Automatica 113 (2020) 108685

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

On the real stability radius of sparse systems[☆]

Vaibhav Katewa[*], Fabio Pasqualetti

Department of Mechanical Engineering, University of California Riverside, Riverside, CA, 92507, USA

ARTICLE INFO

Article history:
Received 4 October 2018
Received in revised form 6 September 2019
Accepted 14 October 2019
Available online 14 November 2019

Keywords:
Stability radius
Sparse network systems
Robust stability
Optimization

ABSTRACT

In this paper, we study robust stability of sparse LTI systems using the stability radius (SR) as a robustness measure. We consider real perturbations with an arbitrary and pre-specified sparsity pattern of the system matrix and measure their size using the Frobenius norm. We formulate the SR problem as an equality-constrained minimization problem. Using the Lagrangian method for optimization, we characterize the optimality conditions of the SR problem, thereby revealing the relation between an optimal perturbation and the eigenvectors of an optimally perturbed system. Further, we use the Sylvester equation based parametrization to develop a penalty based gradient/Newton descent algorithm which converges to the local minima of the optimization problem. Finally, we illustrate how our framework provides structural insights into the robust stability of sparse networks.

1. Introduction

Guaranteed stability under parameter uncertainty is one of the central problems in robust design of dynamical systems. Consider the following Linear Time-Invariant (LTI) dynamical system

$$\mathcal{D}x(t) = Ax(t), \tag{1}$$

where $x \in \mathbb{R}^n$ is the state, $A \in \mathbb{R}^{n\times n}$, and $\mathcal{D}$ can either be the continuous-time differential operator (i.e., $\mathcal{D}x(t) = \dot{x}(t)$) or the discrete-time shift operator (i.e., $\mathcal{D}x(t) = x(t+1)$). Let the complex plane $\mathbb{C}$ be divided into any two disjoint sets as $\mathbb{C} = \mathbb{C}_g \cup, \mathbb{C}_b$, where $\mathbb{C}_g$ is open. Stability of (1) requires the eigenvalues of A to lie in the stability region $\mathbb{C}_g$. Assuming that A is stable, the robust stability analysis of (1) involves characterizing the eigenvalues of the affine perturbations of A given by

$$A \rightsquigarrow A(\Delta) \triangleq A + B\Delta C, \tag{2}$$

where $\Delta \in \mathbb{R}^{m\times p}$ is the perturbation matrix and $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$ are structure matrices. The perturbed matrix $A(\Delta)$ can also be interpreted as the closed loop matrix of the following linear system

$$\mathcal{D}x(t) = Ax(t) + Bu(t),$$
$$y(t) = Cx(t),$$

with static feedback $u(t) = \Delta y(t)$.

There have been numerous studies on eigenvalue characterization and stability of the perturbed matrix (2) (see Hinrichsen & Pritchard, 2005 for a comprehensive treatment). However, an inherent *crucial assumption* in these studies (and of the robust stability theory) is that all entries of the perturbation Δ are allowed to be freely perturbed. Clearly, this assumption is not applicable in modern control systems which are increasingly networked and distributed in nature and, as a result, exhibit a specific sparsity structure. In such systems, the matrix A typically has an associated sparsity pattern, and its certain entries are fixed/zero, and it is feasible to perturb only the non-fixed entries of A. Therefore, the perturbations Δ applied to A cannot be chosen freely and must satisfy certain sparsity constraints as well.

In this paper, we develop a novel framework to study the robust stability of LTI systems with sparsity constraints. Let $S \in \{0,1\}^{m\times p}$ be a binary matrix that specifies the sparsity structure of the perturbation Δ. Specifically,

$$\Delta_{ij} = \begin{cases} 0 & \text{if } S_{ij} = 0, \text{ and} \\ \star & \text{if } S_{ij} = 1, \end{cases}$$

where $\star$ denotes any real number. Further, let $\boldsymbol{\Delta}_S$ denote the set of sparse perturbations given by

$$\boldsymbol{\Delta}_S = \{\Delta \in \mathbb{R}^{m\times p} : S^c \circ \Delta = 0\}, \tag{3}$$

where $S^c \triangleq 1_{m\times p} - S$ denotes the complementary sparsity structure matrix and $\circ$ denotes the Hadamard (element-wise) product. We consider the notion of Stability Radius (SR) as the measure of robust stability, which is the minimum-size real perturbation

[☆] This work was supported in part by ARO, USA award 71603NSYIP and AFOSR, USA award FA9550-19-1-0235. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Tongwen Chen under the direction of Editor Ian R. Petersen.

[*] Corresponding author.
E-mail addresses: vkatewa@engr.ucr.edu (V. Katewa), fabiopas@engr.ucr.edu (F. Pasqualetti).

https://doi.org/10.1016/j.automatica.2019.108685
0005-1098/© 2019 Elsevier Ltd. All rights reserved.

that moves an eigenvalue(s) of $A(\Delta)$ outside the stability region. Formally, the SR is defined as

$$r \triangleq \inf\{\|\Delta\| : \Gamma(A(\Delta)) \cap \mathbb{C}_b \neq \emptyset,\ \Delta \in \mathbf{\Delta}_S\}, \tag{4}$$

where $\Gamma(\cdot)$ denotes the spectrum of a matrix, and $\|\cdot\|$ can be the spectral or Frobenius norm (see the discussion in next paragraph). The SR provides a worst case measure for the robustness of the system in the sense that all perturbations with $\|\Delta\| < r$ are guaranteed to preserve the stability of the perturbed system. Note that while matrices B and C can be chosen to impose certain perturbation structures on A (such as zero column(s) or row(s)), they cannot capture arbitrary sparsity constraints. Thus, we require the explicit sparsity constraint $\Delta \in \mathbf{\Delta}_S$ in (4).

The perturbation size $\|\Delta\|$ can be measured using the spectral norm or Frobenius norm, and these respective cases are referred to as 2-norm SR and F-norm SR. The case when Δ is allowed to be complex is referred to as complex SR. The case $B = C = I$ and $\mathbf{\Delta}_S = \mathbb{R}^{m\times p}$, where each entry of A is allowed to be perturbed independently, is referred to as unstructured SR. In this paper, we study the sparse, real, F-norm SR problem by formulating it as an equality-constrained optimization problem. The real SR problem is more suitable for engineering applications where the dynamics matrix A and its perturbations are typically real. Further, unlike the spectral norm, the Frobenius norm explicitly measures the entry-wise perturbations of Δ, which is useful in characterizing the *size and structure* of sparse perturbations.

Related work The stability radius problem without sparsity constraints has a rich history. Although robust stability problems have been studied in various forms in the past, the notion of 2-norm stability radius was introduced formally in Hinrichsen and Pritchard (1986a, 1986b). Various bounds and characterizations of unstructured, complex, and real SRs were given in Hinrichsen and Pritchard (1986a) and Van Loan (1985). In Hinrichsen and Pritchard (1986b), characterizations of structured, complex SR were presented in terms of the H_∞-norm of the associated transfer function and solutions of parameterized algebraic Riccati equations. Bisection algorithms to compute the complex SR were presented in Byers (1988) and Hinrichsen, Kelb, and Linnemann (1989) and algorithms to compute the H_∞-norm were given in Boyd, Balakrishnan, and Kabamba (1989), Boyd and Balakrishnan (1990) and Bruinsma and Steinbucha (1990). Since the optimal perturbation for the complex SR problem always has rank 1 (Hinrichsen & Pritchard, 2005), the 2-norm and the F-norm SRs are equal for the complex case.

The real SR problem is considerably more difficult than its complex counterpart (Hinrichsen & Pritchard, 1986a). Qiu et al. presented several lower bounds for the unstructured case in Qiu and Davison (1991, 1992) and a complete algebraic characterization of the structured case was presented in Qiu et al. (1995). Based on this characterization, a level-set algorithm was developed in Sreedhar, Van Dooren, and Tits (1996) for the structured case and an implicit determinant method was provided for the unstructured case in Freitag and Spence (2014). For a comprehensive treatment of the 2-norm SR problem, see Hinrichsen and Pritchard (2005, Chapter 5).

While the 2-norm SR problem has been studied extensively, there are limited studies on the F-norm SR problem. Note that due to the fundamental difference between the spectral and Frobenius norms, the procedure in Qiu et al. (1995) to characterize the real, 2-norm SR cannot be used to characterize the real, F-norm SR. In Bobylev, Bulatov, and Diamond (1999, 2001), lower bounds on the real, F-norm SR were provided for the unstructured and structured cases, respectively. Recently, a number of works have appeared that use iterative algorithms to approximate the 2-norm/F-norm real SR (Guglielmi, 2016; Guglielmi, Gurbuzbalaban, Mitchell, & Overton, 2017; Guglielmi & Lubich, 2013; Guglielmi & Manneta, 2015; Rostami, 2015). Typically, these algorithms use two levels of iterations. The inner iteration approximates the rightmost (outermost, for discrete-time case) points of spectral value sets, and the outer iteration verifies the intersection of these points with the stability boundary. All of these aforementioned studies consider the non-sparse SR problem.

In a very recent paper (Johnson, Wicks, Zefran, & DeCarlo, 2018), which was developed independently and concurrently with our paper, the authors study the structured distance of an LTI system from the set of systems that do not exhibit a general property $\mathcal{P}$, such as controllability, observability, and stability. They provide necessary conditions for a locally optimal perturbation and develop an algorithm to obtain such solution. Since the framework in Johnson et al. (2018) is developed for a general class of problems, the provided necessary conditions are implicit in terms of abstract linear maps. On the other hand, we use a different approach to obtain stronger and explicit necessary conditions for the sparse SR problem. In addition, we provide sufficient conditions for a local minimum, thereby completely characterizing the local minima. Finally, our gradient/Newton descent algorithm is simpler than the algorithm proposed in Johnson et al. (2018).

Contributions The contribution of the paper is two-fold. First, we propose a novel approach to compute the sparse SR by formulating the SR problem as an equality-constrained minimization problem. We characterize its local optimality conditions, thereby revealing important geometric properties of the optimal perturbed system. Second, using the Sylvester equation based parametrization, we develop a penalty-based gradient algorithm to solve the optimization problem that is guaranteed to converge to a local minimum. Numerical studies are included to illustrate various properties of the optimal perturbations and the algorithm, and to highlight the usefulness of the framework for sparse networks.

Paper organization In Section 2 we present our mathematical notation and some properties that we use in the paper, and formulate the SR problem as an optimization problem with equality constraints. Section 3 contains the local optimality conditions of the SR problem. In Section 4 we develop a gradient based algorithm to compute local solutions. Numerical examples are presented in Section 5. Finally, we conclude the paper in Section 6.

2. Problem formulation

2.1. Mathematical notation and properties

We use the following notation throughout the paper: $\|\cdot\|_F$ and $\|\cdot\|_2$ denote the Frobenius and spectral norm of a matrix, respectively. $\circ$ and $\otimes$ denote the Hadamard and Kronecker product, respectively. The identity matrix is denoted by I. $\Gamma(\cdot)$, $(\cdot)^\mathsf{T}$ and $\mathrm{tr}(\cdot)$ denote the spectrum, transpose and trace of a matrix, respectively. $(\cdot)^+$ and $\alpha(\cdot)$ denote the pseudo-inverse and spectral abscissa of a matrix, respectively. A positive-definite matrix A is denoted by $A > 0$. x^* and x^H denote the complex conjugate and the conjugate transpose, respectively of a vector x. $\mathrm{Re}(\cdot)$ and $\mathrm{Im}(\cdot)$ denote the real and imaginary parts of a complex number, respectively. $\mathrm{vec}(\cdot)$ denotes the vectorization of a matrix. $\mathrm{diag}(a)$ denotes a $n \times n$ diagonal matrix with diagonal elements given by $a \in \mathbb{R}^n$. $1_{m\times n}$ denotes a $m \times n$ matrix of all ones. Finally, $j = \sqrt{-1}$ denotes the unit imaginary number.

We use the following mathematical properties for the derivation of our results (Magnus & Neudecker, 1999; Petersen & Pedersen, 2012):

P.1 $\|A\|_\mathrm{F}^2 = \mathrm{tr}(A^\mathsf{T}A) = \mathrm{vec}(A)^\mathsf{T}\mathrm{vec}(A)$,

P.2 $\text{vec}(AB) = (I \otimes A)\text{vec}(B) = (B^{\mathsf{T}} \otimes I)\text{vec}(A)$,
P.3 $\text{vec}(ABC) = (C^{\mathsf{T}} \otimes A)\text{vec}(B)$,
P.4 $(A \otimes B)^{\mathsf{T}} = A^{\mathsf{T}} \otimes B^{\mathsf{T}}$ and $(A \otimes B)^{\mathsf{H}} = A^{\mathsf{H}} \otimes B^{\mathsf{H}}$,
P.5 $(A \otimes B)(C \otimes D) = (AC \otimes BD)$ and $(A \otimes B)^{+} = A^{+} \otimes B^{+}$,
P.6 $\text{vec}(A \circ B) = \text{vec}(A) \circ \text{vec}(B)$, $(A \circ B)^{\mathsf{T}} = A^{\mathsf{T}} \circ B^{\mathsf{T}}$,
P.7 $\frac{d}{dX}\text{tr}(AX) = A^{\mathsf{T}}$, $\frac{d}{dX}\text{tr}(X^{\mathsf{T}}X) = 2X$, $\frac{d}{dx}(Ax) = A$,
P.8 Let $D_x f$ and $D_x^2 f$ be the gradient and Hessian of $f(x) : \mathbb{R}^n \to \mathbb{R}$. Then, $df = (D_x f)^{\mathsf{T}} dx$ and, $d^2 f = (dx)^{\mathsf{T}}(D_x^2 f)dx$,
P.9 $\text{vec}(A^{\mathsf{T}}) = T_{m,n}\text{vec}(A)$, where $A \in \mathbb{R}^{m\times n}$ and $T_{m,n} \in \{0, 1\}^{mn\times mn}$ is a binary permutation matrix.

2.2. *Sparse stability radius as an optimization problem*

In this subsection we formulate the real, sparse, F-norm SR problem as an equality-constrained optimization problem. Due to space limitations, we present the analysis only for continuous-time systems in this paper. The analysis for discrete-time systems is analogous and can be obtained using a similar procedure. Since the stability region of continuous-time systems is the open left-half complex plane, we have the following definition of the SR.

Definition 1 (*Sparse Stability Radius*)**.** The stability radius of the continuous-time system (1) is given by

$$r_C \triangleq \inf\{\|\Delta\|_F : \alpha(A(\Delta)) \geq 0,\ \Delta \in \mathbf{\Delta}_S \subset \mathbb{R}^{m\times p}\}, \tag{5}$$

where $A(\Delta) = A + B\Delta C$, and $\mathbf{\Delta}_S$ in (3) is the set of sparse perturbations characterized by the structure matrix S. □

We make the following assumption regarding the stability of the nominal system (1).

A1: The matrix A is stable, i.e., $\alpha(A) < 0$.

Assumption **A1** ensures that the SR is strictly greater than zero. Since the eigenvalues of $A(\Delta)$ are a continuous function of Δ, the infimum in (5) is achieved on the imaginary axis of the complex plane (Hinrichsen & Pritchard, 2005). Thus, we have

$$r_C = \min\{\|\Delta\|_F : \alpha(A(\Delta)) = 0,\ \Delta \in \mathbf{\Delta}_S\}. \tag{6}$$

This motivates the reformulation of the sparse SR problem as the following optimization problem:

$$\textbf{SR}: \quad \min_{\Delta\in\mathbb{R}^{m\times p},\, x\in\mathbb{C}^n,\, \omega\in\mathbb{R}} \ \frac{1}{2}\|\Delta\|_F^2 \tag{7}$$

$$\text{s.t.} \quad (A + B\Delta C)x = j\omega x, \tag{7a}$$

$$x^{\mathsf{H}}x = 1, \tag{7b}$$

$$S^c \circ \Delta = 0, \tag{7c}$$

where the eigenvalue–eigenvector constraint (7a) is a reformulation of the spectral constraint in (6) in terms of an eigenvalue–eigenvector pair $(j\omega, x)$. The normalization constraint (7b) is added to ensure uniqueness of the eigenvector (up to unitary scaling). The sparsity constraint (7c) is a reformulation of $\Delta \in \mathbf{\Delta}_S$ (c.f. (3)).

Several remarks are in order for the optimization problem **SR**. First, the eigenvalue–eigenvector constraint (7a) is not convex. As a result, the optimization problem **SR** is not convex, and it may have multiple local minima. This is a typical property of all 2-norm/F-norm, complex/real SR problems, as well as most other minimum distance problems (Kressner & Voigt, 2015).

Second, besides assigning an eigenvalue(s) on the imaginary axis, the equality constraint in (6) also requires the remaining eigenvalues of $A(\Delta)$ to lie in the open left-half complex plane. However, we have omitted this constraint in **SR** because it will be inherently satisfied by the *global* minimum of **SR** due to (i) Assumption **A1**, (ii) the continuity properties of the eigenvalues of $A(\Delta)$, and (iii) the definition of SR in (6). However, a *local* minimum of **SR** need not satisfy this constraint necessarily. Thus, all the local minima $\hat{\Delta}$ of **SR** should be verified against the constraint $\alpha(A(\hat{\Delta})) = 0$ (see Section 5 for an example).

Third, it may be possible that **SR** is non-feasible and there does not exist any Δ that satisfies constraints (7a)–(7c).[1] Such non-feasible cases are universally robust in the sense that no perturbation with the given sparsity structure can make the system unstable, and thus $r_C = \infty$. To avoid such cases, we make the following assumption:

A2: SR is feasible, i.e., there exists at least one perturbation Δ that satisfies (7a)–(7c).

Finally, since $A(\Delta)$ is real, its eigenvalues are symmetric with respect to the real axis. Hence, if $(\hat{\Delta}, \hat{\omega}, \hat{x})$ is a local minimum of (7), then $(\hat{\Delta}, -\hat{\omega}, \hat{x}^*)$ is also a local minimum.

Let $\delta \triangleq \text{vec}(\Delta) \in \mathbb{R}^{mp}$ and let n_s denote the number of nontrivial sparsity constraints (i.e. number of 1's in S^c). Then, (7c) can be vectorized as:

$$\mathsf{S}\delta = 0, \tag{8}$$

where $\mathsf{S} \in \{0, 1\}^{n_s\times mp}$ is a binary matrix given by $\mathsf{S} = [e_{s_1}, e_{s_2}, \dots, e_{s_{n_s}}]^{\mathsf{T}}$ with $\{s_1, \dots, s_{n_s}\} = \text{supp}(\text{vec}(S^c))$ being the set of indices indicating the 1's in $\text{vec}(S^c)$ and e_i being the i^{th} standard basis vector of $\mathbb{R}^{mp}$. Next, we present an algebraic condition to verify Assumption **A2**. We omit the proof, which follows from vectorizing (7a).

Lemma 2.1 (*Feasibility*)**.** *The optimization problem (7) is feasible if and only if there exist $x \in \mathbb{C}^n$ and $\omega \in \mathbb{R}$ satisfying: $(A - j\omega I)x \in$ Range$\left([(Cx)^{\mathsf{T}} \otimes B][I - \mathsf{S}^{+}\mathsf{S}]\right)$, where Range$(\cdot)$ denotes the column space of a matrix over the field of real numbers.*

3. Solution to the optimization problem

In this section we present the optimality conditions for the local solutions of the optimization problem **SR**, and characterize an optimal perturbation. We use the theory of Lagrange multipliers for equality-constrained minimization to derive the optimality conditions. We begin with the following assumption:

A3: Let $(\hat{\Delta}, \hat{\omega}, \hat{x})$ be a local minimum of **SR**. Then, $\hat{\omega} \neq 0$.

Assumption **A3** is necessary to differentiate between the cases when constraint (7a) is complex or real (if $\hat{\omega} = 0$). We present the analysis for the complex case and will discuss the real case later in this section (see Remark 2).

Remark 1 (*Complex Variables*)**.** Since x may be a complex variable, the constraint (7a) also induces the following conjugate constraint:

$$(A + B\Delta C)x^* = -j\omega x^*. \tag{9}$$

We use the formalism wherein a complex number and its conjugate are treated as independent variables (Brandwood, 1983; Hjørungnes & Gesbert, 2007) and, thus, we treat x and x^* as independent variables. □

In the theory of equality-constrained optimization, the first-order optimality conditions are meaningful only when the optimal point satisfies the following regularity condition: the Jacobian of the constraints, defined by J_b, has full rank. This regularity condition is mild and usually satisfied for most classes of problems (Luenberger & Ye, 2008). Before presenting the main result, we derive the Jacobian, which depends on the ordering of the constraints and the variables. Recalling Remark 1, let $z \triangleq [x^{\mathsf{T}}, x^{\mathsf{H}}, \delta^{\mathsf{T}}, \omega]^{\mathsf{T}}$ be the vector containing all the variables of optimization problem **SR**.

[1] A trivial example is: $A = \left[\begin{smallmatrix} -1 & 2 \\ 0 & -2 \end{smallmatrix}\right]$, $B = C = I_2$ and $S = \left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right]$. In this case, since only A_{12} is allowed to be perturbed, the eigenvalues of $A(\Delta)$ will always be $\{-1, -2\}$ and cannot lie on the imaginary axis.

Lemma 3.1 (*Jacobian of the Constraints*)**.** *The Jacobian of the equality constraints* (7a), (9), (7b), (8) *is*

$$J_b(z)=\begin{bmatrix} A(\Delta)-j\omega I & 0 & (Cx)^{\mathsf{T}}\otimes B & -jx\\ 0 & A(\Delta)+j\omega I & (Cx^*)^{\mathsf{T}}\otimes B & jx^*\\ x^{\mathsf{H}} & x^{\mathsf{T}} & 0 & 0\\ 0 & 0 & \mathsf{S} & 0 \end{bmatrix}.$$

Proof. We construct the Jacobian J_b by taking the derivatives of the constraints (7a), (9), (7b), and (8) with respect to z. Using P.3, constraint (7a) can be written as

$$(A-j\omega I)x+[(Cx)^{\mathsf{T}}\otimes B]\delta=0. \qquad (10)$$

Differentiating (7a) w.r.t. x,ω and (10) w.r.t δ yields the first (block) row of J_b. Similar derivatives of the conjugate constraint (9) w.r.t. z yields the second (block) row of J_b. Differentiating constraint (7b) w.r.t. x and x^* yields the third (block) row of J_b. Finally, differentiating (8) w.r.t. z yields the last (block) row of J_b. □

Next, we provide the local optimality conditions for the optimization problem **SR**.

Theorem 3.2 (*Optimality Conditions*)**.** *Let* $(\hat{\Delta},\hat{x},\hat{\omega})$ *(equivalently* $\hat{z}=[\hat{x}^{\mathsf{T}},\hat{x}^{\mathsf{H}},\hat{\delta}^{\mathsf{T}},\hat{\omega}]^{\mathsf{T}}$*) satisfy the constraints* (7a)–(7c), *and let* $\hat{l}$ *denote the left eigenvector of* $A(\hat{\Delta})$ *corresponding to the eigenvalue* $j\hat{\omega}$*. Let* $J_b(z)$ *be defined in* Lemma 3.1, *and let* $P(z)=I-J_b^+(z)J_b(z)$*. Further, let* $\hat{N}\triangleq C\otimes(B^{\mathsf{T}}\hat{l})$*, and let*

$$\hat{D}\triangleq\begin{bmatrix} 0 & 0 & \hat{N}^{\mathsf{H}} & j\hat{l}^*\\ 0 & 0 & \hat{N}^{\mathsf{T}} & -j\hat{l}\\ \hat{N} & \hat{N}^* & 2I & 0\\ -j\hat{l}^{\mathsf{T}} & j\hat{l}^{\mathsf{H}} & 0 & 0 \end{bmatrix}. \qquad (11)$$

Then, $(\hat{\Delta},\hat{x},\hat{\omega})$ *is a local minimum of the optimization problem* ***SR*** *if and only if*

$$\hat{\Delta}=-S\circ\left[B^{\mathsf{T}}\mathrm{Re}(\hat{l}\hat{x}^{\mathsf{T}})C^{\mathsf{T}}\right], \qquad (11a)$$

$$(A+B\hat{\Delta}C)\hat{x}=j\hat{\omega}\hat{x}, \qquad (11b)$$

$$(A+B\hat{\Delta}C)^{\mathsf{T}}\hat{l}=j\hat{\omega}\hat{l}, \qquad (11c)$$

$$\mathrm{Im}(\hat{l}^{\mathsf{T}}\hat{x})=0, \qquad (11d)$$

$$J_b(\hat{z})\ \textit{has full rank}, \qquad (11e)$$

$$P(\hat{z})\hat{D}P(\hat{z})>0. \qquad (11f)$$

Proof. We prove the result using the Lagrange multiplier method for equality-constrained minimization. Let $l\in\mathbb{C}^n$, l^*, $h\in\mathbb{R}$ and $M\in\mathbb{R}^{m\times p}$ be the Lagrange multipliers associated with constraints (7a), (9), (7b) and (7c), respectively.[2] The Lagrangian function is given by

$$\mathcal{L}\overset{\text{P.1}}{=}\frac{1}{2}\,\mathrm{tr}(\Delta^{\mathsf{T}}\Delta)+h(x^{\mathsf{H}}x-1)+\frac{1}{2}l^{\mathsf{T}}(A+B\Delta C-j\omega I)x$$
$$+\frac{1}{2}l^{\mathsf{H}}(A+B\Delta C+j\omega I)x^*+\underbrace{1_m^{\mathsf{T}}[M\circ(S^c\circ\Delta)]1_p}_{=\mathrm{tr}[(M\circ S^c)^{\mathsf{T}}\Delta]}.$$

Part 1 — First-order necessary conditions (11a)–(11e)*:* Differentiating $\mathcal{L}$ w.r.t. x and setting to 0, we get

$$\frac{d}{dx}\mathcal{L}\overset{\text{P.7}}{=}\frac{1}{2}l^{\mathsf{T}}(A+B\Delta C-j\omega I)+hx^{\mathsf{H}}=0 \qquad (12)$$

$$\Rightarrow\frac{1}{2}l^{\mathsf{T}}(A+B\Delta C-j\omega I)x+hx^{\mathsf{H}}x=0\overset{(7a),(7b)}{\Longrightarrow}h=0,$$

and thus, from (12), we get (11c). Eq. (11b) is a restatement of (7a) for the optimal $(\hat{\Delta},\hat{x},\hat{\omega})$. Differentiating $\mathcal{L}$ w.r.t. Δ and setting to 0, we get

$$\frac{d}{d\Delta}\mathcal{L}\overset{\text{P.7}}{=}\Delta+\mathrm{Re}(B^{\mathsf{T}}lx^{\mathsf{T}}C^{\mathsf{T}})+M\circ S^c=0. \qquad (13)$$

Taking the Hadamard product of (13) with S^c, and using (7c) and $S^c\circ S^c=S^c$, we get

$$S^c\circ\mathrm{Re}(B^{\mathsf{T}}lx^{\mathsf{T}}C^{\mathsf{T}})+M\circ S^c=0. \qquad (14)$$

Replacing $M\circ S^c$ from (14) in (13), we get (11a). Finally, differentiating $\mathcal{L}$ w.r.t. ω and setting to 0, we get

$$\frac{d}{d\omega}\mathcal{L}=\frac{1}{2}jl^{\mathsf{H}}x^*-\frac{1}{2}jl^{\mathsf{T}}x=\mathrm{Im}(l^{\mathsf{T}}x)=0.$$

Eq. (11e) is the necessary regularity condition and follows from Lemma 3.1.

Part 2 — Second-order sufficient condition (11f)*:* We compute the Hessian (denoted by D) of $\mathcal{L}$ w.r.t. z by expressing the second-order differential of $\mathcal{L}$ as $d^2\mathcal{L}=dz^{\mathsf{T}}Ddz$ (c.f. P.8). Taking differential of $\mathcal{L}$ twice, we get

$$d^2\mathcal{L}=\mathrm{tr}[(d\Delta)^{\mathsf{T}}d\Delta]+l^{\mathsf{T}}B(d\Delta)Cdx-j(d\omega)l^{\mathsf{T}}(dx)$$
$$+\ l^{\mathsf{H}}B(d\Delta)C(dx^*)+j(d\omega)l^{\mathsf{H}}(dx^*)+2h(dx)^{\mathsf{H}}(dx),$$
$$\overset{\text{P.1},h=0}{=}(d\delta)^{\mathsf{T}}d\delta+\mathrm{vec}\big[C^{\mathsf{T}}(d\Delta)^{\mathsf{T}}B^{\mathsf{T}}l\big]^{\mathsf{T}}(dx)-j(d\omega)l^{\mathsf{T}}(dx)$$
$$+\ \mathrm{vec}\big[C^{\mathsf{T}}(d\Delta)^{\mathsf{T}}B^{\mathsf{T}}l^*\big]^{\mathsf{T}}(dx^*)+j(d\omega)l^{\mathsf{H}}(dx^*),$$
$$\overset{\text{P.2,P.4}}{=}(d\delta)^{\mathsf{T}}d\delta+(d\delta)^{\mathsf{T}}N(dx)-j(d\omega)l^{\mathsf{T}}(dx)$$
$$+\ (d\delta)^{\mathsf{T}}N^*(dx^*)+j(d\omega)l^{\mathsf{H}}(dx^*)=\frac{1}{2}(dz)^{\mathsf{T}}D(dz),$$

where D is the Hessian defined in (11) and $N = C\otimes(B^{\mathsf{T}}l)$. The sufficient second-order optimality condition requires Hessian to be positive-definite in the kernel of the Jacobian at optimal point (Luenberger & Ye, 2008). That is, $y^{\mathsf{T}}Dy>0,\ \forall y:J_b(z)y=0$. This condition is equivalent to $P^{\mathsf{T}}(z)DP(z)>0$, since $J_b(z)y=0$ iff $y=P(z)s$ for a complex s, where $P(z)$ is the projection matrix onto the null space of $J_b(z)$ (Luenberger & Ye, 2008). Since the projection matrix $P(z)$ is symmetric, (11f) follows. □

The local optimality conditions in Theorem 3.2 reveal the inherent properties of an optimal perturbation and the stability radius. First, (11a) presents the explicit relations between the optimal perturbation $\hat{\Delta}$ and left and right eigenvectors of the optimally perturbed matrix $A(\hat{\Delta})$. Second, (11d) shows that the inner product of the left-conjugate and right eigenvectors of the optimal perturbation is always real. Third, notice that the optimal perturbation in (11a) always satisfies the sparsity constraint (7c) (since $S\circ S^c=0$). The optimality condition (11a) can also be re-written as $\hat{\Delta}=-S\circ[B^{\mathsf{T}}\hat{L}\hat{X}^{\mathsf{T}}C^{\mathsf{T}}]$, where $\hat{L}\triangleq[\mathrm{Re}(\hat{l}),-\mathrm{Im}(\hat{l})]$ and $\hat{X}\triangleq[\mathrm{Re}(\hat{x}),\mathrm{Im}(\hat{x})]$. This shows that, although $\mathrm{rank}(B^{\mathsf{T}}\hat{L}\hat{X}^{\mathsf{T}}C^{\mathsf{T}})\le 2$, the rank of $\hat{\Delta}$ can be greater than 2. In contrast, the optimal perturbation for real, *non-sparse*, 2-norm/F-norm SR always has rank less that or equal to 2 (Qiu et al., 1995).

Remark 2 (*Optimality at* $\hat{\omega}=0$)**.** If $\hat{\omega}=0$ (Assumption **A3** is violated), a similar analysis can be performed to compute the optimality conditions by using the following real eigenvalue assignment equation instead of (7a)

$$(A+B\Delta C)x=0, \qquad (15)$$

where $x\in\mathbb{R}^n$. □

[2] The multipliers associated with (7a) and (9) are conjugate to each other to make the Lagrangian real (see Brandwood, 1983, Section 4).

Next, we present a brief comparison of our optimality conditions with those in Johnson et al. (2018). Since Johnson et al. (2018) considers perturbations of the form $A - \Delta$, we assume $B = -C = -I$. In the current setting, condition (1a) of Theorem 1 in Johnson et al. (2018) becomes: $\hat{\Delta}$ is a real matrix that satisfies the sparsity constraints (7c) and is the minimum-norm matrix that minimizes $\|\hat{\bar{l}}^{\mathsf{T}}\Delta\hat{x} - \bar{l}^{\mathsf{T}}(A - j\hat{\omega}I)\hat{x}\|_F$. Further, condition (1b) in Johnson et al. (2018) can be stated as follows: $\hat{d\omega} = 0$, where $\hat{d\omega}$ minimizes $\|\bar{D}(d\omega) - \hat{\Delta}\|_F$ and $\bar{D}(d\omega)$ is a real matrix that satisfies the sparsity constraints (7c) and is the minimum-norm matrix that minimizes $\|\hat{\bar{l}}^{\mathsf{T}}D\hat{x} - jd\omega\bar{l}^{\mathsf{T}}\hat{x}\|_F$. We can observe that the above conditions are implicit and involve solving optimization problems, whereas our conditions (11a)–(11d) are more explicit.

The solution of the optimization problem **SR** can be obtained by numerically/iteratively solving the optimality equations (11a)–(11d) using any non-linear equation solving technique. The regularity and local minimum property of the solution can be verified using (11e) and (11f), respectively. Finally, the local minimum should be verified against $\alpha(A(\hat{\Delta})) = 0$. Since the optimization problem is not convex, only local minima can be obtained via this procedure. To improve upon the local solutions and to capture the global minimum, the above procedure can be repeated for different initial conditions. Clearly, finding the global minimum is not guaranteed in all cases. In this case, the procedure provides an upper bound to the SR.

Instead of solving the optimality equations, we use a penalty based approach using gradient descent to obtain the local solutions. Details of this approach and the corresponding algorithm are provided in the next section.

4. Gradient based solution algorithm

In this section, we present an iterative gradient based algorithm to obtain a local solution to optimization problem **SR**. We use the penalty based optimization approach and the Sylvester equation based parametrization to convert the constrained optimization problem (7) into an unconstrained optimization problem. Note that we ignore the eigenvector normalization constraints (7b) hereafter.

Using the penalty based optimization approach (Luenberger & Ye, 2008), we modify the cost function to include a penalty when the sparsity constraints are violated. The penalty is imposed by weighing individual entries of the perturbation using a weighing matrix $W \in \mathbb{R}^{m\times p}$ given by

$$W_{ij} = \begin{cases} 1 & \text{if} \quad S_{ij} = 1, \text{ and} \\ \mathbf{w} \gg 1 & \text{if} \quad S_{ij} = 0. \end{cases}$$

Using the weighing matrix W, the penalized cost is $J_W = \frac{1}{2}\|W \circ \Delta\|_F^2$. Next, we reformulate (7a) as a purely real constraint. Let $X \triangleq [\mathrm{Re}(x), \mathrm{Im}(x)] \in \mathbb{R}^{n\times 2}$. Then, (7a) is equivalent to

$$(A + B\Delta C)X = \omega X\bar{I}, \tag{16}$$

where $\bar{I} \triangleq \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$. Next, we use the Sylvester equation based parametrization (Bhattacharyya & De Souza, 1982) and define $G \triangleq \Delta CX \in \mathbb{R}^{m\times 2}$. It follows that (16) is equivalent to

$$AX - \omega X\bar{I} = -BG, \tag{17a}$$

$$G = \Delta CX. \tag{17b}$$

Note that (17a) is a Sylvester equation in X. Due to Assumption **A1**, $\Gamma(A) \cap \Gamma(-\omega\bar{I}) = \emptyset$ and, thus, (17a) has a unique solution for any given (G, ω). Further, for any (G, ω), (17b) has a solution if $CX \in \mathbb{R}^{p\times 2}$ has rank two. Thus, we make the following assumption.

A4: For any (G, ω), CX has full column rank, where X is the unique solution of (17a). This requires $p \geq 2$.

For a given (G, ω) and under Assumption **A4**, we can solve (17b) to obtain Δ, which, in general, may not be unique. Since we aim to minimize the norm of the weighted perturbation $(W \circ \Delta)$, given a (G, ω), we choose Δ as the solution of the following quadratic optimization problem:

$$\begin{aligned} \min_{\Delta\in\mathbb{R}^{m\times p}} \quad & J_W = \frac{1}{2}\|W \circ \Delta\|_F^2 \\ \text{s.t.} \quad & \text{(17b) holds.} \end{aligned} \tag{18}$$

Because the entries of W are positive, the optimization problem (18) is convex. Further, under Assumption **A4**, (18) has a unique global minimum (see the proof of Lemma 4.1 in the Appendix for more details), which can be obtained by solving the following Lagrange conditions:

$$W \circ W \circ \Delta + \Lambda(CX)^{\mathsf{T}} = 0, \tag{19a}$$

$$\Delta CX = G, \tag{19b}$$

where $\Lambda \in \mathbb{R}^{m\times 2}$ is the Lagrange multiplier of the problem (18). To summarize, using the Sylvester equation based parametrization, we can freely vary (G, ω) (under Assumption **A4**) and compute the corresponding unique X using (17a) and unique Δ using (19a)–(19b).

Thus, the constrained optimization problem (7) can be reformulated as the following optimization problem in free variables G, ω:

$$\begin{aligned} \min_{G\in\mathbb{R}^{m\times 2},\, \omega\in\mathbb{R}} \quad & J_W = \frac{1}{2}\|W \circ \Delta\|_F^2 \\ \text{s.t.} \quad & \text{(17a), (19a) and (19b) hold.} \end{aligned} \tag{20}$$

We aim to solve problem (20) using a gradient descent approach. The next result provides analytical expressions for the gradient and Hessian of the cost in (20). Let $g \triangleq \mathrm{vec}(G) \in \mathbb{R}^{2m}$, $x_v = \mathrm{vec}(X) \in \mathbb{R}^{2n}$, $\lambda = \mathrm{vec}(\Lambda) \in \mathbb{R}^{2m}$, and let the free variables of (20) be denoted by $\bar{z} \triangleq [g^{\mathsf{T}}, \omega]^{\mathsf{T}}$. Further, let $e_{2m+1} = [0, \ldots, 0, 1]^{\mathsf{T}} \in \mathbb{R}^{2m+1}$ and $\bar{W} \triangleq \mathrm{diag}(\mathrm{vec}(W \circ W))$.

Lemma 4.1 (*Gradient and Hessian*). *Define the following Kronecker products:*

$$\tilde{B} = I_2 \otimes B, \qquad \tilde{I} = \bar{I} \otimes I_n, \qquad \tilde{A}(\omega) = I_2 \otimes A + \omega\tilde{I},$$

$$\tilde{\Delta} = I_2 \otimes (\Delta C), \quad \tilde{X} = (CX)^{\mathsf{T}} \otimes I_m, \quad \bar{X} = \begin{bmatrix} \bar{W} & \tilde{X}^{\mathsf{T}} \\ \tilde{X} & 0 \end{bmatrix},$$

and $E = \begin{bmatrix} (C \otimes \Lambda)T_{n,2} \\ \tilde{\Delta} \end{bmatrix}$. *Let* $U = \begin{bmatrix} U_1 & U_2 \end{bmatrix}$ *and* V *be the unique solutions of, respectively,*

$$U\bar{X} = \begin{bmatrix} I_{mp} & 0 \end{bmatrix}, \quad \text{and} \quad \tilde{A}(\omega)V = \begin{bmatrix} \tilde{B} & \tilde{I}x_v \end{bmatrix}.$$

Further, let $\bar{U}_1 \in \mathbb{R}^{m\times p}$ *and* $\bar{U}_2 \in \mathbb{R}^{m\times 2}$ *be such that* $\mathrm{vec}(\bar{U}_1) = U_1^{\mathsf{T}}\bar{W}\delta$ *and* $\mathrm{vec}(\bar{U}_2) = U_2^{\mathsf{T}}\bar{W}\delta$. *Define* $Y = \left[\begin{smallmatrix} 0 & 0 \\ I_{2m} & 0 \end{smallmatrix}\right] + EV$, $Z = (UY)^{\mathsf{T}}$, *and*

$$\begin{aligned} M =\, & V^{\mathsf{T}}\begin{bmatrix} (\bar{U}_2^{\mathsf{T}} \otimes C^{\mathsf{T}})T_{m,p} & T_{2,n}(C^{\mathsf{T}}\bar{U}_1^{\mathsf{T}} \otimes I_2)T_{m,2} \end{bmatrix}\bar{X}^{-1}Y \\ & - V^{\mathsf{T}}\tilde{I}^{\mathsf{T}}\tilde{A}(\omega)^{-\mathsf{T}}E^{\mathsf{T}}U^{\mathsf{T}}\bar{W}\delta e_{2m+1}^{\mathsf{T}}. \end{aligned}$$

Then, the gradient and Hessian of J_W *in* (20) *satisfy*

$$\frac{dJ_W}{d\bar{z}} = Z\bar{W}\,\delta, \quad \text{and} \tag{21}$$

$$\frac{d^2J_W}{d^2\bar{z}} \triangleq H = Z\bar{W}Z^{\mathsf{T}} + M + M^{\mathsf{T}}. \tag{22}$$

Proof. See the Appendix.

Using Lemma 4.1, we present a gradient/Newton descent Algorithm 1 to solve the optimization problem (20).

Algorithm 1: Gradient/Newton descent for **SR**

```
Input: $A, B, C, W, g_0, \omega_0$.
Output: Local minimum $(\Delta, X, \omega)$ of (20).
Initialize: $\bar{z}_0 = \left[g_0^{\mathsf{T}}, \omega_0\right]^{\mathsf{T}}$, $(x_0, \delta_0) \leftarrow$ See line 4-5
repeat
1    $\beta \leftarrow$ Update step size (see below) ;
2    $\bar{z} \leftarrow \bar{z} - \beta Z \bar{W} \delta$    or ;
3    $\bar{z} \leftarrow \bar{z} - \beta d$,  where $(H + K)d = Z\bar{W}\delta$ ;
4    $x \leftarrow$ Vectorization of solution of (17a) ;
5    $\delta \leftarrow$ Vectorization of solution of (19a)–(19b) ;
until convergence;
return $(\Delta, X, \omega)$
```

Steps 2 and 3 of Algorithm 1 represent gradient and damped Newton descent steps, respectively. In the Newton descent step, the Hessian H is required to be positive-definite. To satisfy this property, we add the term $K = \epsilon I - M - M^{\mathsf{T}}$ to the Hessian with $0 < \epsilon \ll 1$ (Luenberger & Ye, 2008). Further, the step size β can be updated using backtracking line search or Armijo's rule (Luenberger & Ye, 2008). In general, the Newton descent converges faster as compared to gradient descent. For a detailed discussion of the two algorithms, the interested reader is referred to Luenberger and Ye (2008). Finally, if Assumption **A4** is not satisfied in any iteration of Algorithm 1, (i.e., CX does not have full column rank), then we can slightly modify (g, ω) to ensure that **A4** is satisfied and continue the iterations.

Since the modified Hessian is $H + K = Z\bar{W}Z^{\mathsf{T}} + \epsilon I$, the matrix M in (22) need not be computed in every iteration. The computational complexity of step 2 is dictated by solving Sylvester equations in Lemma 4.1 to obtain U and V, and is given by $O(n^3 + m^3p^3)$. The complexity of step 3 is same as step 2, since the complexity to compute d is $O(m^3)$. The complexity of step 4 is $O(n^3)$. In step 5, δ (and λ) can be obtained by vectorizing (19a)–(19b) (see (26) in the Appendix), and its complexity is $O(m^3p^3)$.

Remark 3 (*Algorithm when $\hat{\omega} = 0$*). The perturbation Δ computed at each iteration of Algorithm 1 assigns two eigenvalues of $A(\Delta)$ at $\pm j\omega$. As a consequence, in cases where $\hat{\omega} = 0$ is a local minimum of (20), the algorithm converges to a local minimum $\hat{\Delta}$ such that $A(\hat{\Delta})$ has eigenvalue 0 with multiplicity two. This may not be a desired local minimum since instability results from at least one (and not necessarily two) eigenvalue of $A(\hat{\Delta})$ being at the origin. In this case, we can use (15) instead of (16) to develop an analogous algorithm. □

Remark 4 (*Choice of Weights*). As the weight $\mathbf{w}$ increases, an optimal solution of (20) satisfies the sparsity constraints (7c) with increasing accuracy. However, an increase in the weights also reduces the convergence speed of Algorithm 1. Thus, there exists a trade-off between the accuracy of the sparse solutions and the convergence time of the algorithm. □

Remark 5 (*Alternative Optimization Techniques*). We emphasize that more sophisticated and efficient optimization procedures to solve the **SR** problem can be used. For instance, alternative choices of the Hessian modification term V that result in better conditioning can be used. Quasi-Newton methods like BFGS (Luenberger & Ye, 2008) to compute approximate Hessian using the expressions in Lemma 4.1 can also be employed. Augmented Lagrangian methods that provide accurate solutions can be explored. Software packages like GRANSO (Curtis, Mitchell, & Overton, 2017) and low-rank solvers for Sylvester equations (Benner, Kohler, & Saak, 2011) can be used for large-scale problems. A detailed study of these methods is not the primary focus of this paper, and is left as the subject of future research.

5. Simulation studies

In this section, we present numerical simulation studies of our algorithm. We perform the simulations using MATLAB R2019a installed on Macbook Pro 2019 with Intel Core i5 processor and 16 GB of RAM. To begin, we consider the following example from Qiu et al. (1995):

$$A = \begin{bmatrix} 79 & 20 & -30 & -20 \\ -41 & -12 & 17 & 13 \\ 167 & 40 & -60 & -38 \\ 33.5 & 9 & -14.5 & -11 \end{bmatrix}, B = \begin{bmatrix} 0.2190 & 0.9347 \\ 0.0470 & 0.3835 \\ 0.6789 & 0.5194 \\ 0.6793 & 0.8310 \end{bmatrix},$$

$$C = \begin{bmatrix} 0.0346 & 0.5297 & 0.0077 & 0.0668 \\ 0.0535 & 0.6711 & 0.3848 & 0.4175 \end{bmatrix}.$$

The eigenvalues of A are $\{-1 \pm j, -1 \pm 10j\}$. We consider two cases:
Case 1: No sparsity constraints, i.e., $S = \left[\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right]$,
Case 2: Only the diagonal entries of Δ are allowed to be perturbed, i.e., $S = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$.

The weight in the weighing matrix W is chosen as $\mathbf{w} = 100$. Table 1 shows the local minima of optimization problem (7) obtained by Algorithm 1 for both the cases. The first local minimum is also the global minimum for both the cases. The initial conditions are randomly selected as $\omega_{0,gl} = 2.5$ and $G_{0,gl} = \begin{bmatrix} 1.0582 & 1.4115 \\ 0.4363 & -0.0146 \end{bmatrix}$ for the global minima for both cases, and as $\omega_{0,lo} = 1$ and $G_{0,lo} = \begin{bmatrix} 0.5201 & -0.0348 \\ -0.0200 & -0.7982 \end{bmatrix}$ for the local minima for both cases. The stopping criteria for all simulations in this section are $\|\frac{dJ_W}{d\bar{z}}\|_F = \|Z\bar{W}\delta\|_F < 10^{-5}$ and the maximum number of iterations is set at 1000. Note that the second local minimum for case 1 satisfies $\alpha(A(\hat{\Delta}^{(2)})) = 0$, whereas the second local minimum for case 2 does not satisfy this constraint (c.f. discussion after (7c)). Thus, it is not a valid local minimum.

Next, we illustrate the relation between the local minima of our optimization problem and the geometry of the spectral value sets. Spectral value set captures the region in which all possible eigenvalues of the perturbed system matrices can lie, and for $\eta \geq 0$, is defined as:

$$\mathcal{S}_\eta \triangleq \{\Gamma(A + B\Delta C) : \|\Delta\|_F \leq \eta, \Delta \in \mathbf{\Delta}_S\}.$$

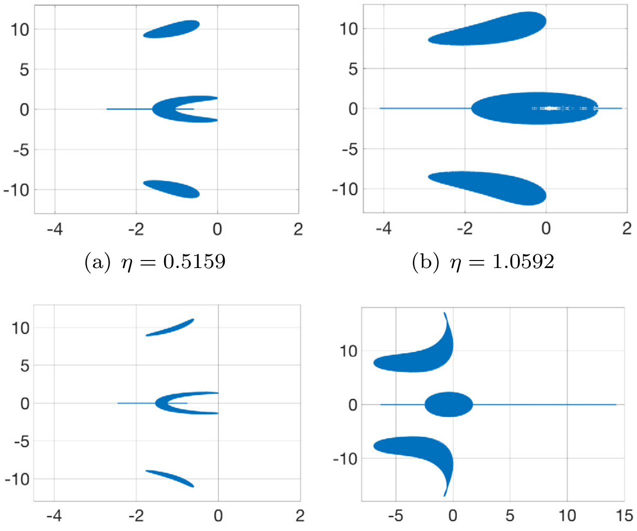
Fig. 1 shows the spectral value sets in the complex plane corresponding to the local minima in Table 1. We observe that the local minima are precisely the cases when the *locally* right-most points of the spectral value sets[3] intersect with the imaginary axis.

Fig. 2 presents a sample run of Algorithm 1 for the case 2 of the above example. It is initialized with $\omega_{0,gl}$ and $G_{0,gl}$ and takes 24 iterations to converge to the global minimum using the Newton descent steps. Fig. 2 shows the penalized cost (scaled), spectral abscissa of the perturbed matrix $A(\Delta)$, and ω at each iteration. Observe that, at the start of the algorithm, $\alpha(A(\Delta)) > 0$ indicating that $A(\Delta)$ has two eigenvalues in the right-half complex plane (the other two are at $\pm j\omega$). As iterations progress, these unstable eigenvalues move towards the left-half plane and, at the global minimum, all eigenvalues are in closed left-half plane (c.f. discussion after (7c)). Further, the optimization cost decreases monotonically during the iterations.

[3] $\mathcal{S}_\eta$ for this example was visualized by performing an exhaustive search over $\Delta = \left[\begin{smallmatrix} \Delta_{11} & 0 \\ 0 & \Delta_{22} \end{smallmatrix}\right]$ such that $\Delta_{11}^2 + \Delta_{22}^2 \leq \eta^2$, and plotting $\Gamma(A(\Delta))$.

Table 1
Minima obtained via Algorithm 1.

Case 1	Case 2
$\hat{\Delta}_1 = \begin{bmatrix} -0.0332 & -0.0717 \\ 0.1975 & 0.4700 \end{bmatrix}$	$\hat{\Delta}_1 = \begin{bmatrix} -0.0418 & 0.0000 \\ 0.0000 & 0.5638 \end{bmatrix}$
$r_C = \lVert\hat{\Delta}_1\rVert_F = 0.5159$	$r_C = \lVert\hat{\Delta}_1\rVert_F = 0.5653$
$\hat{\omega}_1 = 1.3753$	$\hat{\omega}_1 = 1.3365$
$\hat{x}_1 = \begin{bmatrix} 0.1340 - 0.0022j \\ 0.3692 + 0.0456j \\ 0.7733 + 0.2579j \\ -0.2504 - 0.3411j \end{bmatrix}$	$\hat{x}_1 = \begin{bmatrix} 0.0905 - 0.0971j \\ 0.2152 - 0.3108j \\ 0.3295 - 0.7459j \\ 0.0799 + 0.4099j \end{bmatrix}$
$\hat{l}_1 = \begin{bmatrix} -1.3796 + 0.4056j \\ -0.5825 - 0.1855j \\ 0.4576 - 0.1326j \\ 0.2771 - 0.4659j \end{bmatrix}$	$\hat{l}_1 = \begin{bmatrix} -0.7660 - 1.5362j \\ -0.6177 - 0.3611j \\ 0.2590 + 0.5098j \\ -0.1785 + 0.6099j \end{bmatrix}$
Time (ms.) = 42.1	Time (ms.) = 45.2
$\hat{\Delta}_2 = \begin{bmatrix} 0.1841 & 0.5173 \\ -0.8050 & -0.4151 \end{bmatrix}$	$\hat{\Delta}_2 = \begin{bmatrix} 4.8818 & 0.0000 \\ 0.0000 & -0.8898 \end{bmatrix}$
$\lVert\hat{\Delta}_2\rVert_F = 1.0592$	$\lVert\hat{\Delta}_2\rVert_F = 4.9622$
$\hat{\omega}_2 = 10.8758$	$\hat{\omega}_2 = 11.0790$
$\hat{x}_2 = \begin{bmatrix} 0.2032 + 0.3252j \\ 0.0505 - 0.2331j \\ 0.7184 + 0.4695j \\ -0.0532 + 0.2381j \end{bmatrix}$	$\hat{x}_2 = \begin{bmatrix} -0.1927 - 0.3320j \\ 0.2084 + 0.0795j \\ -0.0770 - 0.8566j \\ -0.2483 - 0.0397j \end{bmatrix}$
$\hat{l}_2 = \begin{bmatrix} 8.4752 + 9.7446j \\ 1.8464 + 2.4744j \\ -3.6880 - 3.0693j \\ -2.4960 - 1.8877j \end{bmatrix}$	$\hat{l}_2 = \begin{bmatrix} 435.71 + 76.99j \\ 112.50 - 0.98j \\ -153.99 - 56.91j \\ 95.93 - 42.90j \end{bmatrix}$
Time (ms.) = 19.2	Time (ms.) = 24.7



(a) $\eta = 0.5159$ (b) $\eta = 1.0592$

(c) $\eta = 0.5653$ (d) $\eta = 4.9622$

Fig. 1. The spectral value sets corresponding to the local minima in Table 1. Figures (a)–(b) correspond to Case 1, and (c)–(d) correspond to Case 2.

Next, we present a comparison of the global minimum obtained by Algorithm 1 for different penalty weights. Table 2 shows the global minima for three values of weight **w**. Further, Fig. 3 shows the sparsity error $E \triangleq \lVert\Delta - S \circ \Delta\rVert_F$ and norm of the optimal perturbation as a function of weight **w**. The algorithm is initialized with $\omega_{0,gl}$ and $G_{0,gl}$ for all weights. Observe that as the weight **w** increases, the sparsity error decreases and optimal perturbations become more sparse. Furthermore, the norm of the optimal perturbations increases with **w**, since a larger weight implies a tighter constraint on the perturbation entries.

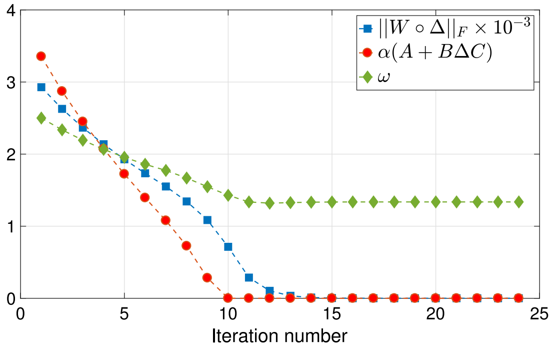
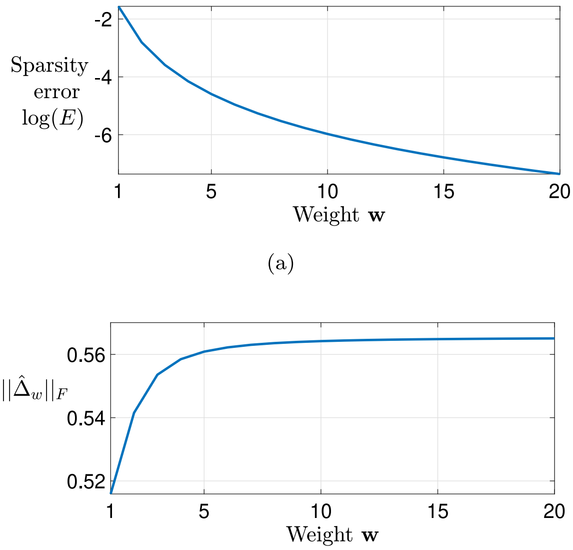


Fig. 2. A sample iteration run of Algorithm 1.



(a)

(b)

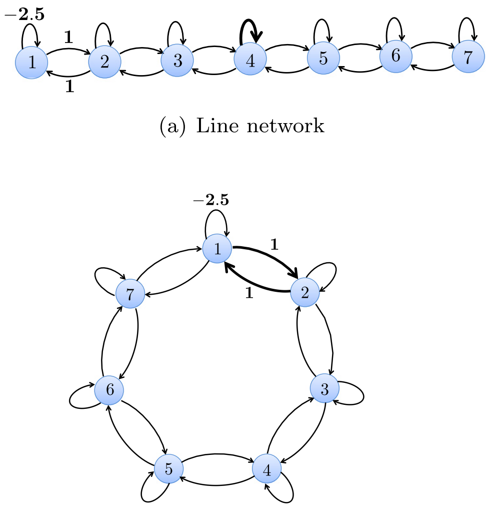
Fig. 3. Variation of (a) sparsity error log(E), and (b) norm of the optimal perturbation $\hat{\Delta}$, as a function of weight **w**.

