Iterative Algorithms for Assessing Network Resilience Against Structured Perturbations

Shenyu Liu Sonia Martínez Jorge Cortés

***Abstract*—This paper studies network resilience against structured additive perturbations to its topology. We consider dynamic networks modeled as linear time-invariant systems subject to perturbations of bounded energy satisfying specific sparsity and entry-wise constraints. Given an energy level, the structured pseudospectral abscissa captures the worst-possible perturbation an adversary could employ to de-stabilize the network, and the structured stability radius is the maximum energy in the structured perturbation that the network can withstand without becoming unstable. Building on a novel characterization of the worst-case structured perturbation, we propose iterative algorithms that efficiently compute the structured pseudospectral abscissa and structured stability radius. We provide theoretical guarantees of the local convergence of the algorithms and illustrate their efficacy and accuracy on several network examples.**

I. Introduction

The resilience of dynamic networks against perturbations and attacks is key across engineering, scientific, and military domains, including the operation of cyberphysical infrastructure, the distributed control of autonomous robots, and time-critical missions. Despite important advances in designing distributed coordination, cooperation, and decision-making algorithms, dynamic networks have proven fragile to targeted attacks, as local and well-orchestrated actions have rapidly cascaded into network-wide destructive perturbations. Because of this, it is critical to develop techniques and notions that characterize network resilience and allow us to understand strengths and vulnerabilities against adversaries and unforeseen failures. However, obtaining such characterizations is difficult because resilience is a complex function of the operator's and adversary's capabilities, knowledge, and resources, the topology of the network, and the physical limitations on remedial and adversarial actions. Motivated by these observations, this paper studies the relationship between network resilience and structured topological perturbations, with the ultimate goal of enabling the deployment of dynamic networks with quantifiable resilience guarantees.

Literature review: Network resilience, understood as the ability of the system to carry out its goals under unexpected failures or malfunctions in its components, is a rich research area. Multiple layers of network activity are involved in ensuring resilience (e.g., detection of failures or attacks, secure communications, injection of false data or actuation signals), which naturally gets reflected in the variety of disciplines employed in its study, e.g., computer security [1], [2], communications [3], control [4], and signal processing [5]. In the context of distributed control of network systems, the literature has studied diverse topics including algorithms for function computation [6], [7] and its robustness against malicious behavior [8], resilient estimation and control [9], [10], [11], attack detection [12], and the scaling of robustness with network size [13], to name a few.

Here, we model the network dynamics using a linear time-invariant system and analyze its stability properties in the face of additive perturbations to the entries of the system matrix. This connected our work with the study of matrix pseudospectra in linear algebra [14] and, in particular, the sign of the pseudospectral abscissa (the real part of the rightmost eigenvalue in the pseudospectrum). The work [15] shows that the pseudospectral abscissa is associated with a low-rank perturbation to the matrix. The works [16] and [17] propose criss-cross algorithms to numerically compute the value of the complex and real, resp., pseudospectral abscissa. Both algorithms rely on the method in [18] to compute the distance to instability, which requires eigenvalue computations of dense matrices and hence is impractical for large-scale systems. Instead, iterative algorithms [19], [20] quickly approximate complex or real pseudospectral abscissa of large matrices with sparse structures. However, due to their gradient-based nature, these algorithms possess local convergence properties and are only guaranteed to yield good approximations of the pseudospectral abscissa when the magnitude of the system perturbation is sufficiently small. A smooth version of spectral abscissa and its connection with pseudospectral abscissa is studied in [21]. Another closely-related concept is that of stability radius, which is the critical value of the magnitude of the perturbation that makes the pseudospectral abscissa become 0. While the works [19], [20], [22] propose efficient iterative algorithms for approximating stability radii of sparse matrices, the work [23] provides a formula for directly computing the stability radius when the perturbation is an arbitrary complex matrix and its magnitude is measured by its induced 2-norm. The work [24] gives a similar formula when the perturbation is a real matrix.

In practice, perturbations to the system matrix might be constrained by physical modeling, specific cyber vulnerabilities, or sparsity patterns. The study of the pseudospectral abcissa and the stability radius when perturbations are structured is much more limited. The fact that structured pseudospectra are closely tied to structured singular values [25], which can be NP-hard to compute [26], explains why no general formula exists for representing structured pseudospectra. The works [27], [28], [29] study structured pseudospectra for specific classes of perturbation matrices such as Toeplitz, symmetric, Hankel, or circulant. The work [30] studies the problem of characterizing

The authors are with the Department of Aerospace and Mechanical Engineering, UC San Diego, {shl055,soniamd, cortes}@ucsd.edu.
This work was partially supported by AFOSR Award FA9550-19-1-0235.

the smallest additive matrix perturbation to an LTI system so that it loses controllability, observability, or stability. The work proposes an asymptotically convergence algorithm to obtain a locally optimal perturbation based on the identification of necessary conditions which, given the problem generality, are expressed implicitly in terms of abstract linear maps. Closely related to this effort, a recent work [31] proposes a subspace method for estimating the structured stability radius of large-scale systems. However, "structured perturbation" is defined differently, in the sense that the considered perturbations belong to a given subspace of matrices, rather than referring to sparse matrices with bounded elements as studied here. On the other hand, the work [32] considers bounded-energy perturbations with sparse structure and studies the structured stability radius. The treatment relaxes the sparsity constraints by incorporating them into a cost function and relies on increasingly large penalties to satisfy them with increasing accuracy. This increasing accuracy comes at the expense of reducing the convergence speed of the proposed gradient-based algorithm, which makes it not well suited for large-scale networks. In contrast, our approach here presents a general characterization of the worst-case structured perturbation that includes the possibility of element-wise constraints. This serves as the basis for our design of efficient iterative algorithms to compute both the structured pseudospectral abscissa and stability radius that are able to deal with large-scale systems.

Statement of contributions: We model the network as a linear-time invariant system and study the effect on stability of additive perturbations to the system matrix of bounded energy and subject to sparsity and element-wise saturation constraints. Our contributions address the questions of whether an adversary can destabilize the network by employing such perturbations and characterizing the maximum amount of energy in the perturbation that the network can withstand without becoming unstable. Our first contribution is a novel necessary condition prescribing that the worst-case structured perturbation to the network must solve an implicit optimization problem. The implicitness arises because of the dependence on the right and left eigenvectors associated to the structured pseudospectral abscissa and, if they were known instead, the optimization would become explicit and convex. We provide a complete description of the solution to the explicit optimization, show it is Lipschitz with respect to the problem parameters, and provide an incremental method to compute it. The observation about the implicit character of the optimization problem is the basis for our second contribution, which is an iterative algorithm alternating between finding the right and left eigenvectors given an estimate of the worst-case structured perturbation and solving the corresponding explicit optimization problem to refine said estimate. We show that the proposed algorithm is guaranteed to converge to the structured pseudospectral abscissa at a linear rate for sufficiently close initial conditions. Our final contribution concerns the structured stability radius, and builds on the fact that this radius corresponds to the zero-crossing of the structured pseudospectral abscissa when viewed as a function of the perturbation energy. We establish the locally Lipschitzness of this function, explicitly compute its gradient at the points of differentiability, and employ Newton's method to design an iterative algorithm that provably approximates the structured stability radius. We illustrate in simulation the efficiency and accuracy of the proposed algorithms on several network examples, including a class of stable large-scale systems.

II. Preliminaries

Here, we introduce[1] basic notions from linear algebra.

Spectral abscissa and stability radius: We denote by $\Lambda(A)$ the spectrum (i.e., set of eigenvalues) of a square matrix A. The spectral abscissa of A is

$$\alpha(A) := \max_{\lambda\in\Lambda(A)} \operatorname{Re}\lambda. \tag{2}$$

We refer to a maximizer $\lambda_{\rm opt}$ of this function as a rightmost eigenvalue of $\Lambda(A)$. For $\epsilon > 0$ and a closed set $\mathcal{H} \subseteq \mathbb{C}^{n\times n}$, the *structured ϵ- pseudospectrum* of A (with respect to the Frobenius norm) is

$$\Lambda_{\epsilon,\mathcal{H}}(A) := \{\lambda \in \mathbb{C} : \lambda \in \Lambda(A+\Delta) \text{ for } \Delta \in \mathcal{H} \cap \mathbb{B}_\epsilon\}.$$

Note that when $\mathcal{H} = \mathbb{C}^{n\times n}$ or $\mathcal{H} = \mathbb{R}^{n\times n}$, $\Lambda_{\epsilon,\mathcal{H}}$ reduces to the usual ϵ-pseudospectrum [19] or real ϵ-pseudospectrum [15], resp. Similar to the spectral abscissa, we also define $\alpha_{\epsilon,\mathcal{H}}(A)$ as the *structured ϵ-pseudospectral abscissa* of A,

$$\alpha_{\epsilon,\mathcal{H}}(A) := \max_{\lambda\in\Lambda_{\epsilon,\mathcal{H}}(A)} \operatorname{Re}\lambda. \tag{3}$$

We refer to a maximizer $\lambda_{\rm opt}$ of this function as a rightmost eigenvalue of $\Lambda_{\epsilon,\mathcal{H}}(A)$. Using (2), one can equivalently express the structured ϵ-pseudospectral abscissa of A as

$$\alpha_{\epsilon,\mathcal{H}}(A) = \max_{\Delta\in\mathcal{H}\cap\mathbb{B}_\epsilon} \alpha(A+\Delta). \tag{4}$$

We refer to a maximizer $\Delta_{\rm opt}$ of this function as a worst-case structured perturbation of energy ϵ. The *structured stability radius* of A is

$$r_{\mathcal{H}}(A) := \min_{\epsilon\geq 0}\{\epsilon : \alpha_{\epsilon,\mathcal{H}}(A) \geq 0\}. \tag{5}$$

[1]Throughout the paper, we use the following notation. For any vector $x \in \mathbb{C}^n$ or matrix $A \in \mathbb{C}^{n\times m}$, let x^*, A^* be their conjugate transpose. Let $|\cdot|$ be the 2-norm of vectors in $\mathbb{C}^n$, that is, $|x| := \sqrt{x^*x}$. In addition, let $\|\cdot\|_2$, $\|\cdot\|_F$ be the induced 2-norm and Frobenius norm, resp., on $\mathbb{C}^{n\times n}$. We let $\mathbb{B}_r := \{\Delta \in \mathbb{C}^{n\times n} : \|\Delta\|_F \leq r\}$ denote the closed ball of radius r in $\mathbb{C}^{n\times n}$. We say vectors $x, y \in \mathbb{C}^n$ are *RP-compatible*, cf. [19], if $|x| = |y| = 1$ and x^*y is real and positive. Given any x, y with $x^*y \neq 0$, one can obtain a pair of RP-compatible vectors $\tilde{x}$, $\tilde{y}$ by scaling x and y as

$$\tilde{x} = \frac{x}{|x|}, \quad \tilde{y} = \frac{y^*x}{|y^*x|}\frac{y}{|y|}.$$

The inner product $\langle\cdot,\cdot\rangle : \mathbb{C}^{n\times n} \times \mathbb{C}^{n\times n} \to \mathbb{C}$ of matrices $A = [a_{ij}]$ and $B = [b_{ij}]$ is $\langle A, B\rangle := \operatorname{Tr}(A^*B) = \sum_{i=1}^n \sum_{j=1}^n a_{ij}^* b_{ij}$. Note that $\langle A, A\rangle = \|A\|_F^2$. In addition, for any $x, y \in \mathbb{C}^n$, $M \in \mathbb{R}^{n\times n}$,

$$\operatorname{Re}(x^*My) = \langle M, \operatorname{Re}(xy^*)\rangle. \tag{1}$$

The *group inverse* of A, denoted $A^\#$, is the unique matrix satisfying $AA^\# = A^\#A$, $A^\#AA^\# = A^\#$, and $AA^\#A = A$, cf. [33]. The group inverse is different from the Moore–Penrose pseudoinverse. For a function $f : \mathbb{R} \to \mathbb{R}$, define the right derivative of f at x to be

$$\partial_+ f(x) := \lim_{\delta\to 0^+} \frac{f(x+\delta) - f(x)}{\delta}.$$

For functions $f, g : \mathbb{R}_{\geq 0} \to \mathbb{R}$, we denote $f(t) = O(g(t))$ if there exists $k, \delta > 0$ such that $|f(t)| \leq kg(t)$ for $t < \delta$.

Clearly, if A is not Hurwitz, $r_{\mathcal{H}}(A) = 0$. When $\mathcal{H} = \mathbb{C}^{n\times n}$ or $\mathcal{H} = \mathbb{R}^{n\times n}$, $r_{\mathcal{H}}(A)$ coincides with the definition of stability radius [23] or real stability radius [24], resp. Note also that if $\mathcal{H}_1 \subseteq \mathcal{H}_2$, then $r_{\mathcal{H}_1}(A) \geq r_{\mathcal{H}_2}(A)$.

Matrix perturbation theory: Here we gather useful results on matrix perturbations. The first describes the derivative of a simple eigenvalue of a matrix that depends linearly on time.

Lemma II.1 ([34, Lemma 6.3.10 and Theorem 6.3.12])**.** *Consider a $n\times n$ matrix trajectory $\mathbb{R} \ni t \mapsto C(t) = C_0 + tC_1$. Let $\lambda(t)$ be an eigenvalue of $C(t)$ converging to a simple eigenvalue λ_0 of C_0 as $t \to 0$. Let x_0 and y_0 be, resp., right and left eigenvectors of C_0 corresponding to λ_0, that is, $(C_0 - \lambda_0 I)x_0 = 0$ and $y_0^*(C_0 - \lambda_0 I) = 0$. Then $y_0^* x_0 \neq 0$ and $\lambda(t)$ is analytic near $t = 0$ with*

$$\left.\frac{d\lambda(t)}{dt}\right|_{t=0} = \frac{y_0^* C_1 x_0}{y_0^* x_0}.$$

The following result describes the time derivative of the product of the right and left eigenvectors corresponding to a simple eigenvalue of a complex-valued time-dependent matrix.

Theorem II.2 ([19, Theorem 5.2])**.** *Consider a $n\times n$ complex-analytic matrix trajectory $\mathbb{C} \ni t \mapsto C(t) = C_0 + tC_1 + O(t^2)$. Let $\lambda(t)$ be a simple eigenvalue of $C(t)$ in a neighborhood $\mathcal{N}$ of $t = 0$, with corresponding RP-compatible right and left eigenvectors $x(t)$ and $y(t)$. Then $Q(t) = x(t)y(t)^*$ is C^∞ on $\mathcal{N}$ and its derivative at $t = 0$ is*

$$\left.\frac{dQ(t)}{dt}\right|_{t=0} = \mathrm{Re}(\beta+\gamma)Q(0) - GC_1Q(0) - Q(0)C_1G, \quad (6)$$

where $G = (C_0 - \lambda I)^{\#}$, $\beta = x^*GC_1x$, $\gamma = y^*C_1Gy$, $\lambda = \lambda(0)$, $x = x(0)$, and $y = y(0)$.

III. Problem statement

We provide here a formal mathematical description of the problem of interest. Let $\mathcal{G} := \{\mathcal{V}, \mathcal{E}\}$ denote a *network graph*, where $\mathcal{V} = \{1, \ldots, n\}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. The network dynamics is described by a linear differential equation

$$\dot{x} = Ax, \quad (7)$$

where the components of $x \in \mathbb{R}^n$ correspond to the states of the nodes, and $A = [a_{ij}] \in \mathbb{R}^{n\times n}$, with $a_{ij} = 0$ for all $(i, j) \notin \mathcal{E}$, is the weighted adjacency matrix. We assume the matrix A is Hurwitz. An adversary seeks to destabilize the dynamics by attacking the network interconnections. Such attacks are structured, in the sense that the adversary is limited to perturbing only certain edges and within some budget. Formally, let $\mathcal{E}_p \subseteq \mathcal{E}$, denote the *perturbation edge set* and for all $(i, j) \in \mathcal{E}_p$, let $\underline{\Delta}_{ij} \in \mathbb{R}_{\leq 0} \cup \{-\infty\}$, $\overline{\Delta}_{ij} \in \mathbb{R}_{\geq 0} \cup \{+\infty\}$ be parameters specifying *saturation constraints*. We denote the set of allowed perturbations whose sparsity pattern is compatible with $\mathcal{E}_p$ by

$$\mathcal{H} := \{\Delta = [\Delta_{ij}] \in \mathbb{R}^{n\times n} : \Delta_{ij} = 0 \text{ if } (i, j) \notin \mathcal{E}_p, \\ \Delta_{ij} \in [\underline{\Delta}_{ij}, \overline{\Delta}_{ij}] \text{ if } (i, j) \in \mathcal{E}_p\}. \quad (8)$$

Note that we always have $0 \in \mathcal{H}$. When $\underline{\Delta}_{ij} = -\infty$ or $\overline{\Delta}_{ij} = +\infty$, then there is no lower or upper bound constraint on the perturbation size of the corresponding edge. We define the energy of an attack Δ to be its Frobenius norm.

After an attack $\Delta \in \mathcal{H}$, the network dynamics changes to

$$\dot{x} = (A + \Delta)x. \quad (9)$$

We are interested in answering the following questions:

1) Given the network dynamics (7), can an adversary destabilize it by employing perturbations of bounded energy $\epsilon > 0$ in $\mathcal{H}$?
2) If by resilience of the network dynamics we understand the maximum amount of energy in the perturbation that it can withstand without becoming unstable, what is the network resilience against the adversary?

Each of the two questions can be transcribed into a mathematically precise statement. In fact, keeping in mind that (9) is globally asymptotically stable if $A + \Delta$ is Hurwitz, we can equivalently say that

- question 1) refers to determining whether the structured ϵ-pseudospectral abscissa $\alpha_{\epsilon,\mathcal{H}}(A)$ of A is positive, and
- question 2) refers to determining the value of the structured stability radius $r_{\mathcal{H}}(A)$.

In our ensuing discussion, we address question 1) by introducing an iterative algorithm that finds both the value of $\alpha_{\epsilon,\mathcal{H}}(A)$ and a maximizer Δ_{opt} of (4). We then answer question 2) by designing another iterative algorithm which finds the value of $r_{\mathcal{H}}(A)$. The solutions to these two questions make it possible to formulate interesting questions for network protection, such as finding the "most vulnerable" edges in the network and determining how much "stronger" the network can become by protecting these edges. We discuss these aspects briefly in the examples of Section VI.

IV. Network stability against perturbations: structured pseudospectral abscissa

In this section, we study the answers to questions 1) and 2) of our problem statement. We begin by characterizing the worst-case structured perturbation of a given energy. We build on this characterization later to propose an algorithm that computes iteratively the structured pseudospectral abscissa.

A. Characterization of the worst-case structured perturbation

Our first result states that any worst-case structured perturbation which gives rise to a simple rightmost eigenvalue is a solution to an implicit optimization problem.

Theorem IV.1 (Necessary condition for optimality)**.** *Let $\mathcal{H}' \subseteq \mathbb{C}^{n\times n}$ be compact and convex, and let*

$$\Delta_{\mathrm{opