

# ROOT-MAX PROBLEMS, HYBRID EXPANSION-CONTRACTION, AND QUADRATICALLY CONVERGENT OPTIMIZATION OF PASSIVE SYSTEMS\*

TIM MITCHELL<sup>†</sup> AND PAUL VAN DOOREN<sup>‡</sup>

**Abstract.** We present quadratically convergent algorithms to compute the extremal value of a real parameter for which a given rational transfer function of a linear time-invariant system is passive. This problem is formulated for both continuous-time and discrete-time systems and is linked to the problem of finding a realization of a rational transfer function such that its passivity radius is maximized. Our new methods make use of a generalization of the hybrid expansion-contraction algorithm, which we have extended to the setting of what we call root-max problems.

**Key words.** positive realness, passivity, robustness, rational transfer functions

**MSC codes.** 93D09, 93C05, 49M15, 37J25

**DOI.** 10.1137/21M1450367

**1. Introduction.** Robustness measures play an important role in systems and control. They provide margins for the perturbations that one can allow on a given nominal dynamical system such that the perturbed system still performs as desired. A classical example of such a measure is the so-called *distance to instability* [31], which measures how much one can perturb a stable matrix before destabilization is a possibility. A generalization of this is the *complex stability radius* (better known by its reciprocal, the  $\mathcal{H}_\infty$  norm), which measures how much (complex-valued) uncertainty in a dynamical system with input and output can be tolerated before stability is no longer guaranteed [34, 16]. Meanwhile, the *real structured stability radius* and  $\mu$ -value further restrict the uncertainty to be real-valued or structured in a particular sense [14, 15]. Such measures are often the subject of optimization in robust control, since it is natural to desire that the robustness of models to uncertainty/perturbation be maximized. Furthermore, in the area of model order reduction, the  $\mathcal{H}_\infty$  norm is one of the main indicators of how well a reduced-order surrogate mimics the behavior of a larger (and often computationally unwieldy) system [11]. Numerical procedures for computing these robustness measures have been developed in the last few decades and have historically been focused on linear time-invariant systems described by their generalized state-space model.

In this paper, we consider a problem that is linked to maximizing the *passivity radius* [27], which measures how much one can perturb a passive system model before it may lose passivity. For a continuous-time linear time-invariant system that is *passive*, the Hermitian part of its transfer function is positive semidefinite in the

---

\*Received by the editors October 4, 2021; accepted for publication (in revised form) by M. Embree December 12, 2022; published electronically May 30, 2023.

<https://doi.org/10.1137/21M1450367>

**Funding:** Two research stays for the second author were supported by the “Visitor’s Program of the MPI Magdeburg.”

<sup>†</sup>Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany. Current address: Department of Computer Science, Queens College/ CUNY, Flushing, NY 11367 USA (tim.mitchell@qc.cuny.edu).

<sup>‡</sup>Department of Mathematical Engineering, Université catholique de Louvain, Louvain-La-Neuve, Belgium (paul.vandooren@uclouvain.be).

closed right half-plane; see Definition 4.1 later on here or [32, Theorem 1] for a complete definition of passivity. When this transfer function is rational and is described by a finite-dimensional state-space model  $\mathcal{M} := \{A, B, C, D\}$ , those conditions can be rephrased in terms of the state-space model parameters. In this paper, we consider a state-space model  $\mathcal{M}_\xi$  that is dependent on a real parameter  $\xi$  and look for an extremal value of the parameter for which the corresponding system is still passive. At this extremal value, the parametric system switches from passive to nonpassive. Computation of this extremal parameter value is important, as it allows one to construct certificates for the passivity of the parametric passive systems. As discussed in [22, 21], these certificates play a crucial role in the solution of two important problems: (i) finding a realization of a given passive system with optimal passivity radius and (ii) finding the closest passive system to a given nonpassive system. The first algorithms to compute this extremal value were recently proposed in [22] and [21], respectively, for the continuous- and discrete-time cases, but no convergence analysis was done nor were the methods tested experimentally. We address these issues, establishing that these methods have at least a superlinear rate of convergence, while also demonstrating some numerical issues with them. Most importantly, we present significantly faster new algorithms with local quadratic convergence and much smaller constant factors in their work complexities. Our new methods are also more numerically robust than the earlier techniques and have variants that are both the first algorithms for large-scale use and guarantee approximations that are locally optimal in a certain sense. Finding a nearby passive system to a nonpassive one has also been considered in [10] and [9], but it was suggested in [22, 21] that techniques like the ones we develop here could be applied to address that problem as well.

A core part of our new methods (from which they derive their quadratic convergence) is our generalization of the hybrid expansion-contraction (HEC) algorithm. HEC was first conceived as a way to approximate the  $\mathcal{H}_\infty$  norm of large-scale systems [23, 24] and was subsequently extended to approximating the real structured stability radius [12]. However, HEC and its convergence properties have only been described for these two specific settings, while the structure of our problem of interest here is quite different. Unlike the  $\mathcal{H}_\infty$  norm, which is computed by obtaining a global maximizer of a function in one real variable, the extremal value we consider here for the optimization of passive systems is computed by iterating over two real variables. Consequently, another contribution of this paper is to connect these seemingly disparate things, namely, by (i) identifying that all of these problems are actually specific instances of what we call *root-max problems* (or equivalently *root-min problems*) and (ii) generalizing HEC and its convergence results to this broad new class. Besides enabling our new methods here, we hope that our generalization of HEC will both increase awareness for identifying root-max and root-min problems and ease facilitation of new HEC-based methods.

The paper is organized as follows. We first establish notation and preliminary material in section 2. Then, in section 3, we introduce root-max problems and generalize HEC and its convergence results to this problem class. In section 4, we describe the continuous-time version of our passivity radius problem and our corresponding new algorithm to solve it, while the discrete-time case is handled in section 5. Numerical experiments and concluding remarks are, respectively, given in sections 6 and 7.

**2. Preliminaries.** We begin with notation. The set of Hermitian matrices in  $\mathbb{C}^{n \times n}$  is denoted by  $\mathbb{H}_n$  with  $A \succ 0$  ( $A \succeq 0$ ) additionally signifying that  $A \in \mathbb{H}_n$  is positive (semi)definite. The spectrum of a square matrix  $A$  is denoted  $\Lambda(A)$ , and

when  $A$  is Hermitian, we additionally use the shorthand  $\lambda_{\min}(A)$  to denote its smallest eigenvalue. Furthermore,  $\alpha(A) := \max\{\operatorname{Re} \lambda : \lambda \in \Lambda(A)\}$  denotes the *spectral abscissa* of  $A$ , while  $\rho(A) := \max\{|\lambda| : \lambda \in \Lambda(A)\}$  denotes the *spectral radius* of  $A$ ; note that  $A$  is continuous-time (discrete-time) *asymptotically stable* if and only if  $\alpha(A) < 0$  ( $\rho(A) < 1$ ) holds. The real and imaginary parts of a complex matrix  $Z$  are denoted by  $\operatorname{Re}(Z)$  and  $\operatorname{Im}(Z)$ , respectively, while the (conjugate) transpose of a vector or matrix  $V$  is denoted by  $V^T$  ( $V^H$ ). We use  $I_n$  for the  $n \times n$  identity matrix.

The models that we consider here are given by their standard state-space form, which means that their associated transfer functions are proper (i.e., bounded at infinity). In the continuous-time setting, the transfer function arises from the *Laplace transform* of the system

$$(2.1) \quad \begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \quad x(0) = 0, \\ y(t) &= Cx(t) + Du(t), \end{aligned}$$

where  $A \in \mathbb{C}^{n \times n}$ ,  $B \in \mathbb{C}^{n \times m}$ ,  $C \in \mathbb{C}^{p \times n}$ ,  $D \in \mathbb{C}^{p \times m}$ , and  $x(\cdot)$ ,  $u(\cdot)$ ,  $y(\cdot)$  are time-dependent vector-valued functions denoting, respectively, the *state*, *input*, and *output* of the system. In the discrete-time setting, the transfer function arises from the *z-transform* applied to

$$(2.2) \quad \begin{aligned} x_{k+1} &= Ax_k + Bu_k, \quad x_0 = 0, \\ y_k &= Cx_k + Du_k, \end{aligned}$$

where now  $x_k$ ,  $u_k$ , and  $y_k$  are vector-valued sequences denoting, respectively, the state, input, and output of the system. In both cases, we denote these systems by 4-tuples of matrices  $\mathcal{M} := \{A, B, C, D\}$  and their associated rational matrices

$$(2.3) \quad \mathcal{T}(\lambda) := C(\lambda I_n - A)^{-1}B + D \quad \text{and} \quad \mathcal{T}^H(\lambda) := B^H(\lambda I_n - A^H)^{-1}C^H + D^H$$

are, respectively, the associated *transfer function* and *paraconjugate transfer function*, where the variable  $\lambda$  stands for the Laplace variable  $s$  in the continuous-time setting and the delay operator  $z$  in the discrete-time case.

We restrict ourselves to system models which are *minimal*, i.e., the pair  $(A, B)$  is *controllable* (for all  $\lambda \in \mathbb{C}$ ,  $\operatorname{rank} [\lambda I_n - A \quad B] = n$ ), and the pair  $(A, C)$  is *observable* (i.e.,  $(A^H, C^H)$  is controllable). If the model is not minimal, one can always construct a minimal realization by removing the uncontrollable and unobservable parts, which can be done in a backward stable manner [30]. The conditions for passivity can then be expressed in terms of a linear matrix inequality involving the matrices of the system model  $\mathcal{M}$ . The *passivity margin* of the system model  $\mathcal{M}$  will then be shown to depend on the extremal value of a real parameter  $\xi$  for which a particular parametric system model  $\mathcal{M}_\xi$  loses its passivity property.

**3. Root-max problems and HEC.** Let  $\mathcal{D}_1 \subseteq \mathbb{R}$  be connected,  $\mathcal{D}_2 \subset \mathbb{R}^N$  be compact,  $g : \mathcal{D}_1 \times \mathcal{D}_2 \rightarrow \mathbb{R}$  be a continuous function, and  $g_x : \mathcal{D}_1 \rightarrow \mathbb{R}$  and  $g_\varepsilon : \mathcal{D}_2 \rightarrow \mathbb{R}$  be the following restrictions of  $g$ :

$$(3.1a) \quad g_x(\varepsilon) := g(\varepsilon, x), \quad \text{where } x \in \mathcal{D}_2 \text{ is fixed,}$$

$$(3.1b) \quad g_\varepsilon(x) := g(\varepsilon, x), \quad \text{where } \varepsilon \in \mathcal{D}_1 \text{ is fixed.}$$

Consider the following root-finding problem, which we call a root-max problem,

$$(3.2) \quad \text{determine an } \varepsilon \in \mathcal{D}_1 : \quad f(\varepsilon) := \max_{x \in \mathcal{D}_2} g(\varepsilon, x) = \max_{x \in \mathcal{D}_2} g_\varepsilon(x) = 0,$$

where we assume that the functions  $g_\varepsilon$  are bounded above for all  $\varepsilon \in \mathcal{D}_1$ . Suppose that there exists  $\varepsilon_{\text{lb}}, \varepsilon_0 \in \mathcal{D}_1$  with  $\varepsilon_{\text{lb}} < \varepsilon_0$  and either  $f(\varepsilon_{\text{lb}}) < 0 \leq f(\varepsilon_0)$  or  $f(\varepsilon_{\text{lb}}) > 0 \geq f(\varepsilon_0)$  holding. Then, by continuity of  $g$ , it is clear that (3.2) has at least one root  $\varepsilon_\star \in (\varepsilon_{\text{lb}}, \varepsilon_0] \subseteq \mathcal{D}_1$  such that  $f(\varepsilon_\star) = 0$ . Of course, if  $f(\varepsilon_0) = 0$  holds, then we can take  $\varepsilon_\star = \varepsilon_0$ . For our purposes in this section, it is convenient to assume the convention that  $f(\varepsilon_{\text{lb}}) < 0$  and  $0 \leq f(\varepsilon_0)$  hold, but note that each of these inequalities can be modified to be (non)strict or reversed, e.g.,  $f(\varepsilon_{\text{lb}}) \geq 0 > f(\varepsilon_0)$ , as desired for a specific setting.

Many well-known distance measures can be written in the form of the root-max problem given by (3.2) or, equivalently, as a root-min problem, where the max functions in (3.2) are switched to min functions and  $g_\varepsilon$  must then be bounded below for any fixed  $\varepsilon \in \mathcal{D}_1$ . For example, the distance to instability and the real stability radius can be naturally expressed as root-max problems. As we explain later, our particular problem of interest, the optimization of passive systems, also falls in this problem class, although we find it more natural to use the root-min form for that context.

In this section, we show that the HEC algorithm of [23, 24], which was originally conceived as a method for approximating the  $\mathcal{H}_\infty$  norm, actually generalizes to address the class of root-max and root-min problems that we have just defined here. We use the root-max form to generalize HEC and its associated convergence properties, since this maintains consistency with its initial usage as well as its name itself: HEC.

*Assumption 3.1.* We assume that the function  $g$  is continuously differentiable.

We use Assumption 3.1 to keep this section from becoming significantly more technical. In Remark 3.9, we discuss how the convergence of HEC is not critically reliant upon this smoothness condition and how it can be weakened.

*Assumption 3.2.* For any  $\varepsilon \in \mathcal{D}_1$ , we assume that we can obtain local maximizers or stationary points of  $g_\varepsilon$  and do so exactly, i.e., the norm of the gradient is zero, but finding a global maximizer of  $g_\varepsilon$  cannot be guaranteed. In other words, we cannot be guaranteed to obtain the value of  $f(\varepsilon)$  in practice, but we are guaranteed to obtain (generally locally optimal) lower bounds to it, which may or may not agree with the value of  $f(\varepsilon)$ .

*Remark 3.3.* Some comments on Assumption 3.2 are in order. Guaranteeing convergence to global maximizers of general functions, e.g., nonconcave ones, is typically not possible, and even in special cases where it is, techniques to do so are often prohibitively expensive. If one could reliably compute the value of  $f(\varepsilon)$ , then (3.2) could simply be solved using standard root-finding techniques with bracketing, e.g., regula falsi. In contrast, as we elucidate below, under the much milder and more realistic assumptions given in Assumption 3.2, standard root-finding techniques can break down when trying to solve instances of (3.2), precisely because obtaining the value of  $f(\varepsilon)$  is not guaranteed. Having an algorithm for (3.2) that performs robustly and predictably under Assumption 3.2 motivated the development of HEC, although it was not until this paper that HEC was actually considered from this general perspective.

HEC was born out of the specific desire for a faster and more reliable alternative to earlier state-of-the-art scalable methods for approximating the  $\mathcal{H}_\infty$  norm. Prior to its introduction, Guglielmi, Gürbüzbalaban, and Overton [13] had proposed an  $\mathcal{H}_\infty$ -norm approximation method that attempts to compute the unique root of a particular monotonically increasing function in one real variable; the reciprocal of this root is the  $\mathcal{H}_\infty$  norm. The main wrinkle here is that with existing techniques, evaluating the function to guaranteed accuracy would actually be more expensive than computing

the  $\mathcal{H}_\infty$  norm directly, but crucially, Guglielmi, Gürbüzbalaban, and Overton devised a powerful, scalable subroutine that efficiently computes a lower bound to the function value, which in practice, also often coincides with the true function value. Hence, they proposed using their fast subroutine inside a Newton-bisection-based outer iteration in order to compute the root of this function, and thus in turn, the  $\mathcal{H}_\infty$  norm. However, as first observed by the first author here [23, section 2.3] (see also [24, section 3.2]), this root-finding approach of Guglielmi, Gürbüzbalaban, and Overton actually can sometimes break down, converging to arbitrarily bad approximations to the  $\mathcal{H}_\infty$  norm that are not even locally optimal. Moreover, when this breakdown happens, their algorithm's typically fast local rate of convergence also degrades to linear.

In line with Remark 3.3, these breakdowns arise precisely because the function whose root is sought is not guaranteed to be computed accurately, and so using a standard root-finding method as the outer iteration is fraught with danger. The crux of the matter in the method of Guglielmi, Gürbüzbalaban, and Overton is that the sometimes inaccurate estimates for the function values can cause the bracket containing the root to be incorrectly and irrevocably updated, which in [23, 24] was coined a *bound mismatch error*; for a full description of how this comes about, see [24, section 3.2]. HEC overcomes this critical problem by instead employing one-sided convergence, which was motivated by a key observation [24, p. 994]: when only lower bounds to the (recall monotonically increasing) function are guaranteed, if the computed estimate is negative, the direction of the unique root cannot be determined, but if the computed estimate is positive, one does know that the root lies to the left. Thus, the HEC algorithm was designed to compute a decreasing sequence of upper bounds in order to converge to a root.

In applications of HEC explored so far [24, 12] where Assumption 3.2 holds, HEC often converges to roots of specific instances of (3.2), which is guaranteed if  $f(\varepsilon)$  can always be computed accurately. However, under Assumption 3.2 as stated, HEC instead guarantees convergence to what we call a *pseudoroot* of (3.2), which is either an actual root of (3.2) or, roughly speaking, a locally optimal approximation to one; we will define this notion exactly momentarily. In [23, 24], no name was given for this concept as it was not considered in that context.

Although HEC uses bracketing, the facts that it (i) only ever updates its upper bound and (ii) deliberately uses one-sided convergence to roots make HEC strikingly different to other root-finding methods. But, as the upcoming theoretical results will clarify, HEC's one-sided convergence does not come at the cost of sacrificing fast local convergence; under mild smoothness assumptions, the local rate of convergence of HEC is at least quadratic. In [24, p. 997], HEC is described as an “adaptively positively or negatively damped Newton method,” which means that HEC sometimes takes steps smaller or larger, respectively, than the regular Newton step. It may seem unintuitive, but the local rate of convergence of HEC does remain at least quadratic even when HEC takes smaller steps (positive damping). Meanwhile, the ability of HEC to take larger steps (negative damping) is a good thing. Suppose that HEC converges to a root  $\tilde{\varepsilon}$ .<sup>1</sup> Due to the one-sided convergence, a step larger than the Newton one can never overshoot  $\tilde{\varepsilon}$ , and so larger steps will always make more progress towards  $\tilde{\varepsilon}$  than the corresponding Newton steps would. Thus, when negative damping is frequent, HEC can be faster than Newton's method. For illustrations of positive and negative damping, please see [24, Fig. 4].

<sup>1</sup>When  $f$  has multiple roots, negative damping may bias HEC towards finding roots closer to  $\varepsilon_{1b}$  before its local convergence behavior sets in.

### 3.1. The generalized HEC algorithm and its convergence properties.

Having put HEC and its properties in the context of its own history and root finding, we now set to the task of precisely describing how HEC actually works and generalizing it to root-max problems (3.2). The convergence properties that we establish for our generalized version of HEC are, at a very high level, proved using similar arguments to those given by the first author here and Overton in [24, section 4] for the specific case of approximating the  $\mathcal{H}_\infty$  norm. However, our generalization here makes these convergence results far more accessible in terms of being much easier to both understand and apply far more broadly.

**DEFINITION 3.4.** *Given  $\tilde{\varepsilon} \in \mathcal{D}_1$  and  $\tilde{x} \in \mathcal{D}_2$ ,  $(\tilde{\varepsilon}, \tilde{x})$  is a pseudoroot of (3.2) if  $g(\tilde{\varepsilon}, \tilde{x}) = 0$  and  $\tilde{x}$  is a stationary point of  $g_{\tilde{\varepsilon}}$ .*

Defining pseudoroot in terms of a stationary point of  $g_{\tilde{\varepsilon}}$ , as opposed to a local maximizer, which might seem more intuitive, is intentional. The reason for this is subtle and requires more context to explain, so we defer this discussion to Remark 3.8. As we see in the following simple result (whose proof we omit as it is elementary), pseudoroots are intimately related to roots of (3.2).

**LEMMA 3.5.** *Let  $\tilde{\varepsilon} \in \mathcal{D}_1$ ,  $\tilde{x} \in \mathcal{D}_2$ , and  $(\tilde{\varepsilon}, \tilde{x})$  be a pseudoroot of (3.2). Then  $\tilde{\varepsilon}$  is a root of (3.2) if and only if  $\tilde{x}$  is a global maximizer of  $g_{\tilde{\varepsilon}}$ . Otherwise,  $0 < f(\tilde{\varepsilon})$ .*

As subroutines, HEC requires both a root-finding method with bracketing and an optimization solver, and we assume these subroutines have the following properties.

*Assumption 3.6.* We assume that the root-finding and optimization subroutines used by HEC are deterministic, i.e., they return the same answer for the same initial data, converge exactly (see also Assumption 3.2), and the root-finding method uses bracketing to ensure convergence to a root, while the optimization solver is monotonic, i.e., it always increases the value of the objective function being maximized at successive iterates until it reaches a stationary point.

Many root-finding methods use bracketing, while unconstrained optimization solvers are typically monotonic by design. Most solvers for these problems are also deterministic, and so this set of assumptions is mild. The remaining assumption that the subroutines converge exactly does not hold in inexact arithmetic, but this assumption is only used to establish our theoretical results. In practice, good implementations of HEC behave as the theory predicts as long as the subroutines are reasonably accurate.

We now are ready to describe how the (generalized) HEC algorithm works. As we define the algorithm here, readers may also wish to refer to Figure 1, which illustrates HEC converging to a pseudoroot  $(\tilde{\varepsilon}, \tilde{x})$ , where  $\tilde{\varepsilon}$  is indeed a root of  $f$  and  $\tilde{x}$  is a maximizer of  $g_{\tilde{\varepsilon}}$ . By construction, HEC generates a monotonically decreasing sequence  $\{\varepsilon_k\}$  that converges to  $\tilde{\varepsilon}$ . For  $x_k \in \mathcal{D}_2$  fixed with  $g_{x_k}(\varepsilon_k) \geq 0$ , first note that we have

$$g_{x_k}(\varepsilon_{\text{lb}}) \leq f(\varepsilon_{\text{lb}}) < 0 \leq g_{x_k}(\varepsilon_k) \leq f(\varepsilon_k).$$

The *one-parameter contraction phase* reduces  $\varepsilon_k$  by finding a root  $\hat{\varepsilon}_k \in (\varepsilon_{\text{lb}}, \varepsilon_k]$  of  $g_{x_k}$ . By the inequalities above, there must be at least one root in this bracket. If  $g_{x_k}(\varepsilon_k) = 0$ , the contraction phase simply returns  $\hat{\varepsilon}_k = \varepsilon_k$ . Otherwise, bisection can be used to find a root in  $(\varepsilon_{\text{lb}}, \varepsilon_k)$ . But if  $g_{x_k}$  is sufficiently smooth at  $\hat{\varepsilon}_k$ , then, e.g., Newton's or Halley's method could find it with far fewer iterations. Of course, these faster root-finding methods are not guaranteed to converge and  $g_{x_k}$  may not always be sufficiently smooth, which is why, per Assumption 3.6, it is important to combine both approaches, e.g., Newton-bisection. Bracketing and bisection ensure convergence

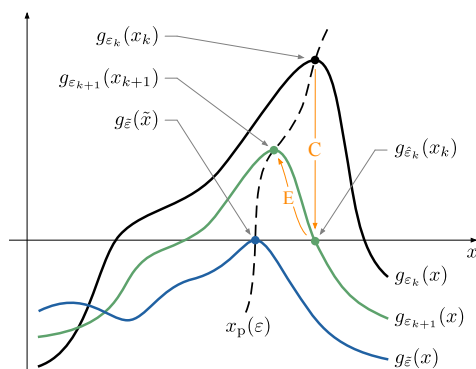


FIG. 1. Illustration of HEC converging to a pseudoroot  $(\tilde{\epsilon}, \tilde{x})$  of (3.2), where  $\tilde{\epsilon}$  is also a root of (3.2). At iteration  $k$ , HEC has found the global maximizer  $x_k$  of  $g_{\epsilon_k}$ . The contraction phase, denoted by “C” above, computes  $\epsilon_{k+1} = \hat{\epsilon}_k \leq \epsilon_k$ , which is a root of  $g_{\epsilon_k}$  in the interval  $(\epsilon_{1b}, \epsilon_k]$ . The subsequent expansion phase, denoted by “E” above, then computes the global maximizer  $x_{k+1}$  of  $g_{\epsilon_{k+1}}$ . The dashed curve  $x_p : \mathcal{D}_1 \rightarrow \mathcal{D}_2$  denotes a continuous path of maximizers of  $g_\epsilon$  as  $\epsilon$  is varied, where  $x_p(\tilde{\epsilon}) = \tilde{x}$  corresponds to the pseudoroot  $(\tilde{\epsilon}, \tilde{x})$ . Sufficiently close to  $(\tilde{\epsilon}, \tilde{x})$ , it is typical that HEC only encounters a single path  $x_p$  as depicted here.

to a root of  $g_{x_k}$ , but the ability to also take Newton (or Halley) steps, assuming that they fall inside the current bracket, can yield quadratic (or cubic) convergence when sufficient smoothness holds. Subsequently, for  $\hat{\epsilon}_k \in \mathcal{D}_1$  now fixed and  $g_{\hat{\epsilon}_k}(x_k) = 0$ , the *multiparameter expansion phase* attempts to maximize  $g_{\hat{\epsilon}_k}$  by initializing an optimization solver at  $x_k$ . If optimization returns  $x_{k+1} = x_k$ , there is nothing to do, e.g., when  $x_k$  is a stationary point of  $g_{\hat{\epsilon}_k}$ . Otherwise, since the optimization solver is monotonic by Assumption 3.6, the solver must converge to a stationary point  $x_{k+1}$  of  $g_{\hat{\epsilon}_k}$  (typically a maximizer) such that  $g_{\hat{\epsilon}_k}(x_{k+1}) > 0$ . Beyond the conditions in Assumption 3.6, HEC does not specify a specific optimization method, though fast methods should be used when possible. This process of alternating between root finding (contraction) and optimization (expansion) is repeated in a loop and it converges to a pseudoroot of (3.2). Pseudocode for HEC is given in Algorithm 3.1.

**THEOREM 3.7** (convergence of HEC). *Under Assumptions 3.1 and 3.6 and given valid initial data, Algorithm 3.1 generates the sequences  $\{\epsilon_k\}$  converging monotonically to a limit  $\tilde{\epsilon}$  and  $\{x_k\}$  with at least one cluster point, where  $(\tilde{\epsilon}, \tilde{x})$  is a pseudoroot of (3.2).*

*Proof.* We assume that the conditional statement in line 4 of Algorithm 3.1 is never met, as otherwise the theorem clearly holds. Since the algorithm ensures that  $\{\epsilon_k\}$  is a monotonically decreasing sequence that is bounded below by  $\epsilon_{1b}$ , it must converge to a limit  $\tilde{\epsilon}$ , and so it follows that  $\lim_{k \rightarrow \infty} \hat{\epsilon}_k = \lim_{k \rightarrow \infty} \epsilon_{k+1} = \tilde{\epsilon}$  as well. By construction, for all  $k \geq 1$ , the algorithm also ensures that  $g_{\epsilon_k}(x_k) > 0$  with  $x_k$  being a stationary point of  $g_{\epsilon_k}$ . Now suppose that  $\lim_{k \rightarrow \infty} g(\epsilon_k, x_k) \neq 0$ . Then there is a subsequence  $\{x_{k_i}\}$  for which  $\{g(\epsilon_{k_i}, x_{k_i})\}$  is bounded below by some  $\gamma > 0$ . Thus, by taking a further subsequence if necessary, we may assume without loss of generality that  $\{x_{k_i}\}$  converges to a limit  $\tilde{x}$ . By the continuity of  $g$ , it follows that  $\{g(\epsilon_{k_i}, x_{k_i})\}$  converges to  $g(\tilde{\epsilon}, \tilde{x}) \geq \gamma$ . However, since  $\{\hat{\epsilon}_{k_i}\}$  also converges to  $\tilde{\epsilon}$ , then  $\{g(\hat{\epsilon}_{k_i}, x_{k_i})\}$  must converge to the same limit  $g(\tilde{\epsilon}, \tilde{x})$ , which is a contradiction, since by definition of the contraction step,  $g_{x_{k_i}}(\hat{\epsilon}_{k_i}) = 0$  must hold for all  $i$ . Thus,  $\lim_{k \rightarrow \infty} g(\epsilon_k, x_k) = 0$  must hold. Although  $\{x_k\}$  may not converge, the sequence is bounded since  $\mathcal{D}_2$

**Algorithm 3.1** HEC.**Input:**  $\varepsilon_{\text{lb}}, \varepsilon_0 \in \mathcal{D}_1$  and  $x_0 \in \mathcal{D}_2$  such that  $f(\varepsilon_{\text{lb}}) < 0 \leq g(\varepsilon_0, x_0) \leq f(\varepsilon_0)$ **Output:**  $(\tilde{\varepsilon}, \tilde{x})$  such that  $g(\tilde{\varepsilon}, \tilde{x}) = 0$  and  $\tilde{x}$  is a stationary point of  $g_{\tilde{\varepsilon}}$ 

```

1: for  $k = 0, 1, 2, \dots$  do
2:   // Contraction: deterministic root-finding method initialized at  $\varepsilon_k$ 
3:    $\hat{\varepsilon}_k \leftarrow$  a root of  $g_{x_k}$  with  $\hat{\varepsilon}_k \in (\varepsilon_{\text{lb}}, \varepsilon_k]$ 
4:   if  $x_k$  is a stationary point of  $g_{\hat{\varepsilon}_k}$  then
5:      $(\tilde{\varepsilon}, \tilde{x}) \leftarrow (\hat{\varepsilon}_k, x_k)$ 
6:     return
7:   end if
8:   // Expansion: deterministic optimization method initialized at  $x_k$ 
9:    $x_{k+1} \leftarrow$  a stationary point of  $g_{\hat{\varepsilon}_k}$  with  $g_{\hat{\varepsilon}_k}(x_{k+1}) > g_{\hat{\varepsilon}_k}(x_k)$ 
10:   $\varepsilon_{k+1} \leftarrow \hat{\varepsilon}_k$ 
11: end for

```

NOTE: If the conditional statement in line 4 is never satisfied, then by Theorem 3.7, HEC produces two infinite sequences  $\{\varepsilon_k\}$  and  $\{x_k\}$ , with the former converging to  $\tilde{\varepsilon}$  and the latter having at least one cluster point, any of which we denote as  $\tilde{x}$ . Contraction must use a root-finding method with bracketing, e.g., Newton-bisection, to ensure a root of  $g_{x_k}$  in the given bracket  $(\varepsilon_{\text{lb}}, \varepsilon_k]$  is found. The inequality in the expansion phase is guaranteed by simply initializing the optimization at  $x_k$  and using a monotonic optimization solver. Finally, HEC can begin with either an expansion or contraction phase, and which is more convenient may depend on the particular application.

is a compact subset of  $\mathbb{R}^N$ , and so  $\{x_k\}$  must have at least one cluster point. As  $\|\nabla g_{\varepsilon_k}(x_k)\| = 0$  holds for all  $k \geq 1$ , clearly  $\|\nabla g_{\tilde{\varepsilon}}(\tilde{x})\| = 0$  also holds, and so  $\tilde{x}$  is also a stationary point of  $g_{\tilde{\varepsilon}}$ , hence  $(\tilde{\varepsilon}, \tilde{x})$  is a pseudoroot of (3.2).  $\square$

*Remark 3.8.* Stationary points of  $g_{\hat{\varepsilon}_k}$  computed in the expansion phases will typically be maximizers, and some optimization solvers can guarantee convergence to maximizers (under appropriate assumptions). However, while Theorem 3.7 guarantees that HEC converges to a pseudoroot  $(\tilde{\varepsilon}, \tilde{x})$  of (3.2), it does not guarantee that  $\tilde{x}$  is a local maximizer of  $g_{\tilde{\varepsilon}}$ , just that it is a stationary point. Nevertheless, whenever the expansion phases consistently return local maximizers, we do observe in practice that  $\tilde{x}$  is also a local maximizer; see [24, 12]. While it seems unlikely that  $\tilde{x}$  would only be stationary, we do not believe it is impossible; e.g., it is easy to imagine that the functions  $g_{\varepsilon_k}$  shown in Figure 1 could instead converge to a function  $g_{\tilde{\varepsilon}}$  that is constant in an interval about  $\tilde{x}$ .

*Remark 3.9.* It is only in the last sentence of the proof of Theorem 3.7 that Assumption 3.1 is used. However, Theorem 3.7 can be extended to functions  $g_{\varepsilon}$  that have some nonsmoothness, e.g., at maximizers, if one instead uses a concept of stationarity that can both handle nonsmooth points and remains continuous so that the limit argument in the proof still holds.

Although by construction, Algorithm 3.1 produces a monotonically decreasing sequence  $\{\varepsilon_k\}$ , note that the sequence  $\{g_{\varepsilon_k}(x_k)\}$  produced by the expansion phases is not necessarily monotonic, even though it must converge to zero. For example, if the highest two curves in Figure 1 were to instead cross each other to the left and right of the continuous path of global maximizers  $x_p$ , then  $g_{\varepsilon_{k+1}}(x_{k+1}) > g_{\varepsilon_k}(x_k)$  would hold. Moreover, nonmonotonicity of  $\{g_{\varepsilon_k}(x_k)\}$  can also result from HEC encountering multiple such paths of stationary points as it progresses. These paths can consist of global or local maximizers or sometimes even both. Figure 2 shows a depiction where



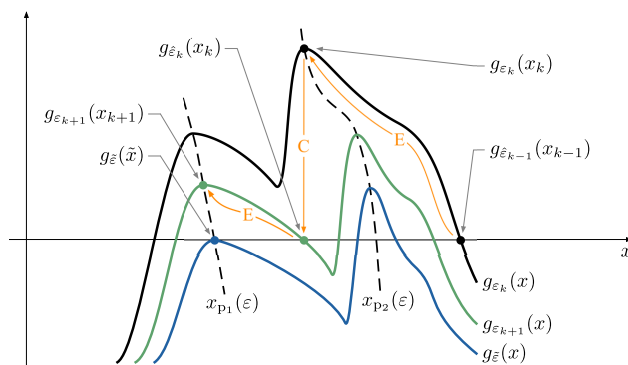


FIG. 2. Illustration of HEC of encountering two different paths of stationary points,  $x_{p_1} : \mathcal{D}_1 \rightarrow \mathcal{D}_2$  and  $x_{p_2} : \mathcal{D}_1 \rightarrow \mathcal{D}_2$ , with HEC eventually converging to a pseudoroot  $(\tilde{\epsilon}, \tilde{x})$  of (3.2) on path  $x_{p_1}$ , but  $\tilde{\epsilon}$  is not a root of (3.2). Note that paths of stationary points do not necessarily need to contain a pseudoroot (intersect with the  $x$ -axis), and although HEC may encounter and/or oscillate between multiple such paths as the algorithm converges, this does not affect the convergence result for HEC described by Theorem 3.7. For more details, see the caption of Figure 1.

$x_{p_1}$  and  $x_{p_2}$  are two separate continuous paths of local maximizers of  $g_\epsilon$  and HEC encounters both paths, but in this illustration,  $\{g_{\epsilon_k}(x_k)\}$  is monotonically converging to zero. Again, encountering multiple such paths does not affect the convergence result of Theorem 3.7. However, to show that the sequence  $\{\epsilon_k\}$  generated by HEC converges quadratically to  $\tilde{\epsilon}$ , it will be simpler to assume that HEC eventually only encounters a single continuous path of local maximizers, like shown in Figure 1.

In the following results, we use the notion of Q-quadratic and Q-superlinear convergence, where “Q” stands for “quotient”; see [26, p. 619] for more details.

**THEOREM 3.10** (quadratic convergence of HEC). *Suppose that Assumptions 3.1 and 3.6 hold, and so with valid initial data, Algorithm 3.1 converges as described in Theorem 3.7. Additionally suppose that the sequence  $\{x_k\}$  only has a single cluster point  $\tilde{x}$ , and  $\tilde{x}$  lies on an open continuous path  $x_p : \mathcal{D}_1 \rightarrow \mathcal{D}_2$  of stationary points of  $g_\epsilon$  as  $\epsilon$  varies with  $\tilde{x} = x_p(\tilde{\epsilon})$ . If  $g$  and  $x_p$  are twice continuously differentiable at  $(\tilde{\epsilon}, \tilde{x})$  and  $\tilde{\epsilon}$ , respectively,  $\tilde{x}$  is a local maximizer of  $g_{\tilde{\epsilon}}$ , and  $g'_{\tilde{x}}(\tilde{\epsilon}) \neq 0$ , then the sequence  $\{\epsilon_k\}$  converges Q-quadratically to  $\tilde{\epsilon}$ .*

*Proof.* We begin by defining the function

$$(3.3) \quad h(\epsilon) := g(\epsilon, x_p(\epsilon)).$$

Note that  $h(\tilde{\epsilon}) = g(\tilde{\epsilon}, \tilde{x}) = 0$  as  $(\tilde{\epsilon}, \tilde{x})$  is a pseudoroot of (3.2). Since the sequence  $\{x_k\}$  only has one cluster point, there also exists some  $K$  such that for all  $k \geq K$ , all of the following properties hold:

- (i) point  $x_k$  lies on path  $x_p$  with  $x_k = x_p(\epsilon_k)$ , and so  $h(\epsilon_k) = g_{x_k}(\epsilon_k)$ ;
- (ii)  $h'(\epsilon_k) = g'_{x_k}(\epsilon_k) \neq 0$ ;
- (iii)  $h$  and  $g_{x_k}$  are twice continuously differentiable at  $\epsilon_k$ .

By our assumptions, all of these statements also hold at  $\tilde{\epsilon}$ . The agreement of the first derivatives in (ii) follows from the envelope theorem (or more generally, [29, Theorem 10.31]), since maximizers of  $g_\epsilon$  do not occur at  $\infty$  as  $\mathcal{D}_2$  is compact.

Having established the needed properties above, we now consider the corresponding Newton steps for  $h$  and  $g_{x_k}$  evaluated at  $\epsilon_k$ , which also must coincide, i.e.,

$$(3.4) \quad \varepsilon_k^N := \varepsilon_k - \frac{h(\varepsilon_k)}{h'(\varepsilon_k)} = \varepsilon_k - \frac{g_{x_k}(\varepsilon_k)}{g'_{x_k}(\varepsilon_k)}.$$

However, Algorithm 3.1 sets  $\varepsilon_{k+1} := \hat{\varepsilon}_k$ , where  $g_{x_k}(\hat{\varepsilon}_k) = 0$ . Separately applying Taylor's theorem to  $h$  and  $g_{x_k}$ , we have that

$$0 = h(\tilde{\varepsilon}) = h(\varepsilon_k) + h'(\varepsilon_k)(\tilde{\varepsilon} - \varepsilon_k) + \frac{1}{2}h''(\xi_k)(\tilde{\varepsilon} - \varepsilon_k)^2$$

for some  $\xi_k \in [\tilde{\varepsilon}, \varepsilon_k]$  and

$$0 = g_{x_k}(\hat{\varepsilon}_k) = g_{x_k}(\varepsilon_k) + g'_{x_k}(\varepsilon_k)(\hat{\varepsilon}_k - \varepsilon_k) + \frac{1}{2}g''_{x_k}(\eta_k)(\hat{\varepsilon}_k - \varepsilon_k)^2$$

for some  $\eta_k \in [\tilde{\varepsilon}, \varepsilon_k]$ . Respectively dividing the two equations above by  $h'(\varepsilon_k)$  and  $g'_{x_k}(\varepsilon_k)$ , and then subtracting the first from the second and using (3.4) along with  $\varepsilon_{k+1} = \hat{\varepsilon}_k$ , we obtain

$$(3.5) \quad \varepsilon_{k+1} - \tilde{\varepsilon} = c_k(\tilde{\varepsilon} - \varepsilon_k)^2 + d_k(\varepsilon_{k+1} - \varepsilon_k)^2,$$

where

$$c_k = \frac{h''(\xi_k)}{2h'(\varepsilon_k)} \quad \text{and} \quad d_k = -\frac{g''_{x_k}(\eta_k)}{2g'_{x_k}(\varepsilon_k)}.$$

To establish quadratic convergence, we need to bound  $\varepsilon_{k+1} - \varepsilon_k$  in terms of  $\tilde{\varepsilon} - \varepsilon_k$ . To do this, consider the Taylor expansions of  $h$  and  $g_{x_k}$  but with only the first two terms, i.e.,

$$0 = h(\tilde{\varepsilon}) = h(\varepsilon_k) + h'(\zeta_k)(\tilde{\varepsilon} - \varepsilon_k)$$

for some  $\zeta_k \in [\tilde{\varepsilon}, \varepsilon_k]$  and

$$0 = g_{x_k}(\hat{\varepsilon}_k) = g_{x_k}(\varepsilon_k) + g'_{x_k}(\tau_k)(\hat{\varepsilon}_k - \varepsilon_k)$$

for some  $\tau_k \in [\tilde{\varepsilon}, \varepsilon_k]$ . As  $h(\varepsilon_k) = g_{x_k}(\varepsilon_k)$  and  $\varepsilon_{k+1} = \hat{\varepsilon}_k$ , it follows that

$$(3.6) \quad \frac{\varepsilon_{k+1} - \varepsilon_k}{\tilde{\varepsilon} - \varepsilon_k} = \frac{h'(\zeta_k)}{g'_{x_k}(\tau_k)},$$

which converges to 1 as  $k \rightarrow \infty$ , since  $h'(\zeta_k)$  and  $g'_{x_k}(\tau_k)$  both<sup>2</sup> converge to  $h'(\tilde{\varepsilon}) \neq 0$ . Dividing (3.5) by  $(\varepsilon_k - \tilde{\varepsilon})^2$  and taking the absolute value yields

$$\frac{|\varepsilon_{k+1} - \tilde{\varepsilon}|}{(\varepsilon_k - \tilde{\varepsilon})^2} = \left| c_k + d_k \left( \frac{\varepsilon_{k+1} - \varepsilon_k}{\varepsilon_k - \tilde{\varepsilon}} \right)^2 \right|.$$

By (3.6), the squared term on the right converges to 1 as  $k \rightarrow \infty$ , while  $c_k$  and  $d_k$  also converge since their numerators are bounded and their denominators each converge to  $h'(\tilde{\varepsilon}) \neq 0$ . Thus, Algorithm 3.1 converges Q-quadratically.  $\square$

*Remark 3.11.* A key part of the proof of Theorem 3.10 is that the derivatives of  $h$  and  $g_{x_k}$  coincide at  $\varepsilon_k$ , which holds because under Assumption 3.6, local maximizers of  $g_{x_k}$  are computed exactly. However, for some applications, it may actually be

<sup>2</sup>Note that in [24, p. 1000], there is a typo: in the second to last line of the proof of Theorem 4.4,  $g'_{u_k v_k}(\varepsilon_k)$  actually should be  $g'_{u_k v_k}(\tau_k)$ .

more efficient to solve the expansion phases inexactly at first, which in [24, section 4.3] is called early contraction. If the expansion phases are solved inexactly, but the inexactness goes to zero in the limit, then HEC still converges at least Q-superlinearly; see [23, sections 3.1 and 3.2] and [7]. This is useful because when the expansion phases are expensive and require many iterations of optimization, the early contraction strategy can significantly reduce the cost of the expansion phases while only slightly increasing the total number of HEC iterations. This can result in significantly faster overall runtimes; see [24, section 8]. As a final comment, note that the quadratic and superlinear rate of convergence results for HEC discussed in this paper hold regardless of how fast the contraction and expansions phases are solved; e.g., if the contraction and expansion phases are solved with linearly convergent methods, HEC still has at least quadratic convergence (or superlinear if early contraction is used).

*Remark 3.12.* If HEC converges to a finite number of cluster points of  $\{x_k\}$ , rather than a unique one as supposed in Theorem 3.10, then it is easy to see that if all the other conditions of Theorem 3.10 hold for any subsequence  $\{x_{k_i}\}$  converging to a particular cluster point, then  $\{\varepsilon_{k_i}\}$  must converge at least quadratically. Thus, if these conditions also hold for any subsequence to any of the finitely many cluster points, then we expect that the rate of convergence of  $\{\varepsilon_k\}$  should still be quadratic.

**4. Continuous-time passive systems.** Returning to the optimization of passive systems, we first consider the continuous-time case where the finite-dimensional state-space model  $\mathcal{M} := \{A, B, C, D\}$  is minimal and is given by (2.1) and its corresponding transfer function  $\mathcal{T}$  (2.3) is thus proper. Furthermore, for the remainder of the paper, we consider transfer functions  $\mathcal{T}$  of passive systems (so  $m = p$ ). We begin with the theoretical background defining the passivity optimization problem we wish to solve, which as we will show, is equivalent to a root-min problem.

**4.1. Passivity of continuous-time proper parametric systems.** The material here in this subsection is mostly drawn from [22] but is recalled here in a concise way so that we can easily refer to it. We also briefly recall definitions and properties following [32] and refer to the literature for proofs and more details.

Given  $\mathcal{T}$ , consider the following rational matrix function of  $s \in \mathbb{C}$ :

$$\Phi(s) := \mathcal{T}^H(-s) + \mathcal{T}(s),$$

which coincides with twice the Hermitian part of  $\mathcal{T}(s)$  on the imaginary axis:

$$\Phi(i\omega) = [\mathcal{T}(i\omega)]^H + \mathcal{T}(i\omega).$$

DEFINITION 4.1. *The continuous-time system  $\mathcal{M}$  with transfer function  $\mathcal{T}$  is*

1. *passive if  $\Phi(i\omega) \succeq 0$  for all  $\omega \in \mathbb{R} \cup \{\infty\}$  and  $\alpha(A) \leq 0$  with any eigenvalues of  $A$  occurring on the imaginary axis being semisimple and with a transfer function residue that is Hermitian and positive semidefinite;*
2. *strictly passive if  $\Phi(i\omega) \succ 0$  for all  $\omega \in \mathbb{R} \cup \{\infty\}$  and  $\alpha(A) < 0$ .*

Using the matrix

$$(4.1) \quad W_c(X, \mathcal{M}) := \begin{bmatrix} -A^H X - X A & C^H - X B \\ C - B^H X & D^H + D \end{bmatrix},$$

we have the following necessary and sufficient conditions for passivity of a finite-dimensional continuous-time system in state-space form; see [32].

**THEOREM 4.2.** *Let  $\mathcal{M} := \{A, B, C, D\}$  be a continuous-time minimal system and let its transfer function  $\mathcal{T}$  thus be proper. Then  $\mathcal{M}$  is (strictly) passive if and only if there exists an  $X \in \mathbb{H}_n$  such that  $X \succ 0$  and  $W_c(X, \mathcal{M}) \succeq 0$  ( $W_c(X, \mathcal{M}) \succ 0$ ).*

In [22], the following class of systems, parameterized by  $\xi \in \mathbb{R}$ , was considered:

$$(4.2a) \quad \mathcal{M}_\xi := \{A_\xi, B, C, D_\xi\} = \{A + \frac{\xi}{2}I_n, B, C, D - \frac{\xi}{2}I_m\},$$

$$(4.2b) \quad \mathcal{T}_\xi(s) := C(sI - A_\xi)^{-1}B + D_\xi = C((s - \frac{\xi}{2})I_n - A)^{-1}B + D - \frac{\xi}{2}I_m,$$

$$(4.2c) \quad \Phi_\xi(s) := \mathcal{T}_\xi^H(-s) + \mathcal{T}_\xi(s).$$

For perturbations  $\Delta_{\mathcal{M}} = \{\Delta_A, \Delta_B, \Delta_C, \Delta_D\}$  allowed in the system model  $\mathcal{M}$ , the size of the smallest value of  $\|[\begin{smallmatrix} \Delta_A & \Delta_B \\ \Delta_C & \Delta_D \end{smallmatrix}]\|_F$  at which the perturbed model  $\mathcal{M} + \Delta_{\mathcal{M}}$  loses passivity depends on this scalar parameter  $\xi$  only; see [22]. It is therefore important to compute the values of  $\xi$  for which these parametric systems are passive or strictly passive. The following theorem, which is a combination of [22, Theorem 4.5 and Lemma 6.2], classifies these values of  $\xi$ . Note that strict passivity of  $\mathcal{M}$  implies regularity of the rational matrix function  $\Phi$ , since  $\Phi(\infty)$  is invertible. Thus,  $\Phi_\xi$  is also guaranteed to be regular for almost all  $\xi$ , since  $\Phi_\xi(\infty)$  is invertible for almost all  $\xi$ .

**THEOREM 4.3.** *Let  $\mathcal{M} := \{A, B, C, D\}$  be a continuous-time minimal system model and let its transfer function  $\mathcal{T}$  thus be proper. Then, for any  $\xi \in \mathbb{R}$ , the parametric system  $\mathcal{M}_\xi$  with transfer function  $\mathcal{T}_\xi$ , as defined in (4.2), is also minimal and*

$$(4.3) \quad \Xi := \sup_{-\infty < \xi < \infty} \{\xi : \mathcal{M}_\xi \text{ is strictly passive}\} = \max_{-\infty < \xi < \infty} \{\xi : \mathcal{M}_\xi \text{ is passive}\}$$

*is bounded. Moreover, the system  $\mathcal{M}_\xi$  is strictly passive for  $\xi \in (-\infty, \Xi)$ , passive but not strictly passive for  $\xi = \Xi$ , and nonpassive for  $\xi \in (\Xi, +\infty)$ .*

*Proof.* It is obvious that adding  $\frac{\xi}{2}I$  to the matrix  $A$  does not affect controllability or observability since it is a mere shift of the variable  $\lambda$  in the corresponding rank conditions. The rest of the proof is based on the fact that passivity of  $\mathcal{M}_\xi$  is linked to the inequality  $W_c(X, \mathcal{M}) \succeq \xi \text{diag}(X, I_m)$  for some  $X \succ 0$ , and that strict passivity of  $\mathcal{M}_\xi$  is linked to the strict inequality  $W_c(X, \mathcal{M}) \succ \xi \text{diag}(X, I_m)$  for some  $X \succ 0$ . Consequently, for all  $\tilde{\xi} < \xi$ ,  $\mathcal{M}_\xi$  being passive implies that  $\mathcal{M}_{\tilde{\xi}}$  is strictly passive, and if  $\mathcal{M}_\xi$  is strictly passive, then  $\mathcal{M}_{\tilde{\xi}}$  is also strictly passive in an open neighborhood about  $\xi$ . This proves that the interval for  $\xi$  corresponding to strictly passive systems and nonpassive systems are both open and connected, and that there is a single boundary point  $\Xi$  that must be passive, but not strictly passive. The boundedness of  $\Xi$  follows from the minimality of the realization for  $\mathcal{M}_\xi$ . We refer to [22] for the details.  $\square$

By computing  $\Xi$ , we can ascertain whether  $\mathcal{M}_\xi$  is (strictly) passive or nonpassive for all  $\xi \in \mathbb{R}$ . For a given value of  $\xi$ , by Definition 4.1,  $\mathcal{M}_\xi$  corresponds to a strictly passive system if and only if  $\Phi_\xi(i\omega) \succ 0$  for all  $\omega \in \mathbb{R} \cup \{\infty\}$ , and  $\alpha(A_\xi) < 0$ . Checking asymptotic stability is done via computing  $\alpha(A_\xi)$ . When  $n$  is not too large, the entire spectrum of  $A_\xi$  can be computed to obtain  $\alpha(A_\xi)$ , while for large  $n$ , one can use, e.g., `eigs` in MATLAB, to efficiently compute a globally rightmost eigenvalue of  $A_\xi$ . Checking the positive definiteness condition is more involved. For  $\xi, \omega \in \mathbb{R}$ , consider

$$(4.4) \quad \gamma(\xi, \omega) := \lambda_{\min}(\Phi_\xi(i\omega)) \quad \text{and} \quad \begin{aligned} \gamma_\xi(\omega) &:= \gamma(\xi, \omega), & \text{where } \xi \in \mathbb{R} \text{ is fixed,} \\ \gamma_\omega(\xi) &:= \gamma(\xi, \omega), & \text{where } \omega \in \mathbb{R} \text{ is fixed.} \end{aligned}$$

Clearly,  $\Phi_\xi(i\omega) \succ 0$  if and only if  $\gamma_\xi(\omega) > 0$ , and at  $\omega = \infty$ , this is simply equivalent to  $D_\xi^H + D_\xi \succ 0$  with  $\lim_{\omega \rightarrow \infty} \gamma_\xi(\omega) = \lambda_{\min}(D_\xi^H + D_\xi) > 0$ . If  $\alpha(A_\xi) < 0$ , then  $\mathcal{T}_\xi$  has no

poles on the imaginary axis, and so neither does  $\Phi_\xi$ , hence,  $\gamma_\xi$  is a continuous function. Thus, if  $D_\xi^H + D_\xi \succ 0$  and  $\alpha(A_\xi) < 0$ , then  $\gamma_\xi(\omega_1) \leq 0$  if and only if  $\det \Phi_\xi(i\omega_2) = 0$  for  $\omega_1, \omega_2 \in \mathbb{R}$  with  $\omega_1 = \omega_2$  not necessarily holding. Summarizing, we have the following necessary and sufficient algebraic continuous-time conditions for the strict passivity of  $\mathcal{M}_\xi$ :

- (C1)  $\alpha(A_\xi) < 0$  (asymptotic stability of  $A_\xi = A + \frac{\xi}{2}I_n$ );
- (C2)  $D_\xi^H + D_\xi = D^H + D - \xi I_m \succ 0$  (positive definiteness at  $\omega = \infty$ );
- (C3)  $\det \Phi_\xi(i\omega) \neq 0$  for all  $\omega \in \mathbb{R}$  (implying positive definiteness for all finite  $\omega$  provided that (C1) and (C2) also hold).

A bracket containing  $\Xi$  can be easily computed. A simple lower bound on  $\Xi$  is

$$(4.5) \quad \Xi_{\text{lb}} := \lambda_{\min}(W_c(I_n, \mathcal{M})),$$

as clearly

$$W_c(I_n, \mathcal{M}_{\Xi_{\text{lb}}}) = W_c(I_n, \mathcal{M}) - \Xi_{\text{lb}} I_{n+m} \succeq 0$$

holds, and so by Theorems 4.2 and 4.3,  $\mathcal{M}_{\Xi_{\text{lb}}}$  is passive. Meanwhile, (C1) and (C2) will no longer be satisfied if  $\xi$  is too large: (C1) holds if and only if  $\xi < -2\alpha(A)$ , and (C2) holds if and only if  $\xi < \lambda_{\min}(D^H + D)$ . Thus, a simple upper bound for  $\Xi$  is

$$(4.6) \quad \Xi_{\text{ub}} := \min \{ -2\alpha(A), \lambda_{\min}(D^H + D) \}.$$

Let us now look at  $\xi \in [\Xi_{\text{lb}}, \Xi_{\text{ub}})$ , where for this half-open interval,  $\alpha(A_\xi) < 0$  and  $D_\xi^H + D_\xi \succ 0$ . Therefore, in order to verify the strict passivity of  $\mathcal{M}_\xi$ , one only needs to verify condition (C3), i.e., that  $\det \Phi_\xi(i\omega) \neq 0$  for all  $\omega \in \mathbb{R}$ . This condition can be checked via the following result for  $\mathcal{M}_\xi$ , which is well known in the literature for general systems  $\mathcal{M}$  (see, e.g., [20]).

**THEOREM 4.4.** *Let  $\xi \in \mathbb{R}$ ,  $\mathcal{M}_\xi$  and  $\Phi_\xi$  be as defined in (4.2), and  $\omega \in \mathbb{C}$  (not only  $\mathbb{R}$ ) be any point such that  $i\omega \notin \Lambda(A_\xi)$ . Then  $\det \Phi_\xi(i\omega) = 0$  if and only if  $\det(M_\xi - \omega N) = 0$ , where the regular Hermitian pencil  $M_\xi - \lambda N$  is defined by*

$$(4.7) \quad M_\xi := \begin{bmatrix} 0 & A_\xi & B \\ A_\xi^H & 0 & C^H \\ B^H & C & D_\xi^H + D_\xi \end{bmatrix} \quad \text{and} \quad N := \begin{bmatrix} 0 & iI_n & 0 \\ -iI_n & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Furthermore, if  $D_\xi^H + D_\xi$  is nonsingular, then  $\det \Phi_\xi(i\omega) = 0$  if and only if  $\det(H_\xi - i\omega I_{2n}) = 0$ , where  $H_\xi$  is the Hamiltonian matrix

$$(4.8) \quad H_\xi := \begin{bmatrix} A_\xi & 0 \\ 0 & -A_\xi^H \end{bmatrix} - \begin{bmatrix} B \\ C^H \end{bmatrix} (D_\xi^H + D_\xi)^{-1} \begin{bmatrix} C & -B^H \end{bmatrix}.$$

*Proof.* Writing

$$M_\xi - \omega N := \begin{bmatrix} 0 & A_\xi - i\omega I_n & B \\ A_\xi^H + i\omega I_n & 0 & C^H \\ B^H & C & D_\xi^H + D_\xi \end{bmatrix},$$

and using the Schur identity of determinants with respect to the leading  $2n \times 2n$  block, which by assumption is nonsingular, we obtain that

$$\det(M_\xi - \omega N) = \det \begin{bmatrix} 0 & A_\xi - i\omega I_n \\ A_\xi^H + i\omega I_n & 0 \end{bmatrix} \det \Phi_\xi(i\omega).$$

As  $\mathbf{i}\omega \notin \Lambda(A_\xi)$ , the first equivalence involving  $M_\xi - \lambda N$  holds, and since  $\Phi_\xi(\mathbf{i}\omega)$  is regular, this matrix pencil must be a regular one. To obtain the second equivalence, we consider  $(H_\xi - \mathbf{i}\omega I_{2n}) \begin{bmatrix} 0 & -I_n \\ I_n & 0 \end{bmatrix}$ , which is the Schur complement of  $M_\xi - \omega N$  with respect to the trailing  $m \times m$  block and requires the additional assumption that  $D_\xi^H + D_\xi$  is nonsingular. Then, via the Schur identity of determinants, we have that

$$\det(M_\xi - \omega N) = \det(D_\xi^H + D_\xi) \det \left( (H_\xi - \mathbf{i}\omega I_{2n}) \begin{bmatrix} 0 & -I_n \\ I_n & 0 \end{bmatrix} \right). \quad \square$$

**COROLLARY 4.5.** *Let  $\Xi_{\text{lb}}$  and  $\Xi_{\text{ub}}$  be as defined in (4.5) and (4.6), respectively, and let  $\xi \in [\Xi_{\text{lb}}, \Xi_{\text{ub}}]$ . Then the function  $\gamma_\xi : \mathbb{R} \rightarrow \mathbb{R}$  defined in (4.4) has at most  $2n$  zeros, all of which must be finite.*

*Proof.* If  $\gamma_\xi(\omega) = 0$ , then  $\det \Phi_\xi(\mathbf{i}\omega) = 0$ . Since  $\xi \in [\Xi_{\text{lb}}, \Xi_{\text{ub}}]$ , (C1) and (C2) both hold, and so the assumptions of Theorem 4.4 are met. Hence,  $\det \Phi_\xi(\mathbf{i}\omega) = 0$  if and only if  $\det(H_\xi - \mathbf{i}\omega I_{2n}) = 0$ . Finally, as  $H_\xi \in \mathbb{C}^{2n \times 2n}$ , it has  $2n$  (finite) eigenvalues.  $\square$

Given the bracket  $[\Xi_{\text{lb}}, \Xi_{\text{ub}}]$ , Theorem 4.4 immediately leads to a bisection method for computing  $\Xi$  [22, p. 144]. For any  $\xi \in [\Xi_{\text{lb}}, \Xi_{\text{ub}}]$ , (C3) can be verified by computing the eigenvalues of either  $M_\xi - \lambda N$  or  $H_\xi$  (the pencil form is preferred numerically, since it only has a linear dependence on  $\xi$ ). Via the following result, Mehrmann and Van Dooren also proposed a second improved algorithm for computing  $\Xi$  [22, p. 146].

**THEOREM 4.6.** *Let  $\gamma : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  and  $\gamma_\xi : \mathbb{R} \rightarrow \mathbb{R}$  be as defined in (4.4) and  $\Xi_{\text{lb}}$  and  $\Xi_{\text{ub}}$  be as defined in (4.5) and (4.6), respectively. Then  $\gamma$  is continuous on the domain  $[\Xi_{\text{lb}}, \Xi_{\text{ub}}] \times \mathbb{R}$ , and  $\gamma_\xi$  has the following properties:*

- (i) *if  $\xi \in [\Xi_{\text{lb}}, \Xi]$ , then  $\min_{\omega \in \mathbb{R}} \gamma_\xi(\omega) > 0$ ;*
- (ii) *if  $\xi = \Xi$ , then  $\min_{\omega \in \mathbb{R}} \gamma_\xi(\omega) = 0$ ;*
- (iii) *if  $\xi \in (\Xi, \Xi_{\text{ub}}]$  with  $\Xi < \Xi_{\text{ub}}$ , then  $\gamma_\xi(\omega) < 0 \forall \omega \in \mathcal{S} \subset \mathbb{R}$ , where  $\mathcal{S}$  is the union of a finite number of nonoverlapping open bounded intervals.*

*Proof.* This was proven in [22, Theorem 5.1] except for the claim in (iii) that the intervals are bounded, which follows directly from Corollary 4.5.  $\square$

Mehrmann and Van Dooren's improved method computes a monotonically decreasing sequence  $\{\xi_k\} \rightarrow \Xi$ , where the initial estimate is  $\xi_0 = \Xi_{\text{ub}} - \tau$  for some small tolerance  $\tau > 0$ . On the  $k$ th iteration, via Theorem 4.4 and computing the eigenvalues of  $M_{\xi_k} - \lambda N$ , the bounded intervals where  $\gamma_{\xi_k}$  is negative are obtained. Taking  $\hat{\omega}$  to be the midpoint of the largest of these intervals with  $\gamma_{\xi_k}(\hat{\omega}) < 0$  holding,  $\xi_{k+1}$  is obtained by setting it to the smallest value of  $\xi$  such that  $\gamma_{\hat{\omega}}(\xi) = 0$ ; this is done by computing all the eigenvalues of a matrix pencil closely related to  $M_{\xi_k} - \lambda N$  (and of the same order). This process is continued in a loop until convergence to  $\Xi$ ; see [22, section 5] for more details.

#### 4.2. An HEC-based algorithm for computing the continuous-time $\Xi$ .

By Theorem 4.6,  $\Xi$  defined in (4.3) can instead be computed via this root-min problem

$$(4.9) \quad \text{determine the } \xi \in \mathcal{D}_1 : \quad f(\xi) = \min_{\omega \in \mathcal{D}_2} \gamma(\xi, \omega) = \min_{\omega \in \mathcal{D}_2} \gamma_\xi(\omega) = 0,$$

where  $\mathcal{D}_1 = [\Xi_{\text{lb}}, \Xi_{\text{ub}}]$ ,  $\mathcal{D}_2 \subset \mathbb{R}$ , and the function  $\gamma$  and its associated restriction  $\gamma_\xi$  are defined in (4.4). If  $\xi \neq \Xi_{\text{ub}}$ , by Corollary 4.5 and Theorem 4.6,  $\mathcal{D}_2$  can be taken to be compact, since for all other values of  $\xi \in \mathcal{D}_1$ ,  $\gamma_\xi$  has at most  $2n$  zeros, which are all

finite, and minimizers of  $\gamma_\xi$  that occur where  $\gamma_\xi$  is negative clearly must lie between these zeros. Since  $\alpha(A_\xi) < 0$  for all  $\xi < \Xi_{\text{ub}}$ , it follows that  $\gamma_\xi$  is also bounded below (and above) for all  $\xi < \Xi_{\text{ub}}$ . Thus, (4.9) meets the criteria to be a valid root-min problem, and so we can use HEC to find pseudoroots of it.

*Remark 4.7.* For a root-min problem, the initialization requirements for HEC, using the notation of (3.2) and Algorithm 3.1, are  $\varepsilon_{\text{lb}}, \varepsilon_0 \in \mathcal{D}_1$  and  $x_0 \in \mathcal{D}_2$  such that  $f(\varepsilon_0) \leq g(\varepsilon_0, x_0) \leq 0 < f(\varepsilon_{\text{lb}})$ . For computing  $\Xi$ , it will also be more convenient to use the convention that  $f(\varepsilon_0) \leq g(\varepsilon_0, x_0) < 0 \leq f(\varepsilon_{\text{lb}})$  holds.

*Remark 4.8.* We wish to compute  $\Xi$  to a desired relative accuracy determined by a relative error tolerance  $\tau \in (0, 1)$ . However, if  $\Xi = 0$  happens to hold, the relative error is undefined. In this case, our algorithm instead automatically estimates  $\Xi$  to within an absolute error determined by  $\tau$ . In practice, one could use two parameters to respectively determine acceptable accuracy in relative and absolute senses. For the sake of simplicity, we assume that  $\Xi \neq 0$  in order to avoid having to refer to both relative and absolute errors.

First note that  $0 \leq f(\Xi_{\text{lb}})$  always holds. If we have a  $\xi_0 \in [\Xi_{\text{lb}}, \Xi_{\text{ub}})$  and  $\omega_0 \in \mathbb{R}$  such that  $\gamma(\xi_0, \omega_0) < 0$ , then the initialization conditions of HEC are met, and so HEC can be used to compute a pseudoroot  $(\tilde{\xi}, \tilde{\omega})$  of (4.9) with both  $\tilde{\xi} \in [\Xi_{\text{lb}}, \xi_0)$  and  $\Xi \leq \tilde{\xi}$  holding. To determine whether the estimate  $\tilde{\xi}$  is sufficiently close to  $\Xi$ , we do the following. For our tolerance  $\tau$ , we set  $\xi = \tilde{\xi} - \tau|\tilde{\xi}|$  (we assume that  $\Xi_{\text{lb}} < \xi$ , as otherwise we are done) and then compute the real eigenvalues of  $M_\xi - \lambda N$ . If this matrix pencil has no real eigenvalues, then by Theorem 4.4, we know that  $\gamma_\xi$  has no zeros, and so by Theorem 4.6,  $\xi < \Xi$  must hold. Thus,  $\Xi \in (\tilde{\xi}, \xi]$ , and so  $\xi$  must agree with  $\Xi$  to the desired number of digits.<sup>3</sup> Otherwise, if  $M_\xi - \lambda N$  does have real eigenvalues, then  $\gamma_\xi$  has zeros, and if  $\gamma_\xi$  is negative on at least one of the intervals derived from these zeros, then  $\Xi < \xi$  holds by Theorem 4.6. Updating  $\omega_0$  to be the midpoint of one of these intervals where  $\gamma_\xi$  is negative, obviously  $\gamma_\xi(\omega_0) < 0$  holds, and so HEC can be restarted to find a new pseudoroot  $(\hat{\xi}, \hat{\omega})$  of (4.9) with  $\hat{\xi} \in [\Xi, \xi)$ . This process of running HEC and computing the real eigenvalues of  $M_\xi - \lambda N$  to find regions where  $\gamma_\xi$  is negative is repeated in a loop until estimate  $\xi$ , which is decreasing monotonically, becomes sufficiently close to  $\Xi$ .

For initializing our new algorithm, it is required that we always choose  $\xi < \Xi_{\text{ub}}$ ; e.g., evaluating  $\mathcal{T}_\xi(i\omega)$  requires solving linear systems with  $i\omega I - A_\xi$ , but this matrix may not always be invertible when  $\xi = \Xi_{\text{ub}}$ . Choosing  $\xi_0 = \Xi_{\text{ub}} - \tau|\Xi_{\text{ub}}|$  as our first estimate suffices, as this still allows us to obtain  $\Xi$  to the desired accuracy. Again, we assume that  $\Xi_{\text{lb}} < \xi_0$ , as otherwise we are done. The user provides some  $\omega_0 \in \mathbb{R}$  as an initial guess for HEC. If  $\gamma_{\xi_0}(\omega_0) < 0$  holds, then our algorithm as described above can begin. Otherwise, we must find another point where  $\gamma_{\xi_0}$  is negative. This can be done in multiple ways. We can evaluate  $\gamma_{\xi_0}$  on a grid or at randomly chosen points. We could also initialize some optimization solver at these points to try to find a minimizer  $\tilde{\omega}$  of  $\gamma_{\xi_0}$  such that  $\gamma_{\xi_0}(\tilde{\omega}) < 0$ . If, after some reasonable amount of effort, such a point has not been found, we then resort to computing the eigenvalues of  $M_{\xi_0} - \lambda N$  in order to obtain all the zeros of  $\gamma_{\xi_0}$ . Then, as described above, we can determine if there exists a point where  $\gamma_{\xi_0}$  is negative. Since evaluating  $\gamma_{\xi_0}(\omega)$  for a given value of  $\omega$  is much cheaper than computing the eigenvalues of  $M_{\xi_0} - \lambda N$  (we elaborate on this momentarily), it is generally beneficial in terms of the overall runtime to first

<sup>3</sup>Note that when  $\tilde{\xi}$  is only sufficiently close to  $\Xi$  (and not equal), it does not necessarily follow that  $\tilde{\omega}$  is close to the global minimizer(s) of  $\gamma_\Xi$ .

---

**Algorithm 4.1** HEC-based algorithm for continuous-time  $\Xi$ .

---

**Input:**  $\mathcal{M}$ ,  $\omega_0 \in \mathbb{R}$ ,  $\tau \in (0, 1)$ ,  $\Xi_{\text{lb}}$  (4.5), and  $\Xi_{\text{ub}}$  (4.6)

**Output:**  $\xi$  such that  $|\Xi - \xi| \leq \tau|\Xi|$  for continuous-time  $\Xi$  for  $\mathcal{M}$

```

1:  $\xi \leftarrow \Xi_{\text{ub}} - \tau|\Xi_{\text{ub}}|$ 
2: if  $\xi \leq \Xi_{\text{lb}}$  then
3:   return
4: end if
5: find_negative  $\leftarrow \gamma_{\xi}(\omega_0) \geq 0$  // a boolean variable
6: while true do
7:   if find_negative then
8:      $\Omega \leftarrow \{\omega \in \mathbb{R} : \det(M_{\xi} - \omega N) = 0\}$ 
9:      $\Omega \leftarrow \{\omega \in \Omega : \gamma_{\xi}(\omega) = 0\}$ 
10:    if  $\exists \omega_1, \omega_2 \in \Omega$  s.t.  $\omega_1 < \omega_2$  and  $\gamma_{\xi}(\omega) < 0 \forall \omega \in (\omega_1, \omega_2)$  then
11:       $\omega_0 \leftarrow 0.5(\omega_1 + \omega_2)$  // (C3) does not hold
12:    else
13:      return //  $\gamma_{\xi}(\omega) \geq 0 \forall \omega \in \mathbb{R}$  and  $\xi \approx \Xi$  to tolerance
14:    end if
15:  end if
16:  //  $\gamma_{\xi}(\omega_0) < 0$  and  $\Xi \in [\Xi_{\text{lb}}, \xi)$  so run HEC with this initial data
17:   $(\tilde{\xi}, \tilde{\omega}) \leftarrow$  a pseudoroot of (4.9) obtained by HEC with  $\Xi \leq \tilde{\xi} < \xi$ 
18:   $\xi \leftarrow \tilde{\xi} - \tau|\tilde{\xi}|$ 
19:  find_negative  $\leftarrow$  true
20: end while

```

NOTE: Per Remark 4.8, we assume that  $\Xi \neq 0$ , and so at termination,  $\xi$  will agree with  $\Xi$  to the desired relative tolerance  $\tau$ . When the matrices defining  $\mathcal{M}$  are all real, there is symmetry, i.e.,  $\gamma(\xi, -\omega) = \gamma(\xi, \omega)$ , and so the search domain for  $\omega$  can be reduced from  $\mathbb{R}$  to  $\omega \in [0, \infty)$ . While taking advantage of this symmetry does not affect the asymptotic work complexity, it can nevertheless reduce the constant factors to speed up the overall run time.

try a decent number of points, possibly with optimization. This also increases the chances that the first pseudoroot  $(\tilde{\xi}, \tilde{\omega})$  of (4.9) found by HEC also provides its root, i.e.,  $\tilde{\xi} = \Xi$ ; in this case, our algorithm only computes the eigenvalues of  $M_{\xi} - \lambda N$  for a single value of  $\xi$ . In contrast, recall that the earlier algorithm of Mehrmann and Van Dooren (described at the end of subsection 4.1), on every iteration, requires computing the eigenvalues of  $M_{\xi} - \lambda N$  plus the eigenvalues of a second related matrix pencil with the same order  $(2n + m)$ .

Pseudocode of our new algorithm for continuous-time  $\Xi$  is given in Algorithm 4.1. In practice, we observe that HEC is only restarted a handful of times, often just once. By construction of the valid data to initialize HEC on every iteration of Algorithm 4.1, it follows from Theorem 3.7 that HEC is indeed guaranteed to compute a pseudoroot of (4.9) on every iteration. Under mild assumptions that generally hold in practice, we now show that Algorithm 4.1 has local quadratic convergence to pseudoroots of (4.9).

**THEOREM 4.9** (quadratic convergence of Algorithm 4.1). *Let  $(\tilde{\xi}, \tilde{\omega})$  be any pseudoroot of (4.9) computed by HEC within Algorithm 4.1. If*

- (i) *after some point, HEC only generates iterates that lie on a single open continuous path  $\omega_p : \mathbb{R} \rightarrow \mathbb{R}$  of stationary points of  $\gamma_{\xi}$  as  $\xi$  varies,*
- (ii)  *$\tilde{\omega} = \omega_p(\tilde{\xi})$  is a local maximizer of  $\gamma_{\tilde{\xi}}$ ,*



- (iii)  $\gamma'_\omega(\tilde{\xi}) \neq 0$ ,
- (iv)  $\gamma$  is twice continuously differentiable at  $(\tilde{\xi}, \tilde{\omega})$ , and
- (v)  $\omega_p$  is twice continuously differentiable at  $\tilde{\xi}$ ,

all hold, then Algorithm 4.1 converges  $Q$ -quadratically to the pseudoroot  $(\tilde{\xi}, \tilde{\omega})$ . Furthermore, if (i)–(iii) hold and  $\gamma(\tilde{\xi}, \tilde{\omega})$  corresponds to a simple eigenvalue of  $\Phi_\xi(\mathbf{i}\tilde{\omega})$  and  $\gamma''_\xi(\tilde{\omega}) \neq 0$ , then (iv) and (v) are automatically satisfied.

*Proof.* Conditions (i)–(v) implying the quadratic convergence of HEC is simply a translation of Theorem 3.10 to the setting of (4.9). For the second part of the theorem, if  $\gamma(\tilde{\xi}, \tilde{\omega})$  corresponds to a simple eigenvalue, then  $\gamma$  is analytic near  $(\tilde{\xi}, \tilde{\omega})$ . Defining the function  $g(\xi, \omega) := \frac{\partial}{\partial \omega} \gamma(\xi, \omega)$ , the path  $\omega_p$  of stationary points of the functions  $\gamma_\xi$  as  $\xi$  varies can be characterized by the equality  $g(\xi, \omega) = 0$  in a neighborhood about the pseudoroot  $(\tilde{\xi}, \tilde{\omega})$ . Since  $g$  is analytic, if  $\frac{\partial g}{\partial \omega}|_{(\tilde{\xi}, \tilde{\omega})} \neq 0$ , it follows from the implicit function theorem that we can rewrite our characterization as  $g(\xi, \omega_p(\xi)) = 0$ , where  $\omega_p$  is analytic near  $\tilde{\xi}$ .  $\square$

Under the smoothness assumptions of Theorem 4.9, in a neighborhood of the pseudoroot in question, the contraction and expansion phases within HEC can also be solved with fast convergence rates. Moreover, even if these assumptions do not hold, an extension of the analysis of Boyd and Balakrishnan [5] shows that near any minimizer  $\omega$ ,  $\gamma_\xi$  is twice continuously differentiable with a Lipschitz second derivative, even if the minimizer is associated with an eigenvalue of  $\Phi_\xi(\mathbf{i}\omega)$  of multiplicity greater than one; for more details, see [25]. Thus, the expansion phases can always be solved quickly using secant or Newton’s method, and any use of the early contraction technique discussed in Remark 3.11 should be limited, e.g., only in initial iterations when one cannot necessarily expect to be sufficiently close to the fast convergence regime.

For the contraction and expansion phases, we now describe how to compute the first and second derivatives of the functions  $\gamma_\xi$  and  $\gamma_\omega$  defined in (4.4). Given a simple eigenvalue of a Hermitian matrix  $H$  depending on a parameter  $t \in \mathbb{R}$ , formulas for the first and second derivatives of that eigenvalue can be found in, e.g., [18, 28]. The matrix derivatives  $H'$  and  $H''$  appear in these formulas, so we give below the first and second matrix derivatives of both  $\mathcal{T}_\xi(\mathbf{i}\omega)$  and  $\Phi_\xi(\mathbf{i}\omega)$  with respect to  $\xi$  and  $\omega$ . Via applications of standard matrix derivative formulas and the chain rule, we have that

$$(4.10a) \quad \frac{\partial}{\partial \xi} \mathcal{T}_\xi(\mathbf{i}\omega) = \frac{1}{2}(Z_2 - I_m), \quad \frac{\partial^2}{\partial \xi^2} \mathcal{T}_\xi(\mathbf{i}\omega) = \frac{1}{2}Z_3,$$

$$(4.10b) \quad \frac{\partial}{\partial \omega} \mathcal{T}_\xi(\mathbf{i}\omega) = -\mathbf{i}Z_2, \quad \frac{\partial^2}{\partial \omega^2} \mathcal{T}_\xi(\mathbf{i}\omega) = -2Z_3,$$

where  $Z_k := C((\mathbf{i}\omega - \frac{\xi}{2})I_n - A)^{-k}B$ , and so

$$(4.11a) \quad \frac{\partial}{\partial \xi} \Phi_\xi(\mathbf{i}\omega) = \frac{1}{2}(Z_2 + Z_2^H) - I_m, \quad \frac{\partial^2}{\partial \xi^2} \Phi_\xi(\mathbf{i}\omega) = \frac{1}{2}(Z_3 + Z_3^H),$$

$$(4.11b) \quad \frac{\partial}{\partial \omega} \Phi_\xi(\mathbf{i}\omega) = -\mathbf{i}Z_2 + \mathbf{i}Z_2^H, \quad \frac{\partial^2}{\partial \omega^2} \Phi_\xi(\mathbf{i}\omega) = -2(Z_3 + Z_3^H).$$

Following a technique of Laub [19], we can compute the Hessenberg form  $A = UHU^H$ , where  $U$  is unitary and  $H$  is Hessenberg, and then substitute it into  $Z_1$ , which yields  $Z_1 = CU((\mathbf{i}\omega - \frac{\xi}{2})I_n - H)^{-1}U^HB$ . Computing  $U$  and  $H$  is  $\mathcal{O}(n^3)$  work but only needs to be done once. Thereafter, the inverses appearing in  $Z_k$  can actually be applied to a vector in just  $\mathcal{O}(n^2)$  work since changing the values of  $\xi$  and  $\omega$  cannot cause the Hessenberg structure to be lost. Thus,  $\Phi_\xi(\mathbf{i}\omega)$  and its matrix derivatives given in (4.11) can be obtained in  $\mathcal{O}(mn^2 + m^2n)$  work. Using the convention that

computing the eigenvalues and eigenvectors of a matrix is an atomic operation with cubic complexity, the total cost to evaluate  $\gamma(\xi, \omega)$  and its first and second derivatives with respect to  $\xi$  and  $\omega$  is  $\mathcal{O}(mn^2 + m^2n + m^3)$  work.

The cost of Algorithm 4.1 is dominated by computing the zeros of  $\gamma_\xi$ . Since HEC generally converges quickly, as do its expansion and contraction phases, we can consider that the total number of evaluations of the function  $\gamma$  is bounded by a constant. Hence, in Algorithm 4.1, HEC does  $\mathcal{O}(mn^2 + m^2n + m^3)$  work. Meanwhile, finding the zeros of  $\gamma_\xi$  involves computing all the eigenvalues of  $M_\xi - \lambda N$ , which itself is  $\mathcal{O}((n+m)^3)$  work. Thus, for all but the smallest values of  $n$ , the HEC portion of Algorithm 4.1 should only be a fraction of the cost to compute the eigenvalues of  $M_\xi - \lambda N$ .

The “improved” algorithm of Mehrmann and Van Dooren has the same asymptotic work complexity as our method, but the hidden constant factor for their algorithm is much larger. This is partly because on each iteration, their algorithm solves two large eigenvalue problems of order  $2n + m$ . However, it also often requires more iterations than Algorithm 4.1 does. While Mehrmann and Van Dooren did not analyze the convergence properties of their method, our new framework of root-max problems and HEC also shows that their method converges at least Q-superlinearly under generic conditions. To see this, note that on each iteration, their method computes a single point where  $\gamma_\xi$  is negative (as opposed to finding a local minimizer), but in the limit, these single points do converge to a minimizer of  $\gamma_\xi$  as  $\xi \rightarrow \Xi$ . In other words, their algorithm can also be seen as an HEC iteration with a very aggressive early contraction scheme. Per Remark 3.11, such an iteration converges at least superlinearly.

**5. Discrete-time passive systems.** We now present the discrete-time analogues of the optimization problem and our new algorithm given in section 4. We reuse the continuous-time notation from section 4. for the discrete-time setting since the different meanings should be clear from the context, and it allows us to generically refer to either setting when needed.

**5.1. Passivity of discrete-time proper parametric systems.** The material in this subsection closely follows [21]. For  $z \in \mathbb{C}$ , we now consider the rational matrix function

$$\Phi(z) := \mathcal{T}^H(z^{-1}) + \mathcal{T}(z),$$

which coincides with twice the Hermitian part of  $\mathcal{T}$  on the unit circle:

$$\Phi(e^{i\omega}) = [\mathcal{T}(e^{i\omega})]^H + \mathcal{T}(e^{i\omega}).$$

DEFINITION 5.1. *The discrete-time system  $\mathcal{M}$  with transfer function  $\mathcal{T}$  is*

1. *passive if  $\Phi(e^{i\omega}) \succeq 0$  for all  $\omega \in (-\pi, \pi]$  and  $\rho(A) \leq 1$ , i.e., its eigenvalues are in the closed unit disk, with any occurring on the unit circle being semi-simple and with a transfer function residue that is Hermitian and positive semidefinite;*
2. *strictly passive if  $\Phi(e^{i\omega}) \succ 0$  for all  $\omega \in (-\pi, \pi]$  and  $\rho(A) < 1$ .*

The necessary and sufficient conditions for passivity in the discrete-time case (see [21]) now make use of the linear matrix function

$$(5.1) \quad W_d(X, \mathcal{M}) := \begin{bmatrix} X & XA & XB \\ A^H X & X & C^H \\ B^H X & C & D^H + D \end{bmatrix}.$$

THEOREM 5.2. *Let  $\mathcal{M} := \{A, B, C, D\}$  be a discrete-time minimal system and let its transfer function  $\mathcal{T}$  thus be proper. Then  $\mathcal{M}$  is (strictly) passive if and only there exists an  $X \in \mathbb{H}_n$  such that  $X \succ 0$  and  $W_d(X, \mathcal{M}) \succeq 0$  ( $W_d(X, \mathcal{M}) \succ 0$ ).*

In [21], the following class of parametric systems was considered:

$$(5.2a) \quad \mathcal{M}_\xi := \{A_\xi, B_\xi, C_\xi, D_\xi\} = \left\{ \frac{A}{1-\xi}, \frac{B}{1-\xi}, \frac{C}{1-\xi}, \frac{D-\xi I_m}{1-\xi} \right\},$$

$$(5.2b) \quad \mathcal{T}_\xi(z) := C_\xi(zI_n - A_\xi)^{-1}B_\xi + D_\xi = \frac{1}{1-\xi} \left( C((1-\xi)zI_n - A)^{-1}B + D - \xi I_m \right),$$

$$(5.2c) \quad \Phi_\xi(z) := \mathcal{T}_\xi^H(z^{-1}) + \mathcal{T}_\xi(z),$$

where  $\xi \in (-\infty, 1)$  and it is again important to compute for which values of  $\xi$  these systems are passive or strictly passive. Similarly to the continuous-time case, it was shown in [21] that the smallest perturbation  $\Delta_{\mathcal{M}}$  such that the perturbed model  $\mathcal{M} + \Delta_{\mathcal{M}}$  loses passivity depends on this scalar parameter  $\xi$ . It is therefore important to compute the values of  $\xi$  for which these parametric systems are passive or strictly passive. The following theorem was given in [21], in a slightly modified form; we omit its proof as it is similar to that of Theorem 4.3. Note that strict passivity of  $\mathcal{M}$  again implies regularity of the rational matrix function  $\Phi$ , since it is invertible for any point on the unit circle. Moreover,  $\Phi_\xi$  is then also regular for almost all  $\xi$  since it is an analytic perturbation of  $\Phi$ .

THEOREM 5.3. *Let  $\mathcal{M} := \{A, B, C, D\}$  be a discrete-time minimal system  $\mathcal{M}$  and let its transfer function  $\mathcal{T}$  thus be proper. Then, for  $\xi \in (-\infty, 1)$ , the parametric system  $\mathcal{M}_\xi$  with transfer function  $\mathcal{T}_\xi$ , as defined in (5.2), is also minimal and*

$$(5.3) \quad \Xi := \sup_{-\infty < \xi < 1} \{ \xi : \mathcal{M}_\xi \text{ is strictly passive} \} = \max_{-\infty < \xi < 1} \{ \xi : \mathcal{M}_\xi \text{ is passive} \}$$

*is bounded. Moreover,  $\mathcal{M}_\xi$  is strictly passive for  $\xi \in (-\infty, \Xi)$ , passive but not strictly passive for  $\xi = \Xi$ , and not passive for  $\xi \in (\Xi, 1)$ .*

By Definition 5.1,  $\mathcal{M}_\xi$  is strictly passive if and only if  $\Phi_\xi(z) \succ 0$  holds over the entire unit circle and  $\rho(A_\xi) < 1$ . Obtaining the value of  $\rho(A_\xi)$  to check asymptotic stability can be done by computing an outermost eigenvalue of  $A_\xi$  via, e.g., `eig` or `eigs` in MATLAB. Checking the discrete-time positive definiteness condition is a little more subtle than it is in the continuous-time case. For  $\xi, \omega \in \mathbb{R}$ , consider

$$(5.4) \quad \gamma(\xi, \omega) := \lambda_{\min}(\Phi_\xi(e^{i\omega})) \quad \text{and} \quad \begin{aligned} \gamma_\xi(\omega) &:= \gamma(\xi, \omega) \quad \text{where } \xi \in \mathbb{R} \text{ is fixed,} \\ \gamma_\omega(\xi) &:= \gamma(\xi, \omega) \quad \text{where } \omega \in \mathbb{R} \text{ is fixed,} \end{aligned}$$

where  $\Phi_\xi$  is defined in (5.2). Clearly  $\Phi_\xi(e^{i\omega}) \succ 0$  is equivalent to  $\gamma_\xi(\omega) > 0$ , and  $\gamma_\xi$  is continuous if  $\rho(A_\xi) < 1$ , as then  $\Phi_\xi$  cannot have any poles on the unit circle. Hence, if  $\rho(A_\xi) < 1$  and  $\Phi_\xi(e^{i\tilde{\omega}}) \succ 0$  for some  $\tilde{\omega} \in (-\pi, \pi]$ , then  $\Phi_\xi(e^{i\omega}) \succ 0$  for all  $\omega \in (-\pi, \pi]$  if and only if  $\det \Phi_\xi(e^{i\omega})$  has no zeros. Thus,  $\mathcal{M}_\xi$  is strictly passive if and only if the following conditions all hold:

- (D1)  $\rho(A_\xi) < 1$  (asymptotic stability of  $A_\xi = \frac{A}{1-\xi}$ );
- (D2)  $\Phi_\xi(e^{i\tilde{\omega}}) \succ 0$  (positive definiteness at a unimodular point, say,  $e^{i\tilde{\omega}} = 1$ );
- (D3)  $\det \Phi_\xi(e^{i\omega}) \neq 0$  for all  $\omega \in (-\pi, \pi]$  (implying positive definiteness on the entire unit circle provided that (D1) and (D2) also hold).

In contrast to its continuous-time analogue (C2), note that (D2) does not require that  $D_\xi^H + D_\xi$  be positive definite (or even invertible).

A bracket containing the discrete-time  $\Xi$  is as follows. Again using the relation

between the linear matrix inequalities of  $\mathcal{M}$  and  $\mathcal{M}_\xi$ , with  $X = 2I_n$ , we can choose

$$(5.5) \quad \Xi_{\text{lb}} := \frac{1}{2} \lambda_{\min}(W_d(2I_n, \mathcal{M}))$$

as a lower bound on  $\Xi$ , since it follows that

$$(1 - \Xi_{\text{lb}})W_d(2I_n, \mathcal{M}_{\Xi_{\text{lb}}}) = W_d(2I_n, \mathcal{M}) - 2\Xi_{\text{lb}}I_{2n+m} \succeq 0$$

holds and so by Theorems 5.2 and 5.3, we have that  $\mathcal{M}_{\Xi_{\text{lb}}}$  is passive. Meanwhile,

$$(5.6) \quad \Xi_{\text{ub}} := 1 - \rho(A)$$

is an upper bound, since obviously  $\rho(A_\xi) < 1$  if  $\xi < \Xi_{\text{ub}}$ .

Given  $\xi \in [\Xi_{\text{lb}}, \Xi_{\text{ub}})$ , (D1) must always hold, so to verify strict passivity of  $\mathcal{M}_\xi$  we need to check that both (D2) and (D3) also hold. Checking (D2) is simple. If  $\lambda_{\min}(\Phi_\xi(e^{i\tilde{\omega}})) \leq 0$  for any  $\tilde{\omega} \in \mathbb{R}$ , then  $\mathcal{M}_\xi$  is not strictly passive, and there is no need to check (D3). Otherwise, since  $\rho(A_\xi) < 1$ , if  $\lambda_{\min}(\Phi_\xi(e^{i\tilde{\omega}})) > 0$ , we have that  $\mathcal{M}_\xi$  is strictly passive if and only if (D3) holds, which can be checked via the following result<sup>4</sup> (see, e.g., [8, 33]).

**THEOREM 5.4.** *Let  $\xi \in (-\infty, 1)$ ,  $\mathcal{M}_\xi$  and  $\Phi_\xi$  be as defined in (5.2), and  $z \in \mathbb{C}$  be any nonzero point such that  $z \notin \Lambda(A_\xi)$  and  $z^{-1} \notin \Lambda(A_\xi^H)$ , which is equivalent to the former when  $|z| = 1$ . Then  $\det \Phi_\xi(z) = 0$  if and only if  $\det(M_\xi - zN_\xi) = 0$ , where  $\tilde{D}_\xi := D^H + D - 2\xi I_m$  and the regular pencil  $M_\xi - \lambda N_\xi$  is defined by*

$$(5.7) \quad M_\xi := \begin{bmatrix} 0 & A & B \\ (\xi - 1)I_n & 0 & 0 \\ B^H & C & \tilde{D}_\xi \end{bmatrix} \quad \text{and} \quad N_\xi := \begin{bmatrix} 0 & (1 - \xi)I_n & 0 \\ -A^H & 0 & -C^H \\ 0 & 0 & 0 \end{bmatrix}.$$

Furthermore, if  $\tilde{D}_\xi$  is nonsingular, then  $\det \Phi_\xi(z) = 0$  if and only if  $\det(S_\xi - zT_\xi) = 0$ , where the symplectic pencil  $S_\xi - \lambda T_\xi$  is defined by

$$(5.8) \quad S_\xi := \begin{bmatrix} (\xi - 1)I_n & 0 \\ -B\tilde{D}_\xi^{-1}B^H & A - B\tilde{D}_\xi^{-1}C \end{bmatrix} \quad \text{and} \quad T_\xi := \begin{bmatrix} (B\tilde{D}_\xi^{-1}C - A)^H & C^H\tilde{D}_\xi^{-1}C \\ 0 & (1 - \xi)I_n \end{bmatrix}.$$

*Proof.* Writing

$$M_\xi - zN_\xi := \begin{bmatrix} 0 & A + (\xi - 1)zI_n & B \\ zA^H + (\xi - 1)I_n & 0 & zC^H \\ B^H & C & \tilde{D}_\xi \end{bmatrix},$$

and using the Schur identity of determinants with respect to the leading  $2n \times 2n$  block, which by assumption is nonsingular, we obtain that

$$\det(M_\xi - zN_\xi) = \det \begin{bmatrix} 0 & A + (\xi - 1)zI_n \\ zA^H + (\xi - 1)I_n & 0 \end{bmatrix} \det((1 - \xi)\Phi_\xi(z)).$$

As  $z \notin \Lambda(A_\xi)$  and  $z^{-1} \notin \Lambda(A_\xi^H)$ , the first equivalence involving  $M_\xi - \lambda N_\xi$  holds, and since  $\Phi_\xi(z)$  is regular, this matrix pencil must be a regular one. To obtain the second equivalence, we again apply the Schur identity of determinants, now with respect to

<sup>4</sup>The generalized eigenvalue problem given by the matrices in (5.7) is denoted  $\Gamma(\xi, \omega)$  in [21, p. 1263], but note that its bottom right block,  $D^H + D - \xi I_m$ , contains a typo; it should be  $D^H + D - 2\xi I_m$ , which we denote  $\tilde{D}_\xi$  in Theorem 5.4.

the trailing  $m \times m$  block, which is possible by our additional assumption that  $\tilde{D}_\xi$  is nonsingular. It then follows that  $\det(M_\xi - zN_\xi)$  is equal to

$$\det \tilde{D}_\xi \det \left( \begin{bmatrix} 0 & A + (\xi - 1)zI_n \\ zA^H + (\xi - 1)I_n & 0 \end{bmatrix} - \begin{bmatrix} B \\ zC^H \end{bmatrix} \tilde{D}_\xi^{-1} \begin{bmatrix} B^H & C \end{bmatrix} \right),$$

and so clearly  $\det(M_\xi - zN_\xi) = 0$  if and only if the second determinant above is zero. Multiplying the matrix inside this second determinant by  $\begin{bmatrix} 0 & I_n \\ I_n & 0 \end{bmatrix}$  from the left and rearranging terms yields  $S_\xi - zT_\xi$ . This matrix pencil is easily verified as symplectic, i.e., for  $J := \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$ ,  $S_\xi^H JS_\xi = T_\xi^H JT_\xi$  holds.  $\square$

**COROLLARY 5.5.** *Let  $\Xi_{\text{lb}}$  and  $\Xi_{\text{ub}}$  be as defined in (5.5) and (5.6), respectively, and let  $\xi \in [\Xi_{\text{lb}}, \Xi_{\text{ub}}]$ . Then the function  $\gamma_\xi : (-\pi, \pi] \rightarrow \mathbb{R}$  defined in (5.4) has at most  $2n$  zeros.*

*Proof.* If  $\gamma_\xi(\omega) = 0$ , then  $\det \Phi_\xi(e^{i\omega}) = 0$ . As  $\xi \in [\Xi_{\text{lb}}, \Xi_{\text{ub}}]$ ,  $\rho(A_\xi) < 1$  holds, and so the assumptions of Theorem 5.4 are met. Hence,  $\det \Phi_\xi(e^{i\omega}) = 0$  if and only if  $\det(M_\xi - e^{i\omega}N_\xi) = 0$ . The proof is completed by noting that  $\text{rank} N_\xi \leq 2n$ .  $\square$

Using Theorem 5.4, Mehrmann and Van Dooren proposed a bisection method to compute discrete-time  $\Xi$ , and via the following result, a discrete-time analogue of their improved procedure we described in subsection 4.1; for more details, see [21, section 7].