Table 2
Approximately-sparse solutions.

w	$\hat{\Delta}$	$\lVert\hat{\Delta}\rVert_F$	$\hat{\omega}$	Time (ms)
5	$\begin{bmatrix} -0.0414 & -0.0036 \\ 0.0095 & 0.5593 \end{bmatrix}$	0.5609	1.3385	28.1
10	$\begin{bmatrix} -0.0417 & -0.0009 \\ 0.0024 & 0.5627 \end{bmatrix}$	0.5642	1.3370	34.5
20	$\begin{bmatrix} -0.0418 & -0.0002 \\ 0.0006 & 0.5635 \end{bmatrix}$	0.5651	1.3367	40.1

Finally, we illustrate that our sparse SR framework provides structural insights into the stability of dynamical networks. We consider symmetric line and circular networks as shown in Fig. 4, where the nodes represent the scalar states and the edges represent the non-zero couplings. All self loops have weight −2.5 and all inter-node edges have weight 1. The weighted incidence matrix A of a network can be easily constructed using these weights, and it is stable.

We are interested in identifying the edge(s) that are most critical for the stability of the network. This can be characterized by assigning a sparsity pattern corresponding to a subset of edges that are perturbed, and computing the sparse SR using the developed framework. Then, the most critical edge set is the one



(a) Line network

(b) Circular network

Fig. 4. Two symmetric networks. The bold edge(s) is/are most critical and result in the smallest sparse SR.

which results in the least SR. For the line network, we allow only a singe edge to be perturbed. It implies that only one entry of Δ is allowed to be perturbed. For the circular network, we allow for two inter-node edges to be perturbed (self loop edges are fixed). This implies that only two non-diagonal entries of Δ are allowed to be perturbed. We set $B = C = I$ for both the networks.

For the line network, we observe that the most critical edge is the self loop of the node in the center of the line (node 4).[4] The SR corresponding to this edge is 1.5118. For the circular network, the two most critical inter-node edges are the edges between any two neighboring nodes. The optimal perturbations for the two edges are 0.9724 and 0.9814 (in any order) and the corresponding SR is 1.3816. Due to the circular symmetry, there exist 7 pairs of critical edges in the network. These examples highlight that our sparse SR framework is useful in studying the robust stability of sparse networks, which was not possible using the previous non-sparse SR theory.

6. Conclusion

In this paper we study the real, sparse, F-norm stability radius of a linear time-invariant system, which measures its ability to maintain stability in the presence of structured additive perturbations. We formulate the stability radius problem as an equality-constrained minimization problem, and characterize its optimality conditions. These conditions reveal important geometric properties of the stability radius and the associated perturbation, and allow us to design a penalty based Newton descent algorithm that provably converges to locally optimal values of the stability radius and the associated perturbation. Using the Frobenius norm to measure the size of perturbations is not only convenient for the analysis, but it also provides selective information regarding which system entries have a greater effect on system stability. Further, imposing an arbitrary sparsity pattern to the perturbation becomes crucial when studying the stability radius of network systems and, more generally, systems where only a subset of the entries can be perturbed. Numerical examples are shown to highlight the utility of our framework for characterizing structural fragility of networks. Future research directions include the exploration of more efficient, accurate and reliable optimization procedures, especially for large-scale systems.

Appendix. Proof of Lemma 4.1

Part 1 — Gradient: We derive the gradient by expressing the first order differential of J_W in terms of the differential of free variables $d\bar{z}$. Using $J_W \overset{\text{P.1,P.6}}{=} \frac{1}{2}(\text{vec}(W)\circ\delta)^{\mathsf{T}}(\text{vec}(W)\circ\delta) = \frac{1}{2}\delta^{\mathsf{T}}\bar{W}\delta$, we get

$$dJ_W = \delta^{\mathsf{T}}\bar{W}(d\delta). \tag{23}$$

Next, we derive an expression for $d\delta$. Taking the differential of (17a) and vectorizing (using $\bar{I}^{\mathsf{T}} = -\bar{I}$) yields

$$\begin{aligned} &A(dX) - \omega(dX)\bar{I} - (d\omega)X\bar{I} = -B(dG) && (24)\\ \Rightarrow\ &\tilde{A}(\omega)dx_v = -\tilde{I}x_v d\omega - \tilde{B}dg &&\\ \Rightarrow\ &dx_v = -Vd\bar{z}. && (25)\end{aligned}$$

Vectorizing and combining (19a)–(19b) using P.2 and the relation $\text{vec}(W\circ W\circ\Delta) = \bar{W}\delta$, we get

$$\bar{X}\begin{bmatrix}\delta\\ \lambda\end{bmatrix} = \begin{bmatrix}0\\ g\end{bmatrix}. \tag{26}$$

Since $\bar{W} > 0$ and $\tilde{X}$ is full column rank (Assumption **A4**), $\bar{X}$ is invertible and (26) has a unique solution. Taking the differential of (26) and using P.2, P.3, P.9, we get

$$\begin{aligned}\bar{X}\begin{bmatrix}d\delta\\ d\lambda\end{bmatrix} &= \begin{bmatrix}0\\ dg\end{bmatrix} - d\bar{X}\begin{bmatrix}\delta\\ \lambda\end{bmatrix} = \begin{bmatrix}0\\ dg\end{bmatrix} - \begin{bmatrix}d\tilde{X}^{\mathsf{T}}\lambda\\ d\tilde{X}\delta\end{bmatrix}\\ &= \begin{bmatrix}0\\ dg\end{bmatrix} - \begin{bmatrix}\text{vec}(\Lambda(CdX)^{\mathsf{T}})\\ \text{vec}(\Delta CdX)\end{bmatrix} = \begin{bmatrix}0\\ dg\end{bmatrix} - Edx_v\\ &\overset{(25)}{=} Yd\bar{z}. \end{aligned} \tag{27}$$

Next, we have

$$d\delta = \begin{bmatrix}I & 0\end{bmatrix}\begin{bmatrix}d\delta\\ d\lambda\end{bmatrix} \overset{(27)}{=} \underbrace{\begin{bmatrix}I & 0\end{bmatrix}\bar{X}^{-1}}_{U} Yd\bar{z} = Z^{\mathsf{T}}d\bar{z}.$$

Substituting $d\delta = Z^{\mathsf{T}}(d\bar{z})$ in (23) and using P.8, we get the gradient in (21).

Part 2 — Hessian: We derive the Hessian by expressing the second order differential of J_W as $d^2J_W = (d\bar{z})^{\mathsf{T}}(\cdot)(d\bar{z})$. Taking the differential of (23), we get

$$d^2J_W = (d\delta)^{\mathsf{T}}\bar{W}(d\delta) + (d^2\delta)^{\mathsf{T}}\bar{W}\delta. \tag{28}$$

Using $d\delta = Z^{\mathsf{T}}(d\bar{z})$ from the first part of the proof, the first term in (28) becomes $(d\bar{z})^{\mathsf{T}}Z\bar{W}Z^{\mathsf{T}}(d\bar{z})$. Next, we compute the second term in (28). Since G and ω are free variables, their second order differentials d^2G and $d^2\omega$ are zero (Magnus & Neudecker, 1999). Taking the differential of (24) and vectorizing, we have

$$\begin{aligned} &A(d^2X) - \omega(d^2X)\bar{I} - 2(d\omega)(dX)\bar{I} = 0 &&\\ \overset{\text{P.2}}{\Rightarrow}\ &d^2x_v = -2(d\omega)\tilde{A}(\omega)^{-1}\tilde{I}(dx_v) \overset{(25)}{=} 2(d\omega)\tilde{A}(\omega)^{-1}\tilde{I}V(d\bar{z}). && (29)\end{aligned}$$

Taking the differential of (27), we get

$$\begin{aligned} &\bar{X}\begin{bmatrix}d^2\delta\\ d^2\lambda\end{bmatrix} = -d\bar{X}\begin{bmatrix}d\delta\\ d\lambda\end{bmatrix} - dEdx_v - Ed^2x_v &&\\ &\qquad\qquad = -Ed^2x_v - 2dEdx_v &&\\ \Rightarrow\ &d^2\delta = \begin{bmatrix}I & 0\end{bmatrix}\begin{bmatrix}d^2\delta\\ d^2\lambda\end{bmatrix} = -\underbrace{\begin{bmatrix}I & 0\end{bmatrix}\bar{X}^{-1}}_{U}(Ed^2x_v + 2dEdx_v) &&\\ \Rightarrow\ &(d^2\delta)^{\mathsf{T}}\bar{W}\delta = -(d^2x_v)^{\mathsf{T}}E^{\mathsf{T}}U^{\mathsf{T}}\bar{W}\delta - 2(dEdx_v)^{\mathsf{T}}U^{\mathsf{T}}\bar{W}\delta. && (30)\end{aligned}$$

[4] If there is an even number of nodes, then there are two most critical edges corresponding to the self loops of the two center nodes.

