	5	26	10.2	18.1
3	8.364927	8.364927	8.364927	3	8	11	13.3
4	32.321399	<b>29.028197</b>	32.321399	6	37	10.5	25.2
5	15.071678	15.071678	15.071678	7	15	12.7	14
6	21.641484	21.641484	21.641484	4	8	10.2	11.6
7	12.858494	<b>12.763161</b>	12.858494	6	4	11.8	11.2
8	31.901305	<b>27.996873</b>	31.901305	8	12	11.5	12.5
9	9.228945	9.228945	9.228945	3	8	11.3	13
10	<b>47.697528</b>	<b>47.697528</b>	71.534252	10	8	11.8	10.2

Table 3: Run-time (in  $s$ ) comparison of Algorithm 2 and [2, Algorithm 1] for the computation of  $r(Q; B, C)$  on 10 dense random examples of order 800.

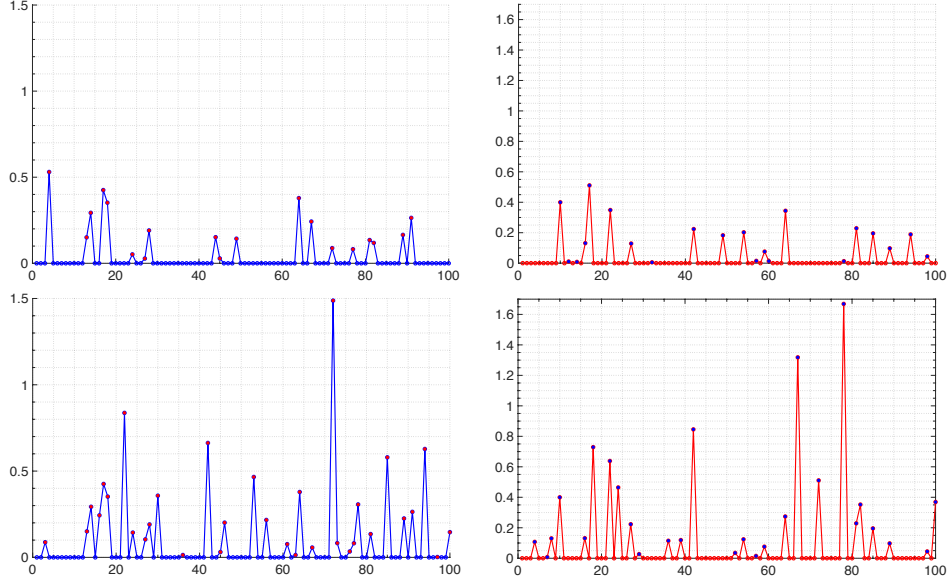


Figure 3: **(Left Column)** Ratios  $(f_{BB} - f_{SF})/((f_{BB} + f_{SF})/2)$  for 100 dense random DH examples of order 800, where  $f_{BB}$  and  $f_{SF}$  denote the maximal values of  $\sigma_{\max}(CQ(i\omega - (J - R)Q)^{-1}B)$  over  $\omega$  computed by the BB algorithm and the subspace framework (i.e., Algorithm 1 for the plot on the top, [2, Algorithm 1] for the plot at the bottom). **(Right Column)** Same as the left column on the same 100 dense random examples of order 800 except now this concerns a comparison of the maximization of  $\sigma_{\max}(C(i\omega - (J - R)Q)^{-1}(J - R)B)$  using Algorithm 2 in the top plot and [2, Algorithm 1] in the bottom plot.

the dense family randomly using the `randn` command, but the entries that fall outside of the bandwidth 10 are set equal to zero. The matrix  $Q > 0$  is created using the commands

```
>> A = sprandn(n,n,1/n); >> Q = (A + A')/2,
```

followed by setting the entries outside the bandwidth 10 again to zero. Finally, the following commands ensure that  $Q > 0$ .

```
>> mineig = eigs(Q,1,'smallestreal');
>> if (mineig<10^-4) Q=Q+(-mineig+5*rand)*speye(n); end
```

To form  $R \geq 0$ , first a diagonal matrix  $D$  of random rank not exceeding 500 is generated by the commands

```
>> p = round(500*rand); D = sparse(5000,5000); h = n/p;
>> for j=1:p k = floor(j*h); D(k,k) = 5*rand; end
```

Then we set  $R = \text{sparse}(X' * D * X)$  for a square random matrix  $X$  with bandwidth 5. The matrices  $B, C$  are random, and of size  $5000 \times 2, 2 \times 5000$ , respectively. The spectrum of a typical such sparse matrix  $(J - R)Q$  is displayed in Figure 1 on the right-hand side.

We again apply Algorithms 1 and 2 to 100 such random sparse examples. Since the matrices are too large to apply the BB algorithm, we compare the structure-preserving algorithms directly with the unstructured algorithm [2, Algorithm 1]. The retrieved estimates of  $\max_{\omega \in \mathbb{R} \cup \infty} \sigma_{\max}(G(i\omega))$  for  $G(i\omega) = CQ(i\omega I - (J - R)Q)^{-1}B$  and  $G(i\omega) = C(i\omega I - (J - R)Q)^{-1}(J - R)B$  are compared on the top and at the bottom, respectively, in Figure 4. Specifically, the ratio  $\frac{2(f_{\text{ST}} - f_{\text{UN}})}{f_{\text{ST}} + f_{\text{UN}}}$  is plotted for each random example with  $f_{\text{UN}}$  denoting the estimate by the unstructured algorithm [2, Algorithm 1], and  $f_{\text{ST}}$  denoting the estimate by the structured algorithm, i.e., Algorithm 1 for the top plot, Algorithm 2 for the bottom plot. According to the top plot, which concerns the computation of  $r(J; B, C) = R(R; B, C)$ , the two algorithms return exactly the same results (up to tolerances) for all but 6 examples; the structured algorithm returns better estimates for 4 of these 6 examples, while the unstructured algorithm returns better estimates for the other two. The structured algorithm appears to be even more robust for the computation of  $r(Q; B, C)$  in terms of avoiding locally optimal solutions away from global solutions; as displayed at the bottom, the structured algorithm returns a better estimate for 40 of the 100 examples, the unstructured algorithm returns the better estimate for 6 examples, and the results match exactly up to the tolerances for the remaining 54 examples.

The structured algorithms perform typically fewer iterations as compared to the unstructured algorithm. Indeed the average value of the number of subspace iterations performed on these 100 examples is 6.3 for the structured and 9.8 for the unstructured algorithm for the computation of  $r(J; B, C) = r(R; B, C)$ , while these average values are 9.7 and 12.9 for the computation of  $r(Q; B, C)$ . On the other hand, the unstructured algorithm is slightly superior when run-times are taken into account. The average run-times are 14.3 s for the structured and 12.7 s for the unstructured algorithm for the computation of  $r(J; B, C) = r(R; B, C)$ , whereas these figures are 16.5 s and 14 s for the computation of  $r(Q; B, C)$ .

We also list the computed maximal values of  $\sigma_{\max}(CQ(i\omega I - (J - R)Q)^{-1}B)$  and  $\sigma_{\max}(C(i\omega I - (J - R)Q)^{-1}(J - R)B)$  over  $\omega$  for the first 10 of these sparse examples in Tables 4 and 5. Included in these tables are also the number of subspace iterations, as well as the run-time required by the structured and the unstructured algorithm.

#### 4.3.3 The FE model of a Disk Brake

The only large-scale computation required by Algorithm 1 is the solution of the linear systems

$$D(i\omega) X = B \quad \text{and} \quad D(i\omega)^2 Y = B \quad (4.24)$$

at a given  $\omega \in \mathbb{R}$  in lines 2 and 6, where  $D(i\omega) = i\omega I - (J - R)Q$ . For the DH system resulting from a FE model of a disk brake in (1.6) and (1.7), the mass matrix  $M$  and the stiffness matrix  $K(\Omega)$  are available from the FE modeling. In other words, we have the sparse matrix  $Q^{-1}$ , but not  $Q$ , which

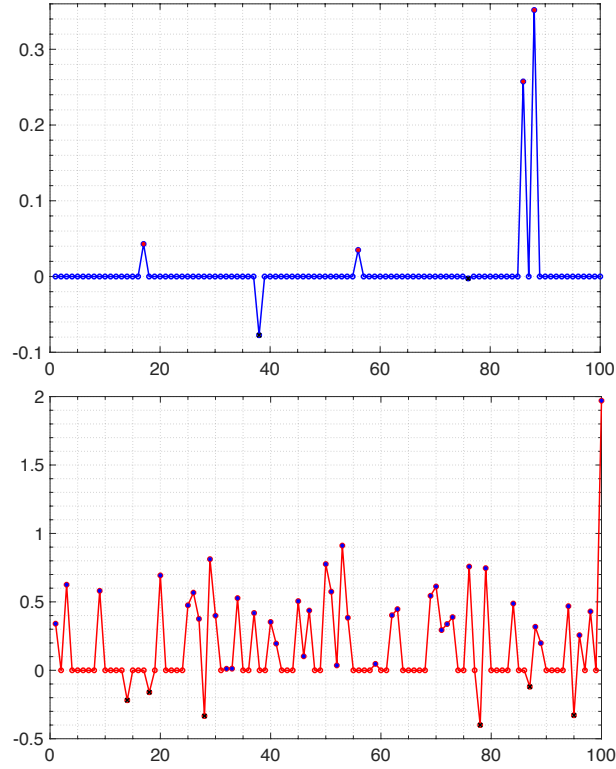


Figure 4: **(Top)** Plot of the ratios  $(f_{\text{ST}} - f_{\text{UN}})/((f_{\text{ST}} + f_{\text{UN}})/2)$  on 100 sparse random examples of order 5000, where  $f_{\text{ST}}, f_{\text{UN}}$  represent the computed maximal value of  $\sigma_{\max}(CQ(i\omega I - (J - R)Q)^{-1}B)$  over  $\omega$  by Algorithm 1 and [2, Algorithm 1], respectively. **(Bottom)** Similar to the top plot, only now for the computed maximal values  $f_{\text{ST}}, f_{\text{UN}}$  of  $\sigma_{\max}(C(i\omega I - (J - R)Q)^{-1}(J - R)B)$  over  $\omega$  by Algorithm 2 and [2, Algorithm 1], respectively.

#	$\max_{\omega} \sigma_{\max}(CQ(i\omega I - (J - R)Q)^{-1}B)$		# iterations		run-time	
	Alg. 1	[2, Alg. 1]	Alg. 1	[2]	Alg. 1	[2]
1	$1.393086 \times 10^3$	$1.393086 \times 10^3$	2	3	10.7	10.1
2	$8.323309 \times 10^2$	$8.323309 \times 10^2$	6	11	12.3	11.9
3	$1.289416 \times 10^3$	$1.289416 \times 10^3$	4	3	11.2	10.6
4	$8.850355 \times 10^2$	$8.850355 \times 10^2$	5	16	13.2	14
5	$6.891467 \times 10^2$	$6.891467 \times 10^2$	26	46	30.3	31.4
6	$5.652337 \times 10^6$	$5.652337 \times 10^6$	1	1	6.6	6.1
7	$8.834190 \times 10^2$	$8.834190 \times 10^2$	1	1	9.7	9.6
8	$3.402375 \times 10^3$	$3.402375 \times 10^3$	1	2	9.1	9
9	$8.240097 \times 10^2$	$8.240097 \times 10^2$	2	3	12.3	12.1
10	$1.256781 \times 10^3$	$1.256781 \times 10^3$	2	10	10.9	11.3

Table 4: Comparison of Algorithms 1 and [2, Algorithm 1] to compute  $r(R; B, C) = r(J; B, C)$  on sparse random DH systems of order 5000 of bandwidth 10. The MATLAB commands to generate these random sparse examples are explained in Section 4.3.2. The third column lists the number of subspace iterations, while the run-times (in  $s$ ) are listed in the last column.

#	$\max_{\omega} \sigma_{\max}(C(i\omega I - (J - R)Q)^{-1}(J - R)B)$		# iterations		run-time	
	Alg. 2	[2, Alg. 1]	Alg. 2	[2]	Alg. 2	[2]
1	$1.320177 \times 10^3$	<b><math>9.360406 \times 10^2</math></b>	15	39	18.3	26.7
2	$8.284036 \times 10^2$	$8.284036 \times 10^2$	2	13	11.1	12.7
3	$9.288427 \times 10^2$	<b><math>4.863413 \times 10^2</math></b>	31	11	46.1	11.8
4	$6.583171 \times 10^2$	$6.583171 \times 10^2$	5	17	13.1	14.6
5	$7.722736 \times 10^2$	$7.722736 \times 10^2$	3	9	11.3	11.4
6	$2.260647 \times 10^6$	$2.260647 \times 10^6$	1	1	6.4	6.1
7	$1.660164 \times 10^3$	$1.660164 \times 10^3$	4	12	10.6	11.1
8	$1.515824 \times 10^3$	$1.515824 \times 10^3$	8	9	11.8	10
9	$4.610935 \times 10^2$	<b><math>2.535771 \times 10^2</math></b>	13	4	17.9	12.1
10	$1.011299 \times 10^3$	$1.011299 \times 10^3$	3	7	10.8	10.6

Table 5: Comparison of Algorithms 2 and [2, Algorithm 1] to compute  $r(Q; B, C)$ . The display is analogous to Table 4, in particular the numerical experiments are carried out exactly on the same 10 sparse random examples of order 5000 employed for Table 4.

turns out to be dense. Trying to invert  $Q^{-1}$  and/or solve a linear system with the coefficient matrix  $Q$  is computationally very expensive and would require full matrix storage.

This difficulty can be avoided by exploiting that

$$(\mathrm{i}\omega I - (J - R)Q)^{-1}B = Q^{-1}(\mathrm{i}\omega Q^{-1} - (J - R))^{-1}B.$$

Hence, to compute  $X, Y$  as in (4.24), we proceed as follows.

- (1) We first solve  $(\mathrm{i}\omega Q^{-1} - (J - R))\hat{X} = B$  for  $\hat{X}$ , and set  $X = Q^{-1}\hat{X}$ .
- (2) Then we solve  $(\mathrm{i}\omega Q^{-1} - (J - R))\hat{Y} = X$  for  $\hat{Y}$ , and set  $Y = Q^{-1}\hat{Y}$ .

A second observation that further speeds up the computation is the particular structure of the coefficient matrix  $\{\mathrm{i}\omega Q^{-1} - (J - R)\}$  with  $N = 0$ . With  $\widetilde{M}(\mathrm{i}\omega; \Omega) := \mathrm{i}\omega M + D(\Omega) + G(\Omega)$ , we have

$$\mathrm{i}\omega Q^{-1} - (J - R) = \begin{bmatrix} \widetilde{M}(\mathrm{i}\omega; \Omega) & K(\Omega) \\ -K(\Omega) & \mathrm{i}\omega K(\Omega) \end{bmatrix}.$$

Hence, to solve  $\{\mathrm{i}\omega Q^{-1} - (J - R)\}Z = W$ , for a given  $W = \begin{bmatrix} W_1^T & W_2^T \end{bmatrix}^T$  and unknown  $Z = \begin{bmatrix} Z_1^T & Z_2^T \end{bmatrix}^T$  with  $W_1, W_2, Z_1, Z_2$  having all equal number of rows, we perform a column block permutation and then eliminate the lower left block to obtain

$$\begin{bmatrix} K(\Omega) & \widetilde{M}(\mathrm{i}\omega; \Omega) \\ 0 & -K(\Omega) - \mathrm{i}\omega \widetilde{M}(\mathrm{i}\omega; \Omega) \end{bmatrix} \begin{bmatrix} Z_2 \\ Z_1 \end{bmatrix} = \begin{bmatrix} W_1 \\ W_2 - \mathrm{i}\omega W_1 \end{bmatrix},$$

which in turn yields

$$(-K(\Omega) - \mathrm{i}\omega \widetilde{M}(\mathrm{i}\omega; \Omega))Z_1 = W_2 - \mathrm{i}\omega W_1, \quad K(\Omega)Z_2 = W_1 - \widetilde{M}(\mathrm{i}\omega; \Omega)Z_1.$$

At every subspace iteration, the highest costs arise from the computation of the  $LU$  factorizations of the sparse matrices  $K(\Omega)$  and  $K(\Omega) + \mathrm{i}\omega \widetilde{M}(\mathrm{i}\omega; \Omega)$ .

The main cost for Algorithm 2 is the solution of the linear systems

$$D(\mathrm{i}\omega)^H X = C^H \quad \text{and} \quad [D(\mathrm{i}\omega)^H]^2 Y = C^H$$

at a given  $\omega \in \mathbb{R}$ . This can be treated similarly by exploiting that

$$D(\mathrm{i}\omega)^{-H} C^H = (\mathrm{i}\omega Q^{-1} - (J - R))^{-H} (CQ^{-1})^H.$$

We have applied Algorithm 1 to compute the unstructured stability radius  $r(R; B, B^T)$  for the DH system of the form (1.6), (1.7) resulting from the FE brake model with  $N = 0$ , where  $G(\Omega), K(\Omega), D(\Omega), M \in \mathbb{R}^{4669 \times 4669}$  so that  $J, R, Q \in \mathbb{R}^{9338 \times 9338}$ .

The plot of the computed  $r(R; B, B^T)$  vs the rotation speed  $\Omega$  is presented in Figure 5 at lower frequencies (i.e.,  $\Omega \in [2.5, 100]$ ) on the top, and at higher frequencies (i.e.,  $\Omega \in [900, 1700]$ ) at the bottom. For smaller frequencies, the

$\Omega$	$r(R; B, B^T)$	$\omega_*$	iterations	run-time	dimension
2.5	0.01066	$-1.938 \times 10^5$	2	22.0	72
5	0.01038	$-1.938 \times 10^5$	2	21.7	72
10	0.01026	$-1.938 \times 10^5$	3	24.9	78
50	0.00999	$-1.938 \times 10^5$	2	22.0	72
100	0.00988	$-1.938 \times 10^5$	2	22.1	72
1000	0.00809	$-1.789 \times 10^5$	2	21.4	72
1050	0.00789	$-1.789 \times 10^5$	2	21.8	72
1100	0.00834	$-1.789 \times 10^5$	3	24.8	78
1116	0.01091	$-1.789 \times 10^5$	3	25.6	78
1150	0.00344	$-1.742 \times 10^5$	2	21.5	72
1200	0.00407	$-1.742 \times 10^5$	2	21.8	72
1250	0.00471	$-1.742 \times 10^5$	2	21.8	72
1300	0.00516	$-1.742 \times 10^5$	2	21.2	72

Table 6: Computed stability radii  $r(R; B, B^T)$  by Algorithm 1 for several  $\Omega$  values for the DH system of order 9338 originating from the FE brake model. The other columns display  $\omega_*$  corresponding to  $\arg \max_{\omega} \sigma_{\max}(B^T Q(i\omega I - (J - R)Q)^{-1} B)$ , the number of subspace iterations, the total run-time (in  $s$ ) and the subspace dimension at termination.

stability radius initially decreases with respect to  $\Omega$ , but around  $\Omega = 1100$  the stability radius suddenly increases. The non-smooth nature of the stability radius with respect to  $\Omega$  is apparent from the figure. One should note, in particular, the sharp turns near  $\Omega = 1120$  and  $\Omega = 1590$ ; this non-smoothness is due to the fact that  $\sigma_{\max}(B^T Q(i\omega I - (J - R)Q)^{-1} B)$  has multiple global maximizers. This means that two distinct points on the imaginary axis can be attained with perturbations of minimal norm.

The computed values of  $r(R; B, B^T)$  are listed in Table 6 for some values of  $\Omega$ . In this table, for each  $\Omega$ , the value  $\omega_*$ , where the singular value function  $\sigma_{\max}(B^T Q(i\omega I - (J - R)Q)^{-1} B)$  is maximized globally is displayed, the number of subspace iterations, the run-time (in  $s$ ) and the subspace dimension at termination are included as well. In all cases, 2 or 3 subspace iterations are sufficient to achieve the prescribed accuracy tolerance. This leads to considerably smaller reduced systems of size  $72 \times 72$  or  $78 \times 78$  compared with the original problem of size  $9338 \times 9338$ . The  $\omega$  value maximizing  $\sigma_{\max}(B^T Q(i\omega I - (J - R)Q)^{-1} B)$  differs substantially, depending on whether the frequency  $\Omega$  is small or large.

The resulting reduced problems at termination capture the full problem remarkably well around the global maximizer. This is depicted in Figure 6, where, for  $\Omega = 1000$ , the singular value function  $f(\omega) = \sigma_{\max}(B^T Q(i\omega I - (J - R)Q)^{-1} B)$  for the full problem (solid curve) and  $f_k(\omega) = \sigma_{\max}(B_k^T Q_k(i\omega I - (J_k - R_k)Q_k)^{-1} B_k)$  for the reduced problem at termination (dashed curve) are plotted near the global maximizer  $\omega_* = -178880.9$ . It is fairly difficult to distinguish these two curves from each other.

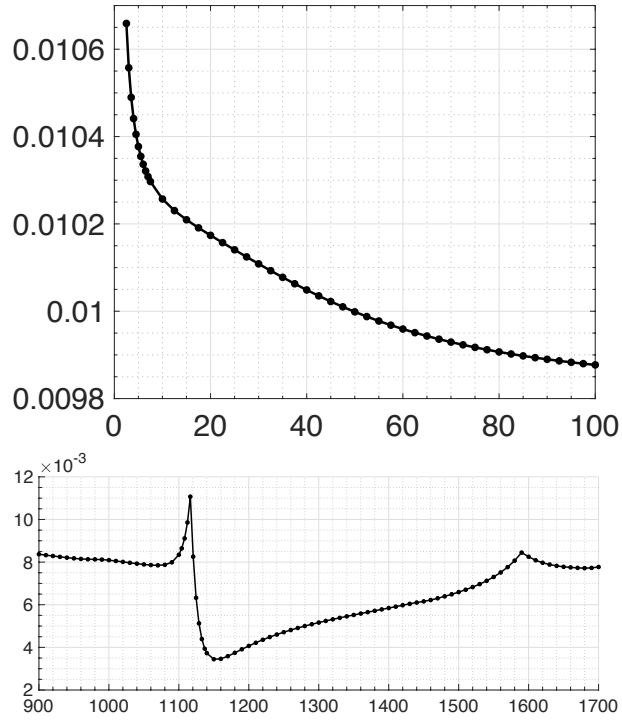


Figure 5: Plot of the stability radius  $r(R; B, B^T)$  for the DH system (1.6), (1.7) resulting from a the FE model of a disk-brake as a function of the rotation speed  $\Omega$  for  $\Omega \in [2.5, 100]$  (top plot), and  $\Omega \in [900, 1700]$  (bottom plot). The order of the DH system under consideration in this plot is 9338.

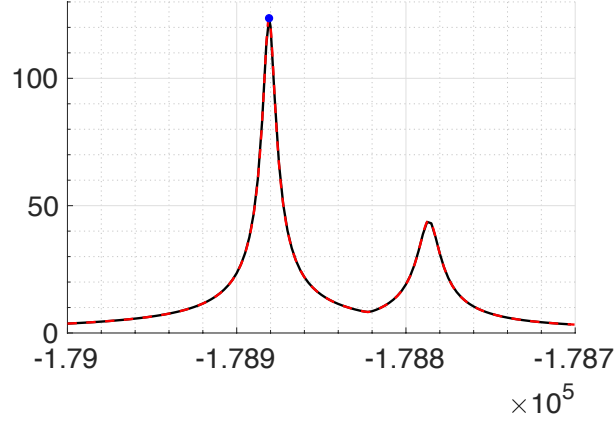


Figure 6: Plot of the singular value functions  $f(\omega) = \sigma_{\max}(B^T Q(i\omega I - (J - R)Q)^{-1} B)$  (solid curve) and  $f_k(\omega) = \sigma_{\max}(B_k^T Q_k(i\omega I - (J_k - R_k)Q_k)^{-1} B_k)$  (dashed curve) at termination of Algorithm 1 near the global maximizer  $\omega_* = -17880.9$  for the DH system of order 9338 arising from the FE disk-brake model with  $\Omega = 1000$ . The circle marks  $(\omega_*, f(\omega_*))$ .

## 5 Computation of the Structured Stability Radius

In the last section we have studied stability radii for dissipative Hamiltonian systems where the restriction matrices, however, allowed unstructured perturbations in the system coefficients. In this section we put additional constraints on the perturbations, in particular we require that the perturbations are structured themselves. We discuss only perturbations in the dissipation matrix, since this is usually the most uncertain part of the system, due to the fact that modeling damping or friction very exactly is usually extremely difficult. We deal with the computation of  $r^{\text{Herm}}(R; B)$  defined as in (2.4). We first describe a numerical technique for small-scale problems in Section 5.1 and then develop a subspace framework that converges superlinearly with respect to the subspace dimension in Section 5.2. Both techniques use the eigenvalue optimization characterization of  $r^{\text{Herm}}(R; B)$  in Theorem 3.2.

### 5.1 Small-Scale Problems

#### 5.1.1 Inner Maximization Problems

The eigenvalue optimization characterization of  $r^{\text{Herm}}(R; B)$  is a min-max problem, where the inner maximization problem is convex, indeed it can alternatively be expressed as a semi-definite program (SDP). Formally, for a given  $\omega \in \mathbb{R}$ , and  $H_0(i\omega), H_1(i\omega)$  representing the Hermitian matrices defined in Theorem (3.2),

we have

$$\begin{aligned}\tilde{\eta}^{\text{Herm}}(R; B, i\omega) &= \sup_{t \in \mathbb{R}} \lambda_{\min}(H_0(i\omega) + tH_1(i\omega)) \\ &= \sup\{z \mid z, t \in \mathbb{R} \text{ s.t. } H_0(i\omega) + tH_1(i\omega) - zI \geq 0\},\end{aligned}\tag{5.1}$$

where the characterization in the second line is a linear convex SDP. Here  $\tilde{\eta}^{\text{Herm}}(R; B, i\omega)$  is related to a structured backward error for the eigenvalue  $i\omega$ , specifically it corresponds to the square of the distance (see [15, Definition 3.2 and Theorem 4.9])

$$\eta^{\text{Herm}}(R; B, i\omega) := \inf\{\|\Delta\|_2 \mid \Delta = \Delta^H, \ i\omega \in \Lambda((J - R)Q - (B\Delta B^H)Q)\}.\tag{5.2}$$

We have that  $\eta^{\text{Herm}}(R; B, i\omega)$  is finite if and only if the suprema in (5.1) are attained, which happens if and only if  $H_1(i\omega)$  is indefinite, i.e.,  $H_1(i\omega)$  has both negative and positive eigenvalues.

The most widely used techniques to solve a linear convex SDP are different forms of interior-point methods. Implementations of some of these interior-point methods are made available through the package `cvx` [8, 9]. Hence, one option is to use `cvx` to compute  $\tilde{\eta}^{\text{Herm}}(R; B, i\omega)$  directly. An alternative, and also theoretically well understood approach, is to employ the software package `eigopt` [16] for the eigenvalue optimization problem in the first characterization in (5.1). This second approach forms piece-wise quadratic functions that lie globally above the eigenvalue function, and maximizes these piece-wise quadratic functions instead of the eigenvalue function. Each piece-wise quadratic function is defined as the minimum of several other quadratic functions, all of which have the same curvature  $\gamma$  (which must be a global upper bound on the second derivative of the eigenvalue function at all points where the eigenvalue function is differentiable). Any slightly positive real number for the curvature  $\gamma$  serves the purpose (e.g.,  $\gamma = 10^{-6}$ ), since the smallest eigenvalue function in (5.1) is a concave function of  $t$ .

In our experience, `eigopt` performs the computation of  $\tilde{\eta}^{\text{Herm}}(R; B, i\omega)$  significantly faster than `cvx`. The only downside is that an interval containing the optimal  $t$  for (5.1) must be supplied to `eigopt`, whereas such an interval is not needed by `cvx`.

### 5.1.2 Outer Minimization Problems

The minimum of  $\tilde{\eta}^{\text{Herm}}(R; B, i\omega)$  with respect to  $\omega \in \mathbb{R}$  yields the distance  $r^{\text{Herm}}(R; B)$ , and the minimizing  $\omega \in \mathbb{R}$  yields the point  $i\omega$  that first becomes an eigenvalue on the imaginary axis under the smallest perturbation possible. This is a non-convex optimization problem, indeed the objective  $\tilde{\eta}^{\text{Herm}}(R; B, i\omega)$  may even blow up at some  $\omega$ .

We again resort to `eigopt` for the minimization of  $\tilde{\eta}^{\text{Herm}}(R; B, i\omega)$ . For the sake of completeness, a formal description is provided in Algorithm 3 below, where we use the abbreviations

$$\tilde{\eta}_j := \tilde{\eta}^{\text{Herm}}(R; B, i\omega_j) \quad \text{and} \quad \tilde{\eta}_j' := \frac{d\tilde{\eta}^{\text{Herm}}(R; B, i\omega_j)}{d\omega}.$$

Introducing  $t(\omega) := \arg \max_{t \in \mathbb{R}} \lambda_{\min}(H_0(i\omega) + tH_1(i\omega))$ , the algorithm approximates the smallest eigenvalue function

$$\lambda_{\min}(H_0(i\omega) + t(\omega)H_1(i\omega)) = \tilde{\eta}^{\text{Herm}}(R; B, i\omega)$$

with the piece-wise quadratic function

$$\begin{aligned} Q_k(\omega) &:= \max\{q_j(\omega) \mid j = 0, \dots, k\}, \\ q_j(\omega) &:= \tilde{\eta}_j + \tilde{\eta}'_j(\omega - \omega_j) + (\gamma/2)(\omega - \omega_j)^2 \end{aligned}$$

at iteration  $k$ . It computes the global minimizer  $\omega_{k+1}$  of  $Q_k(\omega)$ , and refines the piece-wise quadratic function  $Q_k(\omega)$  with the addition of one more quadratic piece, namely  $q_{k+1}(\omega) := \tilde{\eta}_{k+1} + \tilde{\eta}'_{k+1}(\omega - \omega_{k+1}) + (\gamma/2)(\omega - \omega_{k+1})^2$ . Here,  $\gamma$  is supposed to be a lower bound for the second derivative  $\lambda''_{\min}(H_0(i\omega) + t(\omega)H_1(i\omega))$  for all  $\omega$  sufficiently close to the global minimizer of  $\tilde{\eta}^{\text{Herm}}(R; B, i\omega)$ . In theory, it can be shown that for all  $\gamma$  small enough, every convergent subsequence of the sequence  $\{\omega_k\}$  converges to a global minimizer of  $\tilde{\eta}^{\text{Herm}}(R; B, i\omega)$ .

At step  $k$  of the algorithm  $\tilde{\eta}^{\text{Herm}}(R; B, i\omega)$  and its derivative need to be computed at  $\omega_{k+1}$ . We rely on one of the two approaches (`cvx` or `eigopt`) described in Section 5.1.1 for the computation of  $\tilde{\eta}^{\text{Herm}}(R; B, i\omega_{k+1})$ , and employ a finite difference formula to approximate its derivative.

---

**Algorithm 3** Small-Scale Computation of  $r^{\text{Herm}}(R; B)$ .

---

**Input:** Matrices  $B \in \mathbb{C}^{n \times m}$ ,  $J, R, Q \in \mathbb{C}^{n \times n}$ , a negative real number  $\gamma$ , and a closed interval  $\tilde{\Omega} \subseteq \mathbb{R}$  that contains the global minimizer of  $\tilde{\eta}^{\text{Herm}}(R; B, i\omega)$  over  $\omega \in \mathbb{R}$ .

**Output:** The sequence  $\{\omega_k\}$ .

- 1:  $\omega_0 \leftarrow$  an initial point in  $\tilde{\Omega}$
- 2: Compute  $\tilde{\eta}_0 := \tilde{\eta}^{\text{Herm}}(R; B, i\omega_0)$  and  $\tilde{\eta}'_0 := d\tilde{\eta}^{\text{Herm}}(R; B, i\omega_0)/d\omega$
- 3: **for**  $k = 0, 1, \dots$  **do**
- 4:    $Q_k(\omega) \leftarrow \max\{q_j(\omega) \mid j = 0, \dots, k\}$ , where
 
$$q_j(\omega) := \tilde{\eta}_j + \tilde{\eta}'_j(\omega - \omega_j) + (\gamma/2)(\omega - \omega_j)^2$$
- 5:    $\omega_{k+1} \leftarrow \arg \min_{\omega \in \tilde{\Omega}} Q_k(\omega)$
- 6:   Compute

$$\begin{aligned} \tilde{\eta}_{k+1} &:= \tilde{\eta}^{\text{Herm}}(R; B, i\omega_{k+1}), \\ \tilde{\eta}'_{k+1} &:= d\tilde{\eta}^{\text{Herm}}(R; B, i\omega_{k+1})/d\omega \end{aligned}$$

7: **end for**

---

## 5.2 Large-Scale Problems

The characterization via eigenvalue optimization in Theorem 3.2 is in terms of the matrix-valued functions  $H_0(i\omega), H_1(i\omega)$ , which are of small size provided that  $B$  has few columns. The large-scale nature of this characterization is hidden in the matrix-valued function  $W(i\omega) := (J - R)Q - i\omega I$  defined in Theorem 3.2. Note that, in particular, both  $H_0(i\omega)$  and  $H_1(i\omega)$  are defined in terms

of  $W(i\omega)^{-1}B$ . This is also reflected in Algorithm 3 when  $J, R, Q$  are large; at iteration  $k$  of the algorithm, the matrices  $H_0(i\omega_{k+1})$ ,  $H_1(i\omega_{k+1})$  need to be formed for the computation of  $\hat{\eta}^{\text{Herm}}(R; B, i\omega_{k+1})$ , which in turn requires the solution of the linear system  $W(i\omega_{k+1})Z = B$ .

To cope with the large-scale setting, we benefit from structure preserving two-sided projections similar to those described in Section 4.2. In particular, for a given subspace  $\mathcal{V}_k$  and a matrix  $V_k$  whose columns form an orthonormal basis for  $\mathcal{V}_k$ , we set

$$W_k := QV_k(V_k^H QV_k)^{-1}. \quad (5.3)$$

Furthermore, we define the projected matrices

$$\begin{aligned} J_k &:= W_k^H J W_k, & R_k &:= W_k^H R W_k, \\ Q_k &:= V_k^H Q V_k, & B_k &:= W_k^H B. \end{aligned} \quad (5.4)$$

Recall also the identities

$$W_k^H V_k = I \quad \text{and} \quad (W_k V_k^H)^2 = W_k V_k^H, \quad (5.5)$$

the latter of which means that  $W_k V_k^H$  is an oblique projector onto  $\text{Im}(QV_k)$ .

Although these identities are still available, however, we no longer have a tool such as Theorem 4.1 that we could depend on to establish interpolation results. This is because there is no apparent transfer function, as there is indeed no apparent linear port-Hamiltonian system that can be tied to the eigenvalue optimization characterization. But the following simple observation turns out to be very useful.

**Lemma 5.1.** *Consider a DH model (1.1) and a reduced model  $\dot{x}_k = (J_k - R_k)Q_k x_k$  with coefficients as in (5.4). With  $W(\lambda) = (J - R)Q - \lambda I$  and  $W_k(\lambda) := (J_k - R_k)Q_k - \lambda I$ , we then have*

$$W_k(\lambda) = W_k^H W(\lambda) V_k \quad \text{for all } \lambda \in \mathbb{C}.$$

*Proof.* From the definition of  $J_k, R_k, Q_k$  in (5.4) we obtain

$$\begin{aligned} W_k(\lambda) &= (W_k^H (J - R) W_k) V_k^H Q V_k - \lambda I \\ &= W_k^H (J - R) Q V_k - \lambda W_k^H V_k \\ &= W_k^H \{(J - R)Q - \lambda I\} V_k = W_k^H W(\lambda) V_k, \end{aligned}$$

where we have employed the identities in (5.5).  $\square$

Our reduced problems are expressed in terms of the reduced versions of  $H_0(\lambda)$ ,  $L(\lambda)$ , and  $H_1(\lambda)$  defined via

$$H_{k,0}(\lambda) := L_k(\lambda)^{-1} L_k(\lambda)^{-H},$$

with  $L_k(\lambda)$  denoting a lower triangular Cholesky factor of

$$\tilde{H}_{k,0}(\lambda) := B_k^H W_k(\lambda)^{-H} Q_k B_k B_k^H Q_k W_k(\lambda)^{-1} B_k,$$

and  $H_{k,1}(\lambda) := i(\tilde{H}_{k,1}(\lambda) - \tilde{H}_{k,1}(\lambda)^H)$  with

$$\tilde{H}_{k,1}(\lambda) := L_k(\lambda)^{-1} B_k^H W_k(\lambda)^{-H} Q_k B_k L_k(\lambda)^{-H}.$$

Note that to ensure the uniqueness of  $L(\lambda)$  and  $L_k(\lambda)$ , we define them as the Cholesky factors of  $\tilde{H}_0(\lambda)$  and  $\tilde{H}_{k,0}(\lambda)$  with real and positive entries along the diagonal. Our goal is to come up with reduced counterparts of  $\tilde{\eta}^{\text{Herm}}(R; B, i\omega)$ ,  $\eta^{\text{Herm}}(R; B, i\omega)$  that Hermite-interpolate the full functions at prescribed points. For a given subspace  $\mathcal{V}_k$ , a matrix  $V_k$  whose columns form an orthonormal basis for  $\mathcal{V}_k$ , and for  $W_k$  as in (5.3), we introduce

$$\begin{aligned} \tilde{\eta}_k^{\text{Herm}}(R; B, i\omega) &:= \sup_{t \in \mathbb{R}} \lambda_{\min}(H_{k,0}(i\omega) + tH_{k,1}(i\omega)) \\ \eta_k^{\text{Herm}}(R; B, i\omega) &:= \inf\{\|\Delta\|_2 \mid \Delta = \Delta^H, \ i\omega \in \Lambda((J_k - R_k)Q_k - (B_k \Delta B_k^H)Q_k)\}. \end{aligned}$$

Recall that  $\eta^{\text{Herm}}(R; B, i\omega) = \tilde{\eta}^{\text{Herm}}(R; B, i\omega)^{1/2}$ , and a similar relation holds for the reduced problems, i.e.,  $\eta_k^{\text{Herm}}(R; B, i\omega) = \tilde{\eta}_k^{\text{Herm}}(R; B, i\omega)^{1/2}$ .

We start our analysis by establishing that the quantities  $\eta_k^{\text{Herm}}(R; B, i\omega)$ ,  $\tilde{\eta}_k^{\text{Herm}}(R; B, i\omega)$  are independent of the choice of basis for the subspace  $\mathcal{V}_k$ . For this proof we introduce the notation  $\eta_{V_k, W_k}^{\text{Herm}}(R; B, i\omega)$ ,  $\tilde{\eta}_{V_k, W_k}^{\text{Herm}}(R; B, i\omega)$  to emphasize the particular choices of basis  $V_k, W_k$  for the subspaces  $\mathcal{V}_k, \mathcal{W}_k$  used in the definitions of  $\eta_k^{\text{Herm}}(R; B, i\omega)$ ,  $\tilde{\eta}_k^{\text{Herm}}(R; B, i\omega)$ . Similarly, we indicate the spaces and the bases with the help of the notations  $J_{W_k}$ ,  $R_{W_k}$ ,  $Q_{V_k}$ , and  $B_{W_k}$ .

**Lemma 5.2.** *Let the columns of  $V_k$  and  $\tilde{V}_k$  form orthonormal bases for the subspace  $\mathcal{V}_k$ , and let  $W_k := QV_k(V_k^H QV_k)^{-1}$ ,  $\tilde{W}_k := Q\tilde{V}_k(\tilde{V}_k^H Q\tilde{V}_k)^{-1}$ . Then*

$$\begin{aligned} \eta_{V_k, W_k}^{\text{Herm}}(R; B, i\omega) &= \eta_{\tilde{V}_k, \tilde{W}_k}^{\text{Herm}}(R; B, i\omega), \\ \tilde{\eta}_{V_k, W_k}^{\text{Herm}}(R; B, i\omega) &= \tilde{\eta}_{\tilde{V}_k, \tilde{W}_k}^{\text{Herm}}(R; B, i\omega) \end{aligned}$$

for all  $\omega \in \mathbb{R}$ .

*Proof.* It suffices to prove  $\eta_{V_k, W_k}^{\text{Herm}}(R; B, i\omega) = \eta_{\tilde{V}_k, \tilde{W}_k}^{\text{Herm}}(R; B, i\omega)$ , the other equality follows from this equality immediately. Since the columns of  $V_k$  and  $\tilde{V}_k$  form bases for the same space, there exists a unitary matrix  $P$  such that  $\tilde{V}_k = V_k P$ . Furthermore, by definition,

$$\tilde{W}_k = Q(V_k P)(P^H V_k^H QV_k P)^{-1} = QV_k P P^H (V_k^H QV_k)^{-1} P = W_k P.$$

The assertion then follows from the following set of equivalences:

$$\begin{aligned} i\omega \in \Lambda((J_{W_k} - R_{W_k})Q_{V_k} - (B_{W_k} \Delta B_{W_k}^H)Q_{V_k}) &\iff \\ \det(W_k^H (J - R)QV_k - W_k^H B \Delta B^H QV_k - i\omega W_k^H V_k) = 0 &\iff \\ \det(W_k^H ((J - R)Q - B \Delta B^H Q - i\omega I) V_k) = 0 &\iff \\ \det(P^H W_k^H ((J - R)Q - B \Delta B^H Q - i\omega I) V_k P) = 0 &\iff \\ \det(\tilde{W}_k^H (J - R)Q\tilde{V}_k - \tilde{W}_k^H B \Delta B^H Q\tilde{V}_k - i\omega \tilde{W}_k^H \tilde{V}_k) = 0 &\iff \\ i\omega \in \Lambda((J_{\tilde{W}_k} - R_{\tilde{W}_k})Q_{\tilde{V}_k} - (B_{\tilde{W}_k} \Delta B_{\tilde{W}_k}^H)Q_{\tilde{V}_k}). & \end{aligned}$$

□

In the following we will develop a subspace framework including an Hermite interpolation property for DH systems. For this we first show an auxiliary interpolation result for  $B^H QW(\lambda)^{-1}B$ , the matrix through which  $H_0(\lambda), H_1(\lambda)$  are defined.

**Theorem 5.3.** *Consider a DH model (1.1) and a reduced model  $\dot{x}_k = (J_k - R_k)Q_k x_k$  with coefficients as in (5.4), and let  $W(\lambda) = (J - R)Q - \lambda I$  and  $W_k(\lambda) := (J_k - R_k)Q_k - \lambda I$ . For a given  $\hat{\lambda} \in \mathbb{C}$  such that  $W(\hat{\lambda})$  and  $W_k(\hat{\lambda})$  are invertible, the following assertions hold:*

- (i) *If  $\text{Im}(W(\hat{\lambda})^{-1}B) \subseteq \mathcal{V}_k$ , then  $B^H QW(\hat{\lambda})^{-1}B = B_k^H Q_k W_k(\hat{\lambda})^{-1}B_k$ .*
- (ii) *Additionally, if  $\text{Im}(W(\hat{\lambda})^{-2}B) \subseteq \mathcal{V}_k$  and the orthonormal basis  $V_k$  for  $\mathcal{V}_k$  is such that  $V_k = \begin{bmatrix} \tilde{V}_k & \hat{V}_k \end{bmatrix}$  where the columns of  $\tilde{V}_k$  form an orthonormal basis for  $\text{Im}(W(\hat{\lambda})^{-1}B)$ , then  $B^H QW(\hat{\lambda})^{-2}B = B_k^H Q_k W_k(\hat{\lambda})^{-2}B_k$ .*

*Proof.* (i) If  $\text{Im}(W(\hat{\lambda})^{-1}B) \subseteq \mathcal{V}_k$  then, since  $W_k V_k^H$  is a projector onto  $\text{Im}(QV_k)$ , we obtain

$$\begin{aligned} B^H QW(\hat{\lambda})^{-1}B &= B^H QV_k V_k^H W(\hat{\lambda})^{-1}B = B^H W_k V_k^H QV_k V_k^H W(\hat{\lambda})^{-1}B \\ &= B_k^H Q_k V_k^H W(\hat{\lambda})^{-1}B. \end{aligned}$$

To show that  $V_k^H W(\hat{\lambda})^{-1}B = W_k(\hat{\lambda})^{-1}B_k$ , let  $Z := W(\hat{\lambda})^{-1}B$ , and  $Z_k$  be such that  $V_k Z_k = Z$ . (There exists a unique  $Z_k$  with this property, because  $\text{Im}(Z) \subseteq \mathcal{V}_k$ .) Then  $W(\hat{\lambda})Z = B$  implies that  $W(\hat{\lambda})V_k Z_k = B$ , and thus  $W_k^H W(\hat{\lambda})V_k Z_k = W_k^H B$ . Hence, by Lemma 5.1 we see that  $Z_k = W_k(\hat{\lambda})^{-1}B_k$ , implying that

$$V_k^H W(\hat{\lambda})^{-1}B = V_k^H Z = V_k^H (V_k Z_k) = W_k(\hat{\lambda})^{-1}B_k,$$

as asserted.

(ii) Following the steps at the beginning of the proof of part (i), we have

$$B^H QW(\hat{\lambda})^{-2}B = B_k^H Q_k V_k^H W(\hat{\lambda})^{-2}B.$$

To show that  $V_k^H W(\hat{\lambda})^{-2}B = W_k(\hat{\lambda})^{-2}B_k$ , we exploit that

$$V_k^H W(\hat{\lambda})^{-2}B = (V_k^H W(\hat{\lambda})^{-1}\tilde{V}_k)(\tilde{V}_k^H W(\hat{\lambda})^{-1}B). \quad (5.6)$$

Now define  $Z := W(\hat{\lambda})^{-1}\tilde{V}_k$  and  $Z_k$  such that  $V_k Z_k = Z$  (once again such a  $Z_k$  exists uniquely, because  $\text{Im}(Z) \subseteq \mathcal{V}_k$ ) so that  $W(\hat{\lambda})Z = \tilde{V}_k$ . Then  $W(\hat{\lambda})V_k Z_k = \tilde{V}_k$  and hence  $W_k^H W(\hat{\lambda})V_k Z_k = I_{k,m}$ , where  $I_{k,m}$  is the matrix consisting of the first  $m$  columns of the  $k \times k$  identity matrix with  $k > m$ . This implies that  $Z_k = W_k(\hat{\lambda})^{-1}I_{k,m}$ , so the following can be deduced about the first term inside the parenthesis on the right-hand side of (5.6):

$$V_k^H W(\hat{\lambda})^{-1}\tilde{V}_k = V_k^H Z = V_k^H (V_k Z_k) = W_k(\hat{\lambda})^{-1}I_{k,m}.$$

As for the term inside the second parenthesis on the right-hand side of (5.6), we make use of the following observation:

$$V_k^H W(\hat{\lambda})^{-1} B = \begin{bmatrix} \tilde{V}_k^H \\ \hat{V}_k^H \end{bmatrix} W(\hat{\lambda})^{-1} B = \begin{bmatrix} \tilde{V}_k^H W(\hat{\lambda})^{-1} B \\ 0 \end{bmatrix}, \quad (5.7)$$

where the last equality follows, since the columns of  $\tilde{V}_k$  form an orthonormal basis for  $\text{Im}(W(\hat{\lambda})^{-1} B)$ . Putting these observations together in (5.6), we obtain

$$\begin{aligned} V_k^H W_k(\hat{\lambda})^{-2} B_k &= (V_k^H W(\hat{\lambda})^{-1} \tilde{V}_k)(\tilde{V}_k^H W(\hat{\lambda})^{-1} B) \\ &= W_k(\hat{\lambda})^{-1} I_{k,m}(\tilde{V}_k^H W(\hat{\lambda})^{-1} B) \\ &= W_k(\hat{\lambda})^{-1} I_k(V_k^H W(\hat{\lambda})^{-1} B) \\ &= W_k(\hat{\lambda})^{-1} I_k(W_k(\hat{\lambda})^{-1} B_k) = W_k(\hat{\lambda})^{-2} B_k, \end{aligned}$$

where in the third equality we exploit (5.7), and in the fourth equality we employ that  $V_k^H W(\hat{\lambda})^{-1} B = W_k(\hat{\lambda})^{-1} B_k$  which is proven in part (i).  $\square$

After these preparations we can prove our main interpolation result.

**Theorem 5.4.** *Consider a DH model (1.1) and a reduced model  $\dot{x}_k = (J_k - R_k)Q_k x_k$  with coefficients as in (5.4), and let  $W(\lambda) = (J - R)Q - \lambda I$  and  $W_k(\lambda) := (J_k - R_k)Q_k - \lambda I$ . Suppose that the subspace  $\mathcal{V}_k$  is such that*

$$\text{Im}(W(i\hat{\omega})^{-1} B), \text{Im}(W(i\hat{\omega})^{-2} B) \subseteq \mathcal{V}_k, \quad (5.8)$$

and  $W_k$  is defined as in (5.3) in terms of a matrix  $V_k$  whose columns form an orthonormal basis for  $\mathcal{V}_k$ .

- (i) *The quantity  $\tilde{\eta}^{\text{Herm}}(R; B, i\hat{\omega})$  is finite if and only if  $\tilde{\eta}_k^{\text{Herm}}(R; B, i\hat{\omega})$  is finite. If  $\tilde{\eta}^{\text{Herm}}(R; B, i\hat{\omega})$  is finite, then*

$$\tilde{\eta}^{\text{Herm}}(R; B, i\hat{\omega}) = \tilde{\eta}_k^{\text{Herm}}(R; B, i\hat{\omega}). \quad (5.9)$$

- (ii) *Moreover, if  $\tilde{\eta}^{\text{Herm}}(R; B, i\omega)$  and  $\tilde{\eta}_k^{\text{Herm}}(R; B, i\omega)$  are differentiable at  $\hat{\omega}$ , then we have*

$$\frac{d\tilde{\eta}^{\text{Herm}}(R; B, i\hat{\omega})}{d\omega} = \frac{d\tilde{\eta}_k^{\text{Herm}}(R; B, i\hat{\omega})}{d\omega}. \quad (5.10)$$

*Proof.* Without loss of generality, we may assume that the matrix  $V_k$  is such that  $V_k = \begin{bmatrix} \tilde{V}_k & \hat{V}_k \end{bmatrix}$ , with the columns of  $\tilde{V}_k$  forming an orthonormal basis for  $\text{Im}(W(i\hat{\omega})^{-1} B)$ . It suffices to prove the claims for this particular choice of orthonormal basis, because it is established in Lemma 5.2 that the function  $\tilde{\eta}_k^{\text{Herm}}(R; B, i\omega)$ , hence its derivative, are independent of the choice of  $V_k$  as long as its columns form an orthonormal basis for  $\mathcal{V}_k$ .

- (i) By the definitions of  $\tilde{H}_0(\lambda)$ ,  $\tilde{H}_{k,0}(\lambda)$ , and part (i) of Lemma 5.3, we have

$$\begin{aligned} \tilde{H}_0(i\hat{\omega}) &= (B^H Q W(i\hat{\omega})^{-1} B)^H (B^H Q W(i\hat{\omega})^{-1} B) \\ &= (B_k^H Q_k W_k(i\hat{\omega})^{-1} B_k)^H (B_k^H Q_k W_k(i\hat{\omega})^{-1} B_k) = \tilde{H}_{k,0}(i\hat{\omega}). \end{aligned}$$

This also implies that  $L(\mathbf{i}\hat{\omega}) = L_k(\mathbf{i}\hat{\omega})$  due to the uniqueness of the Cholesky factors of  $\tilde{H}_0(\mathbf{i}\hat{\omega})$ ,  $\tilde{H}_{k,0}(\mathbf{i}\hat{\omega})$ . Therefore,

$$H_0(\mathbf{i}\hat{\omega}) = L(\mathbf{i}\hat{\omega})^{-1}L(\mathbf{i}\hat{\omega})^{-H} = L_k(\mathbf{i}\hat{\omega})^{-1}L_k(\mathbf{i}\hat{\omega})^{-H} = H_{k,0}(\mathbf{i}\hat{\omega}).$$

Furthermore,

$$\begin{aligned}\tilde{H}_1(\mathbf{i}\hat{\omega}) &= L(\mathbf{i}\hat{\omega})^{-1}(B^H Q W(\mathbf{i}\hat{\omega})^{-1}B)^H L(\mathbf{i}\hat{\omega})^{-H} \\ &= L_k(\mathbf{i}\hat{\omega})^{-1}(B_k^H Q_k W_k(\mathbf{i}\hat{\omega})^{-1}B_k)^H L_k(\mathbf{i}\hat{\omega})^{-H} = \tilde{H}_{k,1}(\mathbf{i}\hat{\omega})\end{aligned}$$

and

$$H_1(\mathbf{i}\hat{\omega}) = \mathbf{i} \left( \tilde{H}_1(\mathbf{i}\hat{\omega}) - \tilde{H}_1(\mathbf{i}\hat{\omega})^H \right) = \mathbf{i} \left( \tilde{H}_{k,1}(\mathbf{i}\hat{\omega}) - \tilde{H}_{k,1}(\mathbf{i}\hat{\omega})^H \right) = H_{k,1}(\mathbf{i}\hat{\omega}).$$

Since  $H_1(\mathbf{i}\hat{\omega}) = H_{k,1}(\mathbf{i}\hat{\omega})$ , it follows that  $\tilde{\eta}^{\text{Herm}}(R; B, \mathbf{i}\hat{\omega})$  is finite if and only if  $\tilde{\eta}_k^{\text{Herm}}(R; B, \mathbf{i}\hat{\omega})$  is finite. Additionally, if  $\tilde{\eta}^{\text{Herm}}(R; B, \mathbf{i}\hat{\omega})$  is finite, then

$$\begin{aligned}\tilde{\eta}^{\text{Herm}}(R; B, \mathbf{i}\hat{\omega}) &= \max_{t \in \mathbb{R}} \lambda_{\min}(H_0(\mathbf{i}\hat{\omega}) + tH_1(\mathbf{i}\hat{\omega})) \\ &= \max_{t \in \mathbb{R}} \lambda_{\min}(H_{k,0}(\mathbf{i}\hat{\omega}) + tH_{k,1}(\mathbf{i}\hat{\omega})) = \tilde{\eta}_k^{\text{Herm}}(R; B, \mathbf{i}\hat{\omega}),\end{aligned}$$

completing the proof of (5.9).

(ii) Now let us suppose that  $\tilde{\eta}^{\text{Herm}}(R; B, \mathbf{i}\omega)$  and  $\tilde{\eta}_k^{\text{Herm}}(R; B, \mathbf{i}\omega)$  are differentiable at  $\hat{\omega}$ . To prove the interpolation property in the derivatives, we benefit from the analytical expressions [14]

$$\begin{aligned}\frac{d\tilde{\eta}^{\text{Herm}}(R; B, \mathbf{i}\hat{\omega})}{d\omega} &= v^H \left( \frac{dH_0(\mathbf{i}\hat{\omega})}{d\omega} + \hat{t} \frac{dH_1(\mathbf{i}\hat{\omega})}{d\omega} \right) v, \\ \frac{d\tilde{\eta}_k^{\text{Herm}}(R; B, \mathbf{i}\hat{\omega})}{d\omega} &= v^H \left( \frac{dH'_{k,0}(\mathbf{i}\hat{\omega})}{d\omega} + \hat{t} \frac{dH'_{k,1}(\mathbf{i}\hat{\omega})}{d\omega} \right) v,\end{aligned}\tag{5.11}$$

where

$$\hat{t} := \arg \max_t \lambda_{\min}(H_0(\mathbf{i}\hat{\omega}) + tH_1(\mathbf{i}\hat{\omega})) = \arg \max_t \lambda_{\min}(H_{k,0}(\mathbf{i}\hat{\omega}) + tH_{k,1}(\mathbf{i}\hat{\omega}))$$

and  $v$  is a unit eigenvector corresponding to  $\lambda_{\min}(H_0(\mathbf{i}\hat{\omega}) + \hat{t}H_1(\mathbf{i}\hat{\omega})) = \lambda_{\min}(H_{k,0}(\mathbf{i}\hat{\omega}) + \hat{t}H_{k,1}(\mathbf{i}\hat{\omega}))$ . Thus, it suffices to prove that  $dH_0(\mathbf{i}\hat{\omega})/d\omega = dH_{k,0}(\mathbf{i}\hat{\omega})/d\omega$  and  $dH_1(\mathbf{i}\hat{\omega})/d\omega = dH'_{k,1}(\mathbf{i}\hat{\omega})/d\omega$  in order to show (5.10). It follows from parts (i) and (ii) of Lemma 5.3 that

$$\begin{aligned}d\tilde{H}'_0(\mathbf{i}\hat{\omega})/d\omega &= -\mathbf{i}(B^H Q W(\mathbf{i}\hat{\omega})^{-1}B)^H (B^H Q W(\mathbf{i}\hat{\omega})^{-2}B) \\ &\quad + \mathbf{i}(B^H Q W(\mathbf{i}\hat{\omega})^{-2}B)^H (B^H Q W(\mathbf{i}\hat{\omega})^{-1}B) \\ &= -\mathbf{i}(B_k^H Q_k W_k(\mathbf{i}\hat{\omega})^{-1}B_k)^H (B_k^H Q_k W_k(\mathbf{i}\hat{\omega})^{-2}B_k) \\ &\quad + \mathbf{i}(B_k^H Q_k W_k(\mathbf{i}\hat{\omega})^{-2}B_k)^H (B_k^H Q_k W_k(\mathbf{i}\hat{\omega})^{-1}B_k) = d\tilde{H}'_{k,0}(\mathbf{i}\hat{\omega})/d\omega.\end{aligned}$$

Now let us determine the derivatives of the Cholesky factors, which at a given  $\omega$  satisfy

$$\tilde{H}_0(\mathbf{i}\omega) = L(\omega)L(\mathbf{i}\omega)^H \quad \text{and} \quad \tilde{H}_{k,0}(\mathbf{i}\omega) = L_k(\mathbf{i}\omega)L_k(\mathbf{i}\omega)^H.$$

Differentiating these two equations, and setting the derivatives equal to each other at  $\widehat{\omega}$  yield

$$\begin{aligned} L(\mathrm{i}\widehat{\omega}) \left( \frac{dL(\mathrm{i}\widehat{\omega})}{d\omega} \right)^H + \frac{dL(\mathrm{i}\widehat{\omega})}{d\omega} L(\mathrm{i}\widehat{\omega})^H &= L_k(\mathrm{i}\widehat{\omega}) \left( \frac{dL_k(\mathrm{i}\widehat{\omega})}{d\omega} \right)^H + \frac{dL_k(\mathrm{i}\widehat{\omega})}{d\omega} L_k(\mathrm{i}\widehat{\omega})^H \\ &= L(\mathrm{i}\widehat{\omega}) \left( \frac{dL_k(\mathrm{i}\widehat{\omega})}{d\omega} \right)^H + \frac{dL_k(\mathrm{i}\widehat{\omega})}{d\omega} L(\mathrm{i}\widehat{\omega})^H, \end{aligned}$$

where we have used that  $L(\mathrm{i}\widehat{\omega}) = L_k(\mathrm{i}\widehat{\omega})$  as established in part (i). Thus, both  $dL(\mathrm{i}\widehat{\omega})/d\omega$  and  $dL_k(\mathrm{i}\widehat{\omega})/d\omega$  are lower triangular solutions of the matrix equation

$$d\widetilde{H}_0(\mathrm{i}\widehat{\omega})/d\omega = L(\mathrm{i}\widehat{\omega})X^H + X L(\mathrm{i}\widehat{\omega})^H.$$

This linear matrix equation has a unique lower triangular solution, so  $dL(\mathrm{i}\widehat{\omega})/d\omega = dL_k(\mathrm{i}\widehat{\omega})/d\omega$ . Now, by the definitions of  $H_0(\mathrm{i}\omega)$ ,  $H_{k,0}(\mathrm{i}\omega)$ , we have

$$L(\mathrm{i}\omega)H_0(\mathrm{i}\omega)L(\mathrm{i}\omega)^H = I = L_k(\mathrm{i}\omega)H_{k,0}(\mathrm{i}\omega)L_k(\mathrm{i}\omega)^H.$$

Differentiating this equation at  $\omega = \widehat{\omega}$  yields

$$\begin{aligned} \frac{dL(\mathrm{i}\widehat{\omega})}{d\omega} H_0(\mathrm{i}\widehat{\omega}) L(\mathrm{i}\widehat{\omega})^H + L(\mathrm{i}\widehat{\omega}) \frac{dH_0(\mathrm{i}\widehat{\omega})}{d\omega} L(\mathrm{i}\widehat{\omega})^H \\ + L(\mathrm{i}\widehat{\omega}) H_0(\mathrm{i}\widehat{\omega}) \left( \frac{dL(\mathrm{i}\widehat{\omega})}{d\omega} \right)^H &= \frac{dL_k(\mathrm{i}\widehat{\omega})}{d\omega} H_{k,0}(\mathrm{i}\widehat{\omega}) L_k(\mathrm{i}\widehat{\omega})^H + \\ L_k(\mathrm{i}\widehat{\omega}) \frac{dH_{k,0}(\mathrm{i}\widehat{\omega})}{d\omega} L_k(\mathrm{i}\widehat{\omega})^H + L_k(\mathrm{i}\widehat{\omega}) H_{k,0}(\mathrm{i}\widehat{\omega}) \left( \frac{dL_k(\mathrm{i}\widehat{\omega})}{d\omega} \right)^H. \end{aligned}$$

Using that  $L(\mathrm{i}\widehat{\omega}) = L_k(\mathrm{i}\widehat{\omega})$ ,  $dL(\mathrm{i}\widehat{\omega})/d\omega = dL_k(\mathrm{i}\widehat{\omega})/d\omega$ , and  $H_0(\mathrm{i}\widehat{\omega}) = H_{k,0}(\mathrm{i}\widehat{\omega})$ , we deduce that  $dH_0(\mathrm{i}\widehat{\omega})/d\omega = dH_{k,0}(\mathrm{i}\widehat{\omega})/d\omega$ . Next we focus on the derivatives  $d\widetilde{H}_1(\mathrm{i}\omega)/d\omega$ ,  $d\widetilde{H}_{k,1}(\mathrm{i}\omega)/d\omega$ . In particular, we use that

$$L(\mathrm{i}\omega)\widetilde{H}_1(\mathrm{i}\omega)L(\mathrm{i}\omega)^H = (B^H Q W(\mathrm{i}\omega)^{-1} B)^H.$$

Differentiating both sides of the last equation at  $\omega = \widehat{\omega}$  gives rise to

$$\begin{aligned} \frac{d \left\{ L(\mathrm{i}\omega) \widetilde{H}_1(\mathrm{i}\omega) L(\mathrm{i}\omega)^H \right\}}{d\omega} \Big|_{\omega=\widehat{\omega}} &= \mathrm{i} (B^H Q W(\mathrm{i}\widehat{\omega})^{-2} B)^H \\ &= \mathrm{i} (B_k^H Q_k W_k(\mathrm{i}\widehat{\omega})^{-2} B_k)^H \\ &= \frac{d \left\{ L_k(\mathrm{i}\omega) \widetilde{H}_{k,1}(\mathrm{i}\omega) L_k(\mathrm{i}\omega)^H \right\}}{d\omega} \Big|_{\omega=\widehat{\omega}}, \end{aligned}$$

which in turn implies that

$$\begin{aligned} \frac{dL(\mathrm{i}\widehat{\omega})}{d\omega} \widetilde{H}_1(\mathrm{i}\widehat{\omega}) L(\mathrm{i}\widehat{\omega})^H + L(\mathrm{i}\widehat{\omega}) \frac{d\widetilde{H}_1(\mathrm{i}\widehat{\omega})}{d\omega} L(\mathrm{i}\widehat{\omega})^H \\ + L(\mathrm{i}\widehat{\omega}) \widetilde{H}_1(\mathrm{i}\widehat{\omega}) \left( \frac{dL(\mathrm{i}\widehat{\omega})}{d\omega} \right)^H &= \frac{dL_k(\mathrm{i}\widehat{\omega})}{d\omega} \widetilde{H}_{k,1}(\mathrm{i}\widehat{\omega}) L_k(\mathrm{i}\widehat{\omega})^H + \\ L_k(\mathrm{i}\widehat{\omega}) \frac{d\widetilde{H}_{k,1}(\mathrm{i}\widehat{\omega})}{d\omega} L_k(\mathrm{i}\widehat{\omega})^H + L_k(\mathrm{i}\widehat{\omega}) \widetilde{H}_{k,1}(\mathrm{i}\widehat{\omega}) \left( \frac{dL_k(\mathrm{i}\widehat{\omega})}{d\omega} \right)^H. \end{aligned}$$

Once again exploiting that  $L(\mathbf{i}\hat{\omega}) = L_k(\mathbf{i}\hat{\omega})$ ,  $dL(\mathbf{i}\hat{\omega})/d\omega = dL_k(\mathbf{i}\hat{\omega})/d\omega$ , as well as  $\tilde{H}_1(\mathbf{i}\hat{\omega}) = \tilde{H}_{k,1}(\mathbf{i}\hat{\omega})$  in the last equation, we obtain  $d\tilde{H}_1(\mathbf{i}\hat{\omega})/d\omega = d\tilde{H}_{k,1}(\mathbf{i}\hat{\omega})/d\omega$  which implies that

$$\begin{aligned} \frac{dH_1(\mathbf{i}\hat{\omega})}{d\omega} &= \mathbf{i} \left\{ \frac{d\tilde{H}_1(\mathbf{i}\hat{\omega})}{d\omega} - \left( \frac{d\tilde{H}_1(\mathbf{i}\hat{\omega})}{d\omega} \right)^H \right\} \\ &= \mathbf{i} \left\{ \frac{d\tilde{H}_{k,1}(\mathbf{i}\hat{\omega})}{d\omega} - \left( \frac{d\tilde{H}_{k,1}(\mathbf{i}\hat{\omega})}{d\omega} \right)^H \right\} = \frac{dH_{k,1}(\mathbf{i}\hat{\omega})}{d\omega}, \end{aligned}$$

and the proof of (5.10) is complete.  $\square$

**Remark** The function  $\tilde{\eta}^{\text{Herm}}(R; B, \mathbf{i}\omega)$  is differentiable at  $\hat{\omega}$  whenever  $\tilde{\eta}^{\text{Herm}}(R; B, \mathbf{i}\hat{\omega})$  is finite (equivalently  $H_1(\mathbf{i}\hat{\omega})$  is indefinite),  $\lambda_{\min}(H_0(\mathbf{i}\hat{\omega}) + tH_1(\mathbf{i}\hat{\omega}))$  is simple where  $\hat{t} := \arg \max_{t \in \mathbb{R}} \lambda_{\min}(H_0(\mathbf{i}\hat{\omega}) + tH_1(\mathbf{i}\hat{\omega}))$ , and the global minimum of  $\lambda_{\min}(H_0(\mathbf{i}\hat{\omega}) + tH_1(\mathbf{i}\hat{\omega}))$  over  $t$  is attained at a unique  $t$ . These conditions guarantee also the differentiability of  $\tilde{\eta}_k^{\text{Herm}}(R; B, \mathbf{i}\omega)$  at  $\hat{\omega}$  provided the subspace inclusion in (5.8) holds. This latter differentiability property is due to  $H_0(\mathbf{i}\hat{\omega}) = H_{k,0}(\mathbf{i}\hat{\omega})$  and  $H_1(\mathbf{i}\hat{\omega}) = H_{k,1}(\mathbf{i}\hat{\omega})$  from part (i) of Theorem 5.4.

Additionally, when the concave function  $g(t) := \lambda_{\min}(H_0(\mathbf{i}\hat{\omega}) + tH_1(\mathbf{i}\hat{\omega}))$  attains its maximum, the maximizer is nearly always unique. The function  $g(t)$  is the minimum of  $m$  real analytic functions [13, 19] each corresponding to an eigenvalue of  $H_0(\mathbf{i}\hat{\omega}) + tH_1(\mathbf{i}\hat{\omega})$ . If  $g(t)$  does not have a unique maximizer, then at least one of these real analytic functions must be constant and equal to  $\tilde{\eta}^{\text{Herm}}(R; B, \mathbf{i}\hat{\omega}) = \max_t g(t)$  everywhere. Thus, a simple sufficient condition that ensures the uniqueness of the maximizer is that  $H_1(\mathbf{i}\hat{\omega})$  has full rank, in which case all eigenvalues of  $H_0(\mathbf{i}\hat{\omega}) + tH_1(\mathbf{i}\hat{\omega})$  blow up (either to  $\infty$  or  $-\infty$ ) as  $t \rightarrow \infty$  implying that each of the real analytic functions is non-constant.

### 5.2.1 The Subspace Framework for $r^{\text{Herm}}(R; B)$

The Hermite interpolation result of Theorem 5.4 immediately suggests a subspace framework for the computation of  $r^{\text{Herm}}(R; B)$ . This resembles the structure preserving subspace framework to compute the unstructured stability radii  $r(R; B, C) = r(J; B, C)$ , in particular, in the way the subspaces  $\mathcal{V}_k, \mathcal{W}_k$  are built. At every iteration, a reduced problem is solved using the ideas in Section 5.1 using Algorithm 3. Letting  $\hat{\omega}$  be the global minimizer of the reduced problem, the subspaces are expanded so that the original function  $\tilde{\eta}^{\text{Herm}}(R; B, \mathbf{i}\omega)$  is Hermite interpolated by its reduced counter-part at  $\omega = \hat{\omega}$ .

Assuming that the sequence  $\{\omega_k\}$  converges to a minimizer  $\omega_*$  of the function  $\tilde{\eta}^{\text{Herm}}(R; B, \mathbf{i}\omega)$  such that  $H_1(\mathbf{i}\omega_*)$  is indefinite with full rank and  $\lambda_{\min}(H_0(\mathbf{i}\omega_*) + t_*H_1(\mathbf{i}\omega_*))$  is simple where  $t_* := \arg \max_{t \in \mathbb{R}} \lambda_{\min}(H_0(\mathbf{i}\omega_*) + tH_1(\mathbf{i}\omega_*))$ , it can be shown that the sequence  $\{\omega_k\}$  converges at a super-linear rate. Here the analysis in [3] applies. The conditions that  $H_1(\mathbf{i}\omega_*)$  is indefinite with full rank,  $\lambda_{\min}(H_0(\mathbf{i}\omega_*) + t_*H_1(\mathbf{i}\omega_*))$  is simple, and the interpolation properties  $H_0(\mathbf{i}\omega_k) = H_{k,0}(\mathbf{i}\omega_k)$ ,  $H_1(\mathbf{i}\omega_k) = H_{k,1}(\mathbf{i}\omega_k)$  ensure that the full function  $\tilde{\eta}^{\text{Herm}}(R; B, \mathbf{i}\omega)$ ,

as well as the reduced function  $\hat{\eta}_k^{\text{Herm}}(R; B, i\omega)$  for all large  $k$  are differentiable at all  $\omega$  in a ball  $\mathcal{B}(\omega_*, \delta)$  centered at  $\omega_*$  and of radius  $\delta$ . This differentiability property is essential for the applicability of the rate-of-convergence analysis in [3].

---

**Algorithm 4** Subspace method for large-scale computation of the structured stability radius  $r^{\text{Herm}}(R; B)$ .

---

**Input:** Matrices  $B \in \mathbb{C}^{n \times m}$ ,  $J, R, Q \in \mathbb{C}^{n \times n}$ .

**Output:** The sequence  $\{\omega_k\}$ .

- 1: Choose the initial interpolation points  $\omega_1, \dots, \omega_j \in \mathbb{R}$ .
  - 2:  $V_j \leftarrow \text{orth} \begin{bmatrix} W(i\omega_1)^{-1}B & W(i\omega_1)^{-2}B & \dots & W(i\omega_j)^{-1}B & W(i\omega_j)^{-2}B \end{bmatrix}$ ,  
 $W_j \leftarrow QV_j(V_j^H QV_j)^{-1}$ .
  - 3: **for**  $k = j, j+1, \dots$  **do**
  - 4:    $\omega_{k+1} \leftarrow \arg\min_{\omega \in \mathbb{R}} \hat{\eta}_k^{\text{Herm}}(R; B, i\omega)$ .
  - 5:    $\hat{V}_{k+1} \leftarrow [W(i\omega_{k+1})^{-1}B \quad W(i\omega_{k+1})^{-2}B]$ .
  - 6:    $V_{k+1} \leftarrow \text{orth} \left( \begin{bmatrix} V_k & \hat{V}_{k+1} \end{bmatrix} \right)$    and    $W_{k+1} \leftarrow QV_{k+1}(V_{k+1}^H QV_{k+1})^{-1}$ .
  - 7: **end for**
- 

### 5.3 Numerical Experiments

In this section we present several numerical tests for Algorithms 3 and 4, on synthetic examples and the FE model of the disk brake. The test set-up is similar to the one for the unstructured case in Section 4.3. In particular, for Algorithm 4, we use the same stopping criteria with the same parameters, but now in terms of  $f_k := \arg\min_{\omega} \hat{\eta}_k^{\text{Herm}}(R; B, i\omega)$ , so we terminate when  $|f_k - f_{k-1}| \leq \varepsilon|f_k + f_{k-1}|/2$  holds, or when one of the other two conditions stated in Section 4.3.1 holds. The initial subspaces are also chosen as described in that section.

#### 5.3.1 Synthetic Examples

**Small Scale Examples.** We first present numerical results for a small dense random example, where  $J, R, Q \in \mathbb{R}^{20 \times 20}$  and  $B \in \mathbb{R}^{20 \times 2}$ . These matrices are generated by means of the MATLAB commands employed for the generation of the dense family in Section 4.3.2; only here the matrices are  $20 \times 20$  instead of  $800 \times 800$  and  $R$  is of rank 5.

The spectrum of  $(J - R)Q$  is depicted on the top in Figure 7. Application of Algorithm 3 to this example yields  $r^{\text{Herm}}(R; B) = 0.0501$ , and the point that is first reached on the imaginary axis under the smallest perturbation is  $1.9794i$ , i.e.,  $\hat{\eta}_k^{\text{Herm}}(R; B, i\omega)$  is minimized at  $\omega = 1.9794$ . At the bottom of Figure 7, the spectra of matrices of the form  $(J - (R + B\Delta B^H))Q$  are plotted for 100000 randomly chosen Hermitian  $\Delta$  such that  $\|\Delta\|_2 = 0.0501$ . One can notice that some of the eigenvalues (nearly) touch the imaginary axis at  $1.9794i$  marked with a circle.

The subspace framework for this example is illustrated in Figure 8, where the solid, dashed curves correspond to the plots of the full function  $\hat{\eta}^{\text{Herm}}(R; B, i\omega)$ ,

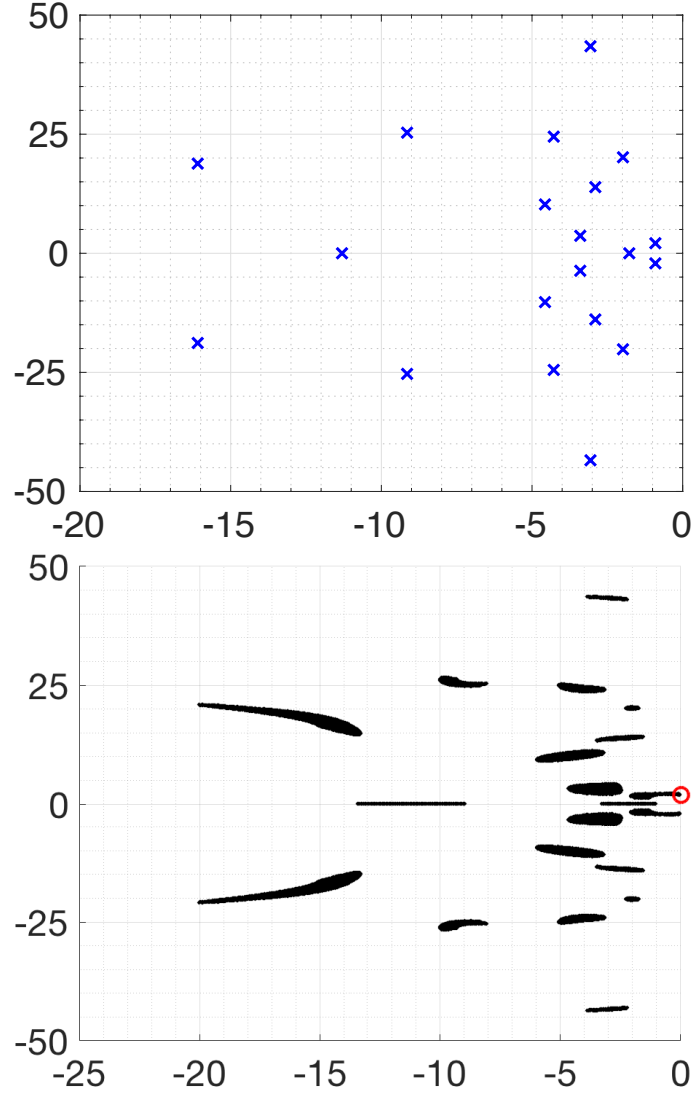


Figure 7: A DH system of order 20 with random system matrices. **(Top)** A plot of the spectrum for  $(J - R)Q$ . **(Bottom)** All eigenvalues of all matrices of the form  $(J - (R + B\Delta B^H))Q$  for 100000 randomly chosen Hermitian  $\Delta$  with  $\|\Delta\|_2 = r^{\text{Herm}}(R; B)$  are displayed. The circle marks  $1.9794i$ , the global minimizer of  $\hat{\eta}_k^{\text{Herm}}(R; B, \lambda)$  over  $\lambda \in i\mathbb{R}$ .

the reduced function  $\tilde{\eta}_k^{\text{Herm}}(R; B, i\omega)$ , respectively, and the circle represents the minimizer of  $\tilde{\eta}_k^{\text{Herm}}(R; B, i\omega)$ . On the top row, the framework is initiated with two interpolation points at 0 and near -20; the dashed curve interpolates the solid curve at these points. Then the subspaces are expanded so that the Hermitian interpolation property is also satisfied at the minimizer of the dashed curve on the top, leading to the dashed curve at the bottom, which has nearly the same global minimizer as the solid curve. Note that starting from  $\omega$  near -14 and for smaller  $\omega$  values, the matrix  $H_1(i\omega)$  turns out to be definite for this example, meaning for such  $\omega$  values the point  $i\omega$  is not attainable as an eigenvalue with Hermitian perturbations. In practice, we set the objective to be minimized at such  $\omega$  a value considerably larger than the minimal value of the objective, which in this example is 0.1.

**Large Examples.** The remaining synthetic examples are with larger random matrices. We created three sets of matrices  $J, R, Q, B$  using the commands at the beginning of Section 4.3.2 for the generation of the dense family. Each set consists of four quadruples  $J, R, Q \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times 2}$  with the same  $n$ , specifically  $n = 1000, 2000, 4000$  for the first, second, third set. The results obtained by applications of Algorithm 4 to compute  $r^{\text{Herm}}(R; B)$  are reported in Tables 7, 8, 9 for these sets, respectively.

For the first family with  $n = 1000$ , the subspace framework, i.e., Algorithm 4 already needs less computing time than Algorithm 3 except for the last example where quite a few additional subspace iterations have been performed. On this family, the direct application of Algorithm 3 and the subspace framework return exactly the same values for  $r^{\text{Herm}}(R; B)$ .

For bigger systems Algorithm 3 becomes too computationally expensive, so we do not report results here for the larger dimensions. A remarkable fact we have observed is that the number of subspace iterations to reach the prescribed accuracy is usually small and seems independent of  $n$ . By the definitions of the structured and unstructured radii, we must have  $r^{\text{Herm}}(R; B) \geq r(R; B, B^H)$ , and the presented radii in the tables are in harmony with this.

All of these examples involve optimization of highly non-convex and non-smooth functions. Figure 9 depicts  $\tilde{\eta}^{\text{Herm}}(R; B, i\omega)$  as a function of  $\omega$  with the solid curve for the first example in the first family with  $n = 1000$ . The same figure also depicts the reduced function  $\tilde{\eta}_k^{\text{Herm}}(R; B, i\omega)$  with the dashed curve for the same example at termination after 7 subspace iterations. Even though the reduced function  $\tilde{\eta}_k^{\text{Herm}}(R; B, i\omega)$  represented by the dashed curve in this plot is defined for projected matrices onto 48 dimensional subspaces, it captures the original function  $\tilde{\eta}^{\text{Herm}}(R; B, i\omega)$  remarkably well near the global minimizer  $\omega_* = -70.9623$ .

### 5.3.2 FE Model of a Disk Brake

We also applied our implementation of Algorithm 4 to the FE model of the disk brake described in Section 4.3.3 which is of the form (1.6), (1.7) with  $G(\Omega), D(\Omega), K(\Omega), M \in \mathbb{R}^{4669 \times 4669}$ , and  $J, R, Q \in \mathbb{R}^{9338 \times 9338}$ .

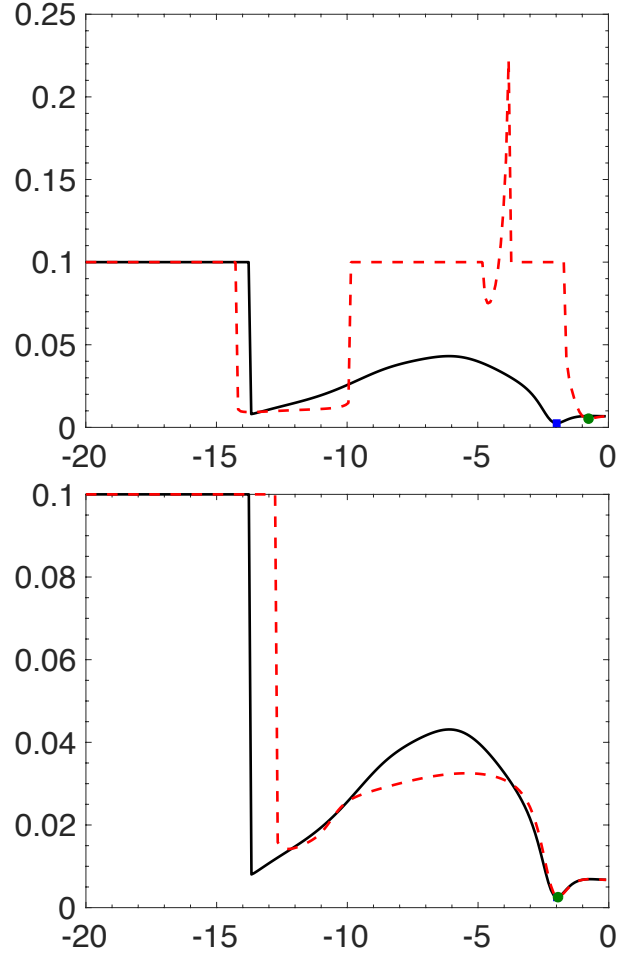


Figure 8: Progress of Algorithm 4 on a random DH system of order 20. The solid, dashed curves display  $\tilde{\eta}^{\text{Herm}}(R; B, i\omega)$ ,  $\tilde{\eta}_k^{\text{Herm}}(R; B, i\omega)$  as functions of  $\omega$ , respectively. The square, circle represent the global minima of  $\tilde{\eta}^{\text{Herm}}(R; B, i\omega)$ ,  $\tilde{\eta}_k^{\text{Herm}}(R; B, i\omega)$ . **(Top Plot)** The reduced function interpolates the full function at two points, at  $\omega = 0$  and near  $\omega = -20$ . **(Bottom Plot)** Plot of the refined reduced function after applying one subspace iteration.