**THEOREM 5.6.** *Let  $\gamma : \mathbb{R} \times (-\pi, \pi] \rightarrow \mathbb{R}$  and  $\gamma_\xi : (-\pi, \pi] \rightarrow \mathbb{R}$  be as defined in (5.4), and  $\Xi_{\text{lb}}$  and  $\Xi_{\text{ub}}$  be as defined in (5.5) and (5.6), respectively. Then  $\gamma$  is continuous on the domain  $[\Xi_{\text{lb}}, \Xi_{\text{ub}}] \times (-\pi, \pi]$  and  $\gamma_\xi$  has the following properties:*

- (i) *if  $\xi \in [\Xi_{\text{lb}}, \Xi]$ , then  $\min_{\omega \in (-\pi, \pi]} \gamma_\xi(\omega) > 0$ ;*
- (ii) *if  $\xi = \Xi$ , then  $\min_{\omega \in (-\pi, \pi]} \gamma_\xi(\omega) = 0$ ;*
- (iii) *if  $\xi \in (\Xi, \Xi_{\text{ub}}]$  with  $\bar{\Xi} < \Xi_{\text{ub}}$ , then  $\gamma_\xi(\omega) < 0 \forall \omega \in \mathcal{S} \subseteq (-\pi, \pi]$ , where  $\mathcal{S}$  is the union of a finite number of nonoverlapping open intervals.*

*Proof.* Statements (i) and (ii) follow from [21], while (iii) follows from the facts that  $\gamma_\xi$  is continuous, and by Corollary 5.5, it can have at most  $2n$  zeros.  $\square$

**Remark 5.7.** For any  $\xi \in (\Xi, \Xi_{\text{ub}}]$ , the continuous-time function  $\gamma_\xi$  always has at least two zero-crossings, but note that the discrete-time version of  $\gamma_\xi$  may not have any zeros; i.e.,  $\max_{\omega \in (-\pi, \pi]} \gamma_\xi(\omega) < 0$  can hold. This is why it is necessary to check that both (D2) and (D3) hold at each estimate  $\xi$  encountered when computing discrete-time  $\Xi$ , but in the continuous-time case, only (C3) needs to be checked at each estimate. Note that as written, the pseudocodes for the discrete-time algorithms in [21, section 7] assume that (D2) holds automatically whenever (D3) holds, but for general use, it is indeed necessary to check (D2).

**5.2. An HEC-based algorithm for computing discrete-time  $\Xi$ .** By Theorem 5.6,  $\Xi$  defined in (5.3) can be computed via this root-min problem

$$(5.9) \quad \text{determine the } \xi \in \mathcal{D}_1 : \quad f(\xi) = \min_{\omega \in \mathcal{D}_2} \gamma(\xi, \omega) = 0,$$

where now  $\xi \in \mathcal{D}_1 = [\Xi_{\text{lb}}, \Xi_{\text{ub}}]$ ,  $\mathcal{D}_2 = [-\pi, \pi]$  is obviously compact,  $\gamma$  is defined in (5.4), and  $\gamma_\xi$  is bounded below. Our continuous-time HEC-based algorithm and results from subsection 4.2 extend to the discrete-time setting and work similarly, so for brevity, we only focus on the key points and differences.

Pseudocode for our new algorithm for discrete-time  $\Xi$  is given in Algorithm 5.1. Per Remark 5.7, the need to check that both (D2) and (D3) hold on each iteration

---

**Algorithm 5.1** HEC-based algorithm for discrete-time  $\Xi$ .
 

---

**Input:**  $\mathcal{M}$ ,  $\omega_0 \in (-\pi, \pi]$ ,  $\tau \in (0, 1)$ ,  $\Xi_{\text{lb}}$  (5.5), and  $\Xi_{\text{ub}}$  (5.6)  
**Output:**  $\xi$  such that  $|\Xi - \xi| \leq \tau|\Xi|$  for discrete-time  $\Xi$  for  $\mathcal{M}$

```

1:  $\xi \leftarrow \Xi_{\text{ub}} - \tau|\Xi_{\text{ub}}|$ 
2: if  $\xi \leq \Xi_{\text{lb}}$  then
3:   return
4: end if
5: find_negative  $\leftarrow \gamma_\xi(\omega_0) \geq 0$  // a boolean variable
6: while true do
7:   if find_negative then
8:     if  $\gamma_\xi(0) < 0$  then
9:        $\omega_0 \leftarrow 0$  // (D2) does not hold
10:    else
11:       $\Omega \leftarrow \{\omega \in (-\pi, \pi] : \det(M_\xi - e^{i\omega}N_\xi) = 0\}$ 
12:       $\Omega \leftarrow \{\omega \in \Omega : \gamma_\xi(\omega) = 0\}$ 
13:       $\Omega \leftarrow \Omega \cup \{\min \Omega + 2\pi\}$ 
14:      if  $\exists \omega_1, \omega_2 \in \Omega$  s.t.  $\omega_1 < \omega_2$  and  $\gamma_\xi(\omega) < 0 \forall \omega \in (\omega_1, \omega_2)$  then
15:         $\omega_0 \leftarrow 0.5(\omega_1 + \omega_2)$  // (D2) and (D3) do not hold
16:      else
17:        return //  $\gamma_\xi(\omega) \geq 0 \forall \omega \in (-\pi, \pi]$  and  $\xi \approx \Xi$  to tolerance
18:      end if
19:    end if
20:  end if
21:  //  $\gamma_\xi(\omega_0) < 0$  and  $\Xi \in [\Xi_{\text{lb}}, \xi)$  so run HEC with this initial data
22:   $(\tilde{\xi}, \tilde{\omega}) \leftarrow$  a pseudoroot of (5.9) obtained by HEC with  $\Xi \leq \tilde{\xi} < \xi$ 
23:   $\xi \leftarrow \tilde{\xi} - \tau|\tilde{\xi}|$ 
24:  find_negative  $\leftarrow$  true
25: end while

```

---

NOTE: See Remark 4.8 and Algorithm 4.1 for more details on tolerances and symmetry. In line 13,  $\Omega \cup \{\min \Omega + 2\pi\}$  is used so that the “wrap-around” interval, i.e.,  $[\max \Omega, \min \Omega + 2\pi]$  is not missed.

means that the pseudocode is a bit more complicated than for continuous-time  $\Xi$ . As such, one might conclude that the problem of computing  $\Xi$  is trickier in the discrete-time case; however, as we explain in the numerical results, it seems that the exact opposite is true, due to a numerical issue that only arises in the continuous-time case. To implement HEC for Algorithm 5.1, we make use of the first and second derivatives of  $\gamma_\xi$  and  $\gamma_\omega$  defined in (5.4). To that end, we provide the discrete-time analogues of the matrix derivatives given in (4.10), as the remaining computations are readily apparent. Letting  $Z_k := C((1 - \xi)e^{i\omega}I_n - A)^{-k}B$ , we have that

$$(5.10a) \quad \frac{\partial}{\partial \xi} \mathcal{T}_\xi(e^{i\omega}) = \frac{\mathcal{T}_\xi(e^{i\omega}) + e^{i\omega}Z_2 - I_m}{1 - \xi}, \quad \frac{\partial^2}{\partial \xi^2} \mathcal{T}_\xi(e^{i\omega}) = \frac{2}{1 - \xi} \left( e^{2i\omega}Z_3 + \frac{\partial \mathcal{T}_\xi(e^{i\omega})}{\partial \xi} \right),$$

$$(5.10b) \quad \frac{\partial}{\partial \omega} \mathcal{T}_\xi(e^{i\omega}) = -ie^{i\omega}Z_2, \quad \frac{\partial^2}{\partial \omega^2} \mathcal{T}_\xi(e^{i\omega}) = e^{i\omega}Z_2 - 2(1 - \xi)e^{2i\omega}Z_3.$$

In Algorithm 5.1, the costs to run HEC and compute zeros of  $\gamma_\xi$  are the same as in the continuous-time setting discussed in subsection 4.2. Theorem 4.9 also extends, and so under mild assumptions that generally hold in practice, Algorithm 5.1 converges quadratically to pseudoroots of (5.9). Relatedly, Mehrmann and Van Dooren’s improved algorithm [21, section 7] for discrete-time  $\Xi$  also converges at

least superlinearly.

**6. Numerical experiments.** We implemented the continuous- and discrete-time versions of our new HEC-based method and the improved midpoint-based iteration of Mehrmann and Van Dooren. In this section, for brevity, we use HEC to refer to former (Algorithms 4.1 and 5.1) and MP (for midpoint) to refer to the latter. All codes were implemented with relative tolerances and set to compute  $\Xi$  to 14 digits. Experiments were done using MATLAB R2021a on a 2020 MacBook Pro with a quad-core Intel Core i5 1038NG7 CPU and 16 GB of RAM running macOS 10.15.7. Code and data to reproduce all experiments is included in the supplementary material (hec\_pass\_opt\_code.zip [local/web 1.27MB]).

**6.1. Implementation details.** We first discuss implementing Algorithm 3.1. The expansion phase was implemented using `fmincon`, while the contraction phase was implemented using our own Halley-bisection root-finding code; first and second derivative information is used in both. Due to rounding errors, it may be that the contraction phase sometimes computes an approximate root  $\hat{\varepsilon}_k$  of  $g_{x_k}$  such that  $g_{x_k}(\hat{\varepsilon}_k) < 0$ , instead of  $g_{x_k}(\hat{\varepsilon}_k) \geq 0$ , which is required at every iteration (for a root-max problem). However, if this occurs, it suffices to just perturb the computed root by a small multiple of the Halley step to correct the sign; a more complicated workaround involving shifting the root problems is suggested in [24, section 7] and [12, Appendix A], but we do not recommend that. Algorithm 3.1 is terminated at an approximate pseudoroot once both  $\varepsilon_k$  and  $x_k$  are no longer changing significantly with respect to their respective previous values; this condition is checked twice per iteration, after the contraction phase and after the expansion phase. Since in the context of computing  $\Xi$ , the expansion phases can be solved quickly, we did not use early contraction.

For simplicity, we used `eig` for all eigenvalue problems, though it is advisable to use structure-preserving solvers for numerical robustness; e.g., see [1, 2, 17, 33]. To compute zeros of  $\gamma_\xi(\omega)$ , we used the pencils given by the matrices in (4.7) and (5.7) and, respectively, identified their real and unimodular eigenvalues using a tolerance. Note that if  $\gamma_\xi$  has a minimizer or maximizer  $\hat{\omega}$  such that  $\gamma_\xi(\hat{\omega}) = 0$  (or approximately equal), then this corresponds to a (nearly) multiple eigenvalue (with multiplicity at least two) of the pencil given by (4.7) or (5.7), as appropriate. This always happens as any of the methods approach  $\Xi$ , and it is generally also true at computed pseudoroots and at  $\omega = 0$  when the problems have symmetry. Due to rounding errors, such eigenvalues, even when computed via a structure-preserving solver, may not be detected as (close to) real or unimodular. If this happens, a zero of  $\gamma_\xi$  will be missed, which in turn can cause any of the algorithms to stagnate. Fortunately, a robust fix is easy: if  $(\tilde{\xi}, \tilde{\omega})$  is the most recent computed pseudoroot, simply explicitly add  $\tilde{\omega}$  as a zero of  $\gamma_\xi$ ; a similar fix is also necessary for the MP. For more details, see [6, pp. 371–373], where this fix was proposed in the context of computing the pseudospectral abscissa.

For continuous-time  $\Xi$ , there is an additional numerical difficulty when computing the zeros of  $\gamma_\xi$  when  $\xi \approx \Xi_{\text{ub}}$ . Although these zeros must be finite, they still may be arbitrarily far away from the origin, and so there may be large errors in the imaginary parts of the corresponding computed real eigenvalues of  $M_\xi - \lambda N$ . Mehrmann and Van Dooren recommended using a tolerance so that the first estimate  $\xi$  tested was sufficiently far away from  $\Xi_{\text{ub}}$  to help avoid such problems. However, we have observed that even a relatively large perturbation may still be insufficient to avoid failure of the MP. Our MP code implementing their method uses  $\xi_0 = \Xi_{\text{ub}} - |\Xi_{\text{ub}}|10^{-4}$ , but only

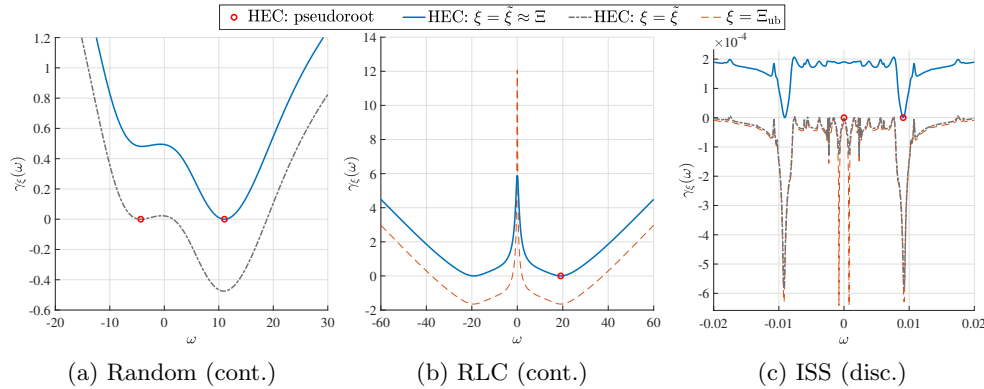


FIG. 3. The pseudoroots  $(\tilde{\xi}, \tilde{\omega})$  and corresponding estimates  $\tilde{\xi}$  for  $\Xi$  computed by HEC until convergence with  $\tilde{\xi} \approx \Xi$ .

small perturbations are done for subsequent estimates in order to obtain the desired 14-digit accuracy; of course, if  $\Xi \approx \Xi_{ub}$ , high accuracy may not be possible with MP. In contrast, our HEC-based method is much less susceptible to this issue, since even if only one root of  $\gamma_\xi$  is detected, it generally can still be used to start Algorithm 3.1. Even if this root is a stationary point, a small perturbation to the left or right generally yields a point for starting Algorithm 3.1. In general, structure-preserving eigensolvers can be used or one can increase the allowed amount of rounding error in the imaginary part of an eigenvalue in proportion with the magnitude of the eigenvalue.

Finally, in line 8 of Algorithm 5.1 when checking (D2), instead of always looking at the sign of  $\gamma_\xi(0)$ , after the first pseudoroot has been computed we instead test if  $\gamma_\xi(\tilde{\omega} + \frac{1}{2}\pi) < 0$ , and if so set  $\omega_0 \leftarrow \tilde{\omega} + \frac{1}{2}\pi$  in line 9. The reason is because if the previous pseudoroot has  $\tilde{\omega} = 0$ ,  $\gamma_\xi(\tilde{\omega}) < 0$  almost always holds due to rounding error even though it should be exactly zero. Shifting by, e.g.,  $\frac{1}{2}\pi$ , ensures that (D2) is checked at a new point; note that shifting by  $\pi$  or  $2\pi$  would not ensure this.

**6.2. Experiments.** We begin with a randomly generated continuous-time example with complex matrices (denoted Random) to illustrate (i) when our method encounters at least two pseudoroots before converging (see Figure 3(a)) and (ii) the aforementioned difficulty of computing zeros of  $\gamma_\xi$  when  $\xi \approx \Xi_{ub}$  (see Figure 4(a)). In Table 1, we see that MP is about ten times slower than HEC. Although HEC required more computations of  $\lambda_{\min}(\Phi_\xi(e^{i\omega}))$ , it only needed to solve two of the large eigenvalue problems involving  $M_\xi - \lambda N$ . Meanwhile, MP required 27 solves with the pencils and took 14 iterations to converge. HEC converged to  $\Xi$  at its second pseudoroot, and Algorithm 3.1 on average took 5.0 iterations to converge to a pseudoroot.

Our second continuous-time example is the electric RLC circuit model used in [3]. We refer to Figure 3b and Table 1 for the complete performance details, but note that HEC was over three times faster than MP for this RLC example, with both methods converging faster and with less work than on the random example.

To compare the discrete-time methods, we used the ISS model from the SLICOT benchmark examples.<sup>5</sup> Since ISS is a continuous-time model, we converted it to a minimal discrete-time one by calling `c2d` using a sampling time of 0.001 followed by `minreal`. In Figure 3b and Table 1, we see that HEC was almost 23 times faster

<sup>5</sup> Available at <http://slicot.org/20-site/126-benchmark-examples-for-model-reduction>.



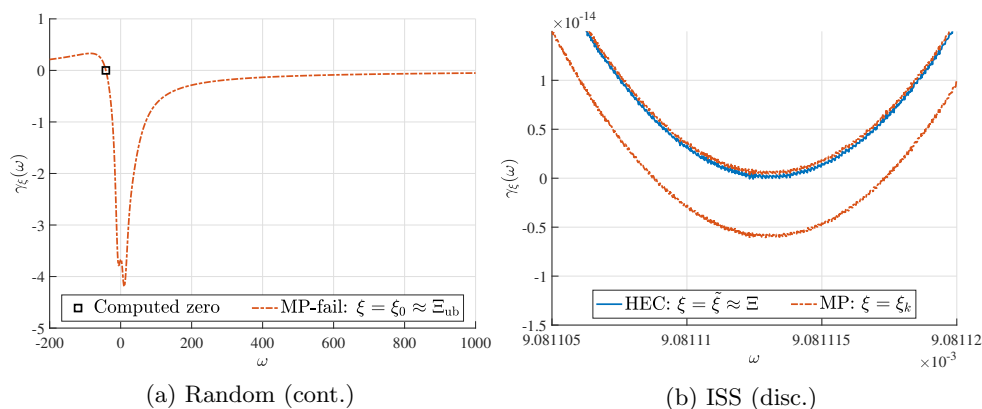


FIG. 4. On the left, MP was initialized with  $\xi_0 = \Xi_{ub} - |\Xi_{ub}|10^{-5}$ . In this case,  $\gamma_{\xi_0}$  has two zeros at approximately  $-41.6$  and  $460600.9$ , but the latter is not detected due to rounding errors when computing the eigenvalues of  $M_{\xi_0} - \lambda N$ . Consequently, the MP erroneously terminates at  $\xi_0$  with no digits of accuracy but does converge properly when initialized with  $\xi_0 = \Xi_{ub} - |\Xi_{ub}|10^{-4}$ . On the right, we see that eigenvalue computations used in the MP to compute the smallest roots of  $\gamma_\omega$  incur more rounding errors than our HEC-based approach. See [4, section 9.2] for an example where half of the precision can be lost when solving root problems using these eigenvalue techniques.

TABLE 1

MP and HEC compared on two continuous-time (cont.) and one discrete-time (disc.) examples. For Random, the MP was tested in two configurations: MP-fail with  $\xi_0 = \Xi_{ub} - |\Xi_{ub}|10^{-5}$  and MP with  $\xi_0 = \Xi_{ub} - |\Xi_{ub}|10^{-4}$ . The number of iterations is shown in the “iters.” column; the average number of iterations of Algorithm 3.1 is also given in parentheses for HEC. The number of eigenvalue problems solved is shown under the “# eig (order, type)” columns, separated into the number of order  $2n + m$  matrix pencils “(2n + m, P)” and the number of order  $m$  matrices “(m, M)”. The overall running time in seconds is given under “time (sec.)”, while the computed estimates for  $\Xi$  are given in the rightmost column.

Alg.	iters.	# eig (order, type)		time (sec.)	$\Xi$ estimate
		(2n + m, P)	(m, M)		
Random (n = 200, m = 10, cont.) — $[\Xi_{lb}, \Xi_{ub}] = [-25.56407, -10.90965]$					
MP-fail	1	1	1	0.440	-10.9097612001839
MP	14	27	49	8.892	-14.4073741346323
HEC	2(5.0)	2	78	0.909	-14.4073741346323
RLC (n = 200, m = 1, cont.) — $[\Xi_{lb}, \Xi_{ub}] = [-32.1267, 2.022606]$					
MP	4	8	18	2.360	0.562483988863916
HEC	1(4.0)	2	41	0.767	0.562483988863891
ISS (n = 228, m = 3, disc.) — $[\Xi_{lb}, \Xi_{ub}] = [-3.007437, 3.117278 \times 10^{-6}]$					
MP	15	29	550	8.490	$-9.37320364701040 \times 10^{-5}$
HEC	2(4.0)	1	97	0.374	$-9.37320364699013 \times 10^{-5}$

than MP, again due to the great disparity in the number of large generalized eigenvalue problems solved. In fact, for ISS, HEC also solved far fewer smaller standard eigenvalue problems as well. From Table 1 and Figure 4b, we also see that MP did not quite compute  $\Xi$  to the requested 14-digit accuracy, while HEC apparently did.

This slight inaccuracy is the result of MP solving root problems via solving eigenvalue problems, but such errors can be larger; see the caption of Figure 4b for more details.

While we have established that MP converges at least superlinearly, an examination of its iterates (not shown) seems to indicate that it too may converge quadratically like HEC. However, as demonstrated by Random and ISS, where MP respectively required 14 and 15 iterations, MP can incur many iterations before it gets near its faster convergence regime. The key problem on these examples is that MP chooses the largest interval where  $\gamma_\xi$  is negative to determine how to reduce the estimate  $\xi$ . But this can be a particularly bad strategy if  $\gamma_\xi$  has a zero very far away from the origin, as is the case for both Random and ISS when  $\xi \approx \Xi_{\text{ub}}$ . While one could consider altering this strategy to improve performance, such an MP variant would still be slower than HEC and also still have the aforementioned numerical issues.

**7. Conclusion.** By generalizing the HEC algorithm, we have presented faster and more numerically robust algorithms to compute  $\Xi$ , the extremal real value for which a given parametric linear time-invariant system is passive, a problem which is linked to maximizing the passivity radius. Our new methods outperform the existing algorithms of Mehrmann and Van Dooren, and for large-scale problems, when using sparse eigenvalue solvers, HEC can be used by itself to efficiently estimate  $\Xi$ , which the earlier methods cannot do. We hope that our generalization of HEC, its convergence guarantees, and identification of root-max problems will also help facilitate new fast and robust numerical methods for other quantities, for small- and large-scale problems.

While we have established local rates of convergence for our new methods (at least quadratic) and the earlier methods of Mehrmann and Van Dooren [22, 21] (at least superlinear), one thing that remains unaddressed is the question of global convergence, i.e., do these methods have unconditional convergence to  $\Xi$ ? We believe that they do, but a potential wrinkle towards proving this is that the measure of the set of intervals where  $\gamma_\xi$  is negative (see Corollaries 4.5 and 5.5) is not always decreasing. Consequently, computing  $\Xi$  is a fundamentally different problem than maximizing a one-variable function using the level-set technique of Boyd and Balakrishnan [5].

**Acknowledgment.** The authors are grateful to the referees and the editor for carefully reviewing the paper and providing many helpful comments.

#### REFERENCES

- [1] P. BENNER, R. BYERS, V. MEHRMANN, AND H. XU, *Numerical methods for linear-quadratic and  $H_\infty$  control problems*, in Dynamical Systems, Control, Coding, Computer Vision: New Trends, Interfaces, and Interplay, G. Picci and D. S. Gilliam, eds., Progr. Systems Control Theory 25, Birkhäuser, Basel, 1999, pp. 203–222, [https://doi.org/10.1007/978-3-0348-8970-4\\_10](https://doi.org/10.1007/978-3-0348-8970-4_10).
- [2] P. BENNER, R. BYERS, V. MEHRMANN, AND H. XU, *Numerical computation of deflating subspaces of skew-Hamiltonian/Hamiltonian pencils*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 165–190, <https://doi.org/10.1137/S0895479800367439>.
- [3] P. BENNER, P. GOYAL, AND P. VAN DOOREN, *Identification of port-Hamiltonian systems from frequency response data*, Systems Control Lett., 143 (2020), 104741, <https://doi.org/10.1016/j.sysconle.2020.104741>.
- [4] P. BENNER AND T. MITCHELL, *Extended and improved criss-cross algorithms for computing the spectral value set abscissa and radius*, SIAM J. Matrix Anal. Appl., 40 (2019), pp. 1325–1352, <https://doi.org/10.1137/19M1246213>.
- [5] 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 (1990), pp. 1–7, [https://doi.org/10.1016/0167-6911\(90\)90037-U](https://doi.org/10.1016/0167-6911(90)90037-U).

- [6] J. V. BURKE, A. S. LEWIS, AND M. L. OVERTON, *Robust stability and a criss-cross algorithm for pseudospectra*, IMA J. Numer. Anal., 23 (2003), pp. 359–375, <https://doi.org/10.1093/imanum/23.3.359>.
- [7] R. S. DEMBO, S. C. EISENSTAT, AND T. STEIHAUG, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408, <https://doi.org/10.1137/0719025>.
- [8] H. FASSBENDER, *Symplectic Methods for the Symplectic Eigenvalue Problem*, Kluwer Academic, Dordrecht, Netherlands, 2002.
- [9] A. FAZZI, N. GUGLIELMI, AND C. LUBICH, *Finding the nearest passive or nonpassive system via Hamiltonian eigenvalue optimization*, SIAM J. Matrix Anal. Appl., 42 (2021), pp. 1553–1580, <https://doi.org/10.1137/20M1376972>.
- [10] N. GILLIS AND P. SHARMA, *Finding the nearest positive-real system*, SIAM J. Numer. Anal., 56 (2018), pp. 1022–1047, <https://doi.org/10.1137/17M1137176>.
- [11] K. GLOVER, *All optimal Hankel-norm approximations of linear multivariable systems and their  $L^\infty$ -error norms*, Internat. J. Control, 39 (1984), pp. 1115–1193, <https://doi.org/10.1080/00207178408933239>.
- [12] N. GUGLIELMI, M. GÜRBÜZBALABAN, T. MITCHELL, AND M. L. OVERTON, *Approximating the real structured stability radius with Frobenius-norm bounded perturbations*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 1323–1353, <https://doi.org/10.1137/16M1110169>.
- [13] N. GUGLIELMI, M. GÜRBÜZBALABAN, AND M. L. OVERTON, *Fast approximation of the  $H_\infty$  norm via optimization over spectral value sets*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 709–737, <https://doi.org/10.1137/120875752>.
- [14] D. HINRICHSSEN AND A. J. PRITCHARD, *A note on some differences between real and complex stability radii*, Systems Control Lett., 14 (1990), pp. 401–408, [https://doi.org/10.1016/0167-6911\(90\)90090-H](https://doi.org/10.1016/0167-6911(90)90090-H).
- [15] D. HINRICHSSEN AND A. J. PRITCHARD, *Real and complex stability radii: A survey*, in Control of Uncertain Systems (Bremen, 1989), Progr. Systems Control Theory 6, Birkhäuser Boston, Boston, MA, 1990, pp. 119–162, [https://doi.org/10.1007/978-1-4757-2108-9\\_7](https://doi.org/10.1007/978-1-4757-2108-9_7).
- [16] D. HINRICHSSEN AND A. J. PRITCHARD, *Mathematical Systems Theory I*, Springer, Berlin, 2005, <https://doi.org/10.1007/b137541>.
- [17] D. KRESSNER, C. SCHRÖDER, AND D. WATKINS, *Implicit QR algorithms for palindromic and even eigenvalue problems*, Numer. Algorithms, 51 (2010), pp. 209–238.
- [18] P. LANCASTER, *On eigenvalues of matrices dependent on a parameter*, Numer. Math., 6 (1964), pp. 377–387, <https://doi.org/10.1007/BF01386087>.
- [19] A. LAUB, *Efficient multivariable frequency response computations*, IEEE Trans. Automat. Control, 26 (1981), pp. 407–408, <https://doi.org/10.1109/TAC.1981.1102602>.
- [20] V. MEHRMANN, *The Autonomous Linear Quadratic Control Problem, Theory and Numerical Solution*, Lect. Notes Control Inf. Sci. 163, Springer, Heidelberg, 1991.
- [21] V. MEHRMANN AND P. VAN DOOREN, *Optimal robustness of passive discrete-time systems*, IMA J. Math. Control Inform., 37 (2020), pp. 1248–1269, <https://doi.org/10.1093/imamci/dnaa013>.
- [22] V. MEHRMANN AND P. M. VAN DOOREN, *Optimal robustness of port-Hamiltonian systems*, SIAM J. Matrix Anal. Appl., 41 (2020), pp. 134–151, <https://doi.org/10.1137/19M1259092>.
- [23] T. MITCHELL, *Robust and Efficient Methods for Approximation and Optimization of Stability Measures*, Ph.D. thesis, New York University, New York, NY, 2014, <http://cs.nyu.edu/media/publications/mitchell.tim.pdf>.
- [24] T. MITCHELL AND M. L. OVERTON, *Hybrid expansion-contraction: A robust scalable method for approximating the  $H_\infty$  norm*, IMA J. Numer. Anal., 36 (2016), pp. 985–1014, <https://doi.org/10.1093/imanum/drv046>.
- [25] T. MITCHELL AND M. L. OVERTON, *On properties of univariate max functions at local maximizers*, Optim. Lett., 16 (2022), pp. 2527–2541, <https://doi.org/10.1007/s11590-022-01872-y>.
- [26] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, Springer, New York, 1999, <https://doi.org/10.1007/b98874>.
- [27] M. L. OVERTON AND P. VAN DOOREN, *On computing the complex passivity radius*, in Proceedings of the 44th IEEE Conference on Decision and Control, IEEE, Piscataway, NJ, 2005, pp. 7960–7964, <https://doi.org/10.1109/CDC.2005.1583449>.
- [28] M. L. OVERTON AND R. S. WOMERSLEY, *Second derivatives for optimizing eigenvalues of symmetric matrices*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 697–718, <https://doi.org/10.1137/S089547989324598X>.
- [29] R. T. ROCKAFELLAR AND R. J.-B. WETS, *Variational Analysis*, Grundlehren Math. Wiss. 317, Springer, Berlin, 1998, <https://doi.org/10.1007/978-3-642-02431-3>.

- [30] P. VAN DOOREN, *The generalized eigenstructure problem in linear system theory*, IEEE Trans. Automat. Control, 26 (1981), pp. 111–129, <https://doi.org/10.1109/TAC.1981.1102559>.
- [31] C. F. VAN LOAN, *How near is a stable matrix to an unstable matrix?*, in Linear Algebra and its Role in Systems Theory (Brunswick, Maine, 1984), Contemp. Math. 47, American Mathematical Society, Providence, RI, 1985, pp. 465–478, <https://doi.org/10.1090/conm/047/828319>.
- [32] J. C. WILLEMS, *Dissipative dynamical systems. II. Linear systems with quadratic supply rates*, Arch. Ration. Mech. Anal., 45 (1972), pp. 352–393, <https://doi.org/10.1007/BF00276494>.
- [33] H. XU, *On equivalence of pencils from discrete-time and continuous-time control*, Linear Algebra Appl., 414 (2006), pp. 97–124.
- [34] K. ZHOU, J. C. DOYLE, AND K. GLOVER, *Robust and Optimal Control*, Prentice-Hall, Upper Saddle River, NJ, 1996.