Using (29) and $d\omega = e_{2m+1}^{\mathsf{T}}(d\bar{z})$, the first term on the right side of (30) is

$$\begin{aligned} &-(d^2x_v)^{\mathsf{T}}E^{\mathsf{T}}U^{\mathsf{T}}\bar{W}\delta \\ &\qquad = -2d\bar{z}^{\mathsf{T}}V^{\mathsf{T}}\tilde{I}^{\mathsf{T}}\tilde{A}(\omega)^{-\mathsf{T}}E^{\mathsf{T}}U^{\mathsf{T}}\bar{W}\delta e_{2m+1}^{\mathsf{T}}d\bar{z}. \end{aligned} \tag{31}$$

The second term on the right side of (30) is

$$\begin{aligned} &-2(dEdx_v)^{\mathsf{T}}U^{\mathsf{T}}\bar{W}\delta \\ &\overset{(25)}{=} 2d\bar{z}^{\mathsf{T}}V^{\mathsf{T}}\left[T_{2,n}(C^{\mathsf{T}}\otimes d\Lambda^{\mathsf{T}}) \quad I_2\otimes C^{\mathsf{T}}d\Delta^{\mathsf{T}}\right]\begin{bmatrix} U_1^{\mathsf{T}}\bar{W}\delta \\ U_2^{\mathsf{T}}\bar{W}\delta \end{bmatrix} \\ &= 2d\bar{z}^{\mathsf{T}}V^{\mathsf{T}}[T_{2,n}(C^{\mathsf{T}}\otimes d\Lambda^{\mathsf{T}})\mathrm{vec}(\bar{U}_1) + (I_2\otimes C^{\mathsf{T}}d\Delta^{\mathsf{T}})\mathrm{vec}(\bar{U}_2)] \\ &\overset{\text{P.3}}{=} 2d\bar{z}^{\mathsf{T}}V^{\mathsf{T}}[T_{2,n}((\bar{U}_1C)^{\mathsf{T}}\otimes I_2)T_{m,2}d\lambda + (\bar{U}_2^{\mathsf{T}}\otimes C^{\mathsf{T}})T_{m,p}d\delta] \\ &\overset{(27)}{=} 2d\bar{z}^{\mathsf{T}}V^{\mathsf{T}}\left[(\bar{U}_2^{\mathsf{T}}\otimes C^{\mathsf{T}})T_{m,p} \quad T_{2,n}(C^{\mathsf{T}}\bar{U}_1^{\mathsf{T}}\otimes I_2)T_{m,2}\right]\bar{X}^{-1}Yd\bar{z} \end{aligned} \tag{32}$$

Using (30), (31) and (32) in (28), we get

$$\begin{aligned} d^2J_W &= d\bar{z}^{\mathsf{T}}Z\bar{W}Z^{\mathsf{T}}d\bar{z} + 2(d\bar{z})^{\mathsf{T}}M(d\bar{z}) \\ &= (d\bar{z})^{\mathsf{T}}\left[Z\bar{W}Z^{\mathsf{T}} + M + M^{\mathsf{T}}\right](d\bar{z}). \end{aligned}$$

The expression of Hessian H in (22) follows from P.8. □

References

Benner, P., Kohler, M., & Saak, J. (2011). *Sparse-dense sylvester equations in $\mathcal{H}_2$-model order reduction. Technical Report MPIMD/11-11*. Magdeburg: Max Planck Institute for Dynamics of Complex Technical Systems.

Bhattacharyya, S. P., & De Souza, E. (1982). Pole assignment via Sylvester's equation. *Systems & Control Letters*, *1*(4), 261–263.

Bobylev, N. A., Bulatov, A. V., & Diamond, P. (1999). An easily computable estimate for the real unstructured F-stability radius. *International Journal of Control*, *72*(6), 493–500.

Bobylev, N. A., Bulatov, A. V., & Diamond, P. (2001). Estimates of the real structured radius of stability of linear dynamic systems. *Automation and Remote Control*, *62*(4), 505–512.

Boyd, S., & Balakrishnan, V. (1990). A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its H_∞-norm. *Systems & Control Letters*, *15*(1), 1–7.

Boyd, S., Balakrishnan, V., & Kabamba, P. (1989). A bisection method for computing the H_∞-norm of a transfer matrix and related problems. *Mathematics of Control, Signals, and Systems*, *2*(3), 207–219.

Brandwood, D. H. (1983). A complex gradient operator and its application in adaptive array theory. *IEE Proceedings F - Communications, Radar and Signal Processing*, *130*(1), 11–16.

Bruinsma, N. A., & Steinbucha, M. (1990). A fast algorithm to compute the H_∞-norm of a transfer function matrix. *Systems & Control Letters*, *14*(4), 287–293.

Byers, R. (1988). A bisection method for measuring the distance of a stable matrix to the unstable matrices. *SIAM Journal on Scientific and Statistical Computing*, *9*(5), 875–881.

Curtis, F. E., Mitchell, T., & Overton, M. L. (2017). A BFGS-SQP method for nonsmooth, nonconvex, constrained optimization and its evaluation using relative minimization profiles. *Optimization Methods & Software*, *32*(1), 148–181.

Freitag, M. A., & Spence, A. (2014). A new approach for calculating the real stability radius. *BIT Numerical Mathematics*, *54*, 381–400.

Guglielmi, N. (2016). On the method by Rostami for computing the real stability radius of large and sparse matrices. *SIAM Journal on Scientific Computing*, *38*(3), A1662–A1681.

Guglielmi, N., Gurbuzbalaban, M., Mitchell, T., & Overton, M. L. (2017). Approximating the real structured stability radius with Frobenius-norm bounded perturbations. *SIAM Journal on Matrix Analysis and Applications*, *38*(4), 1323–1353.

Guglielmi, N., & Lubich, C. (2013). Low-rank dynamics for computing extremal points of real pseudospectra. *SIAM Journal on Matrix Analysis and Applications*, *34*(1), 40–66.

Guglielmi, N., & Manneta, M. (2015). Approximating real stability radii. *IMA Journal of Numerical Analysis*, *35*(3), 1402–1425.

Hinrichsen, D., Kelb, B., & Linnemann, A. (1989). An algorithm for the computation of the structured complex stability radius. *Automatica*, *25*(5), 771–775.

Hinrichsen, D., & Pritchard, A. J. (1986a). Stability radii of linear systems. *Systems & Control Letters*, *7*(1), 1–10.

Hinrichsen, D., & Pritchard, A. J. (1986b). Stability radius for structured perturbations and the algebraic Riccati equation. *Systems & Control Letters*, *8*(2), 105–113.

Hinrichsen, D., & Pritchard, A. J. (2005). *Mathematical systems theory I: Modeling, state space analysis, stability and robustness*. Berlin Heidelberg: Springer-Verlag.

Hjørungnes, A., & Gesbert, D. (2007). Complex-valued matrix differentiation: Techniques and key results. *IEEE Transactions on Signal Processing*, *55*(6), 2740–2746.

Johnson, S. C., Wicks, M., Zefran, M., & DeCarlo, R. A. (2018). The structured distance to the nearest system without property $\mathcal{P}$. *IEEE Transactions on Automatic Control*, *63*(9), 2960–2975.

Kressner, D., & Voigt, M. (2015). Distance problems for linear dynamical systems. In *Numerical algebra, matrix theory, differential-algebraic equations and control theory* (pp. 559–583). Springer.

Luenberger, D. G., & Ye, Y. (2008). *Linear and nonlinear programming* (3rd ed.). Springer.

Magnus, J. R., & Neudecker, H. (1999). *Matrix differential calculus with applications in statistics and econometrics*. John Wiley and Sons.

Petersen, K. B., & Pedersen, M. S. (2012). *The matrix cookbook*. Technical University of Denmark.

Qiu, L., Bernhardsson, B., Rantzer, A., Davison, E. J., Young, P. M., & Doyle, J. C. (1995). A formula for computation of the real stability radius. *Automatica*, *31*(6), 879–890.

Qiu, L., & Davison, E. J. (1991). The stability robustness determination of state space models with real unstructured perturbations. *Mathematics of Control, Signals, and Systems*, *4*(3), 247–267.

Qiu, L., & Davison, E. J. (1992). Bounds on the real stability radius. In *Robustness of dynamic systems with parameter uncertainties* (pp. 139–145).

Rostami, M. W. (2015). New algorithms for computing the real structured pseudospectral abscissa and the real stability radius of large and sparse matrices. *SIAM Journal on Scientific Computing*, *37*(5), S447–S471.

Sreedhar, J., Van Dooren, P., & Tits, A. L. (1996). A fast algorithm to compute the real structured stability radius. *Stability Theory: International Series of Numerical Mathematics*, *121*, 219–230.

Van Loan, C. (1985). How near is a stable matrix to an unstable matrix?. *Contemporary Mathematics*, *47*, 465–478.

Vaibhav Katewa is a postdoctoral scholar in the department of Mechanical Engineering at University of California Riverside. He received his M.S. and Ph.D. degrees from University of Notre Dame in 2012 and 2016 and his Bachelor's degree from Indian Institute of Technology, Kanpur in 2007, all in Electrical Engineering. His research interests include analysis and design of security and privacy methods for cyber–physical systems and complex networks, decentralized and sparse feedback control, and protocol design for networked control systems.

Fabio Pasqualetti is an Associate Professor in the Department of Mechanical Engineering, University of California at Riverside. He completed a Doctor of Philosophy degree in Mechanical Engineering at the University of California, Santa Barbara, in 2012, a Laurea Magistrale degree (M.Sc. equivalent) in Automation Engineering at the University of Pisa, Italy, in 2007, and a Laurea degree (B.Sc. equivalent) in Computer Engineering at the University of Pisa, Italy, in 2004. His main research interests include the analysis and control of complex networks, security of cyber–physical systems, distributed control, and network neuroscience. He has received several awards, including a Young Investigator Research Program award from AFOSR in 2020, a Young Investigator Program award from ARO in 2017, and the 2016 TCNS Outstanding Paper Award from IEEE CSS. He is a member of IEEE and SIAM.