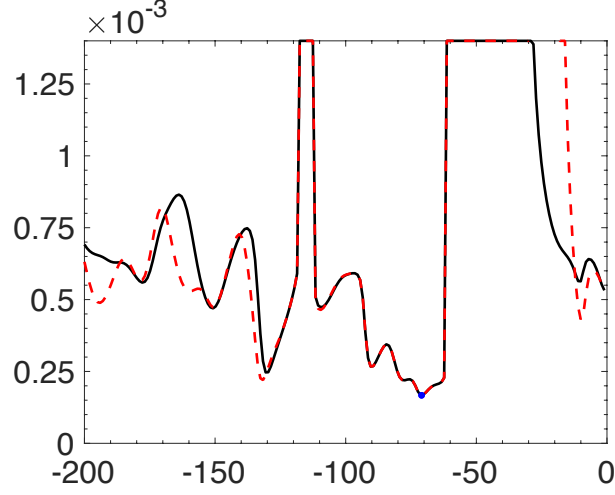


Figure 9: Application of Algorithm 4 to compute  $r^{\text{Herm}}(R; B)$  on a dense random DH system of order 1000. The full function  $\hat{\eta}_k^{\text{Herm}}(R; B, i\omega)$  and the reduced function  $\hat{\eta}_k^{\text{Herm}}(R; B, i\omega)$  at termination after 7 subspace iterations are plotted with the solid and dashed curves, respectively. The point  $(\omega_*, \hat{\eta}_k^{\text{Herm}}(R; B, i\omega_*))$  with  $\omega_*$  denoting the global minimizer of  $\hat{\eta}_k^{\text{Herm}}(R; B, i\omega)$  is marked with a circle.

#	$r^{\text{Herm}}(R; B)$		$r(R; B, B^H)$	iterations	run-time	
	Alg. 3	Alg. 4	Alg. 1	Alg. 4	Alg. 3	Alg. 4
1	0.0129	0.0129	0.0113	7	99.2	66.2
2	0.0101	0.0101	0.0101	1	66.5	7.1
3	0.0141	0.0141	0.0128	5	128.5	72.5
4	0.0096	0.0096	0.0072	20	121.2	247.2

Table 7: Performance of Algorithm 4 to compute  $r^{\text{Herm}}(R; B)$  on dense random DH systems of order 1000. For comparison, the results from Algorithm 3, as well as the unstructured radii  $r(R; B, B^H)$  by Algorithm 1 are also included. The fourth column contains the number of subspace iterations, and the fifth the run-times (in s).

	$r^{\text{Herm}}(R; B)$	$r(R; B, B^H)$	iterations	run-time
#	Alg. 4	Alg. 1	Alg. 4	Alg. 4
1	0.0102	0.0061	6	149.1
2	0.0040	0.0012	1	17.1
3	0.0125	0.0109	6	175.9
4	0.0101	0.0100	3	126.7

Table 8: Performance of Algorithm 4 to compute  $r^{\text{Herm}}(R; B)$  on dense random DH systems of order  $n = 2000$ . The fourth column contains the number of subspace iterations, and the fifth the run-times (in  $s$ ).

	$r^{\text{Herm}}(R; B)$	$r(R; B, B^H)$	iterations	run-time
#	Alg. 4	Alg. 1	Alg. 4	Alg. 4
1	0.0090	0.0087	2	282.6
2	0.0084	0.0068	3	319.3
3	0.0104	0.0086	2	333.1
4	0.0040	0.0007	46	786.5

Table 9: Performance of Algorithm 4 to compute  $r^{\text{Herm}}(R; B)$  on dense random DH systems of order  $n = 4000$ . The fourth column contains the number of subspace iterations, and the fifth the run-times (in  $s$ ).

$\Omega$	$r^{\text{Herm}}(R; B, B^T)$	$\omega_*$	iterations	run-time	dimension
2.5	0.01067	$-1.938 \times 10^5$	2	145.8	72
5	0.01038	$-1.938 \times 10^5$	2	133.5	72
10	0.01026	$-1.938 \times 10^5$	2	132.0	72
50	0.01000	$-1.938 \times 10^5$	2	132.5	72
100	0.00988	$-1.938 \times 10^5$	1	94.4	66
1000	0.00810	$-1.789 \times 10^5$	2	127.7	72
1050	0.00794	$-1.789 \times 10^5$	2	126.4	72
1100	0.00835	$-1.789 \times 10^5$	3	171.2	78
1116	0.01092	$-1.789 \times 10^5$	2	127.4	72
1150	0.00346	$-1.742 \times 10^5$	2	124.3	72
1200	0.00408	$-1.742 \times 10^5$	2	121.2	72
1250	0.00472	$-1.742 \times 10^5$	2	119.1	72
1300	0.00517	$-1.742 \times 10^5$	2	117.1	72

Table 10: Structured stability radii  $r^{\text{Herm}}(R; B, B^T)$  computed by Algorithm 4 for the FE model of a disk brake of order 9338 for several values of  $\Omega$ . The column  $\omega_*$  depicts the computed global minimizer of  $\tilde{\eta}^{\text{Herm}}(R; B, i\omega)$ , whereas the last three columns depict the number of subspace iterations, the total run-time (in  $s$ ), and the subspace dimension at termination.

The computed value of the structured radius  $r^{\text{Herm}}(R; B, B^T)$  along with the computed global minimizer  $\omega_*$  of  $\tilde{\eta}^{\text{Herm}}(R; B, i\omega)$ , as well as the number of subspace iterations, run-time (in  $s$ ) and the subspace dimension at termination are listed in Table 10 for various values of  $\Omega$ . It is worth comparing the computed values of  $r^{\text{Herm}}(R; B, B^T)$  in this table with those for the unstructured stability radius  $r(R; B, B^T)$  listed in Table 6. The computed structured and the unstructured stability radii are close, though the structured stability radii are slightly larger as expected in theory.

## 6 Concluding Remarks

We have proposed subspace frameworks to compute the stability radii for large scale dissipative Hamiltonian systems. The frameworks operate on the eigenvalue optimization characterizations of the stability radii derived in [15]. At every iteration, we apply DH structure preserving Petrov-Galerkin projections to small subspaces. This leads to the computation of the corresponding stability radii for the reduced system. We expand the subspaces used in the Petrov-Galerkin projections so that Hermite-interpolation properties between the objective eigenvalue function of the full and the reduced problems are attained at the optimizer of the reduced problem. This strategy results in super-linear convergence with respect to the subspace dimensions. We have illustrated that the frameworks work well in practice on several synthetic examples, and a FE model of a disk brake.

Matlab implementations of the proposed algorithms and subspace frameworks are made publicly available on the web<sup>1</sup>. Some of the data (including the one associated with the disk brake example) used in the numerical experiments are also available on the same website.

One difficulty is that the proposed frameworks converge only locally. As a remedy for this, we have initiated the subspaces to attain Hermite interpolation at several points on the imaginary axis between the full and initial reduced problems. One potential strategy that is currently investigated is to employ equally spaced interpolation points. Another potential strategy finds the poles closest to these equally spaced points, then employs the imaginary parts of the poles as the initial interpolation points.

Another research direction that is currently investigated is the maximization of the stability radii, when  $J, R, Q$  depend on parameters in a given parameter set. As an example, for the dissipative Hamiltonian system arising from the FE model of a disk brake, even in the simple setting considered here,  $J, R, Q$  depend on the rotation speed  $\Omega$ .

## References

- [1] A. Akay. Acoustics of friction. *J. Acoust. Soc. Am.*, 111(4):1525–1548, 2002.
- [2] A. Aliyev, P. Benner, E. Mengi, P. Schwerdtner, and M. Voigt. Large-scale computation of  $\mathcal{L}_\infty$ -norms by a greedy subspace method. *SIAM J. Matrix Anal. Appl.*, 38(4):1496–1516, 2017.
- [3] A. Aliyev, P. Benner, E. Mengi, and M. Voigt. A subspace framework for  $\mathcal{H}_\infty$ -norm minimization. Technical report, 2018. In Preparation.
- [4] S. Boyd and V. Balakrishnan. A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its  $L_\infty$ -norm. *Systems Control Lett.*, 15(1):1–7, 1990.
- [5] N. A. Bruinsma and M. Steinbuch. A fast algorithm to compute the  $H_\infty$ -norm of a transfer function matrix. *Systems Control Lett.*, 14(4):287–293, 1990.
- [6] K. Gallivan, A. Vandendorpe, and P. Van Dooren. Model reduction of MIMO systems via tangential interpolation. *SIAM J. Matrix Anal. Appl.*, 26(2):328–349, 2005.
- [7] N. Gräbner, V. Mehrmann, S. Quraishi, C. Schröder, and U. von Wagner. Numerical methods for parametric model reduction in the simulation of disc brake squeal. *Z. Angew. Math. Mech.*, 96:1388–1405, 2016.

---

<sup>1</sup><http://home.ku.edu.tr/~emengi/software/PH-stabradii.html>

- [8] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. [http://stanford.edu/~boyd/graph\\_dcp.html](http://stanford.edu/~boyd/graph_dcp.html).
- [9] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. <http://cvxr.com/cvx>, March 2014.
- [10] S. Gugercin, R. V. Polyuga, C. Beattie, and A. J. van der Schaft. Structure-preserving tangential interpolation for model reduction of port-Hamiltonian systems. *Automatica*, 48(9):1963–1974, 2012.
- [11] S. Gugercin, R. V. Polyuga, C. A. Beattie, and A. J. van der Schaft. Interpolation-based model reduction for port-Hamiltonian systems. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 5362–5369, Dec 2009.
- [12] B. Jacob and H. Zwart. *Linear port-Hamiltonian systems on infinite-dimensional spaces*. Operator Theory: Advances and Applications, 223. Birkhäuser/Springer Basel AG, Basel CH, 2012.
- [13] T. Kato. *Perturbation Theory for Linear Operators*. Springer-Verlag, Berlin Heidelberg, 1995.
- [14] P. Lancaster. On eigenvalues of matrices dependent on a parameter. *Numer. Math.*, 6:377–387, 1964.
- [15] C. Mehl, V. Mehrmann, and P. Sharma. Stability radii for linear Hamiltonian systems with dissipation under structure-preserving perturbations. *SIAM J. Matrix Anal. Appl.*, 37(4):1625–1654, 2016.
- [16] E. Mengi, E. A. Yildirim, and M. Kiliç. Numerical optimization of eigenvalues of Hermitian matrix functions. *SIAM J. Matrix Anal. Appl.*, 35(2):699–724, 2014.
- [17] R. V. Polyuga and A. J. van der Schaft. Moment matching for linear port-Hamiltonian systems. In *Control Conference (ECC), 2009 European*, pages 4715–4720, Aug 2009.
- [18] R. V. Polyuga and A. J. van der Schaft. Structure preserving moment matching for port-Hamiltonian systems: Arnoldi and Lanczos. *IEEE Trans. Autom. Cont.*, 56(6):1458–1462, June 2011.
- [19] F. Rellich. *Perturbation Theory of Eigenvalue Problems*. Notes on Mathematics and its Applications. Gordon and Breach, New York, NY, USA, 1969.

- [20] A. J. van der Schaft and D. Jeltsema. Port-Hamiltonian systems theory: An introductory overview. *Foundations and Trends in Systems and Control*, 1(2-3):173–378, 2014.
- [21] A. J. van der Schaft and R. V. Polyuga. Structure-preserving model reduction of complex physical systems. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 4322–4327, Dec 2009.
- [22] Y. Wu, B. Hamroun, Y. Le Gorrec, and B. Maschke. Structure preserving reduction of port Hamiltonian system using a modified lqg method. In *Control Conference (CCC), 2014 33rd Chinese*, pages 3528–3533, July 2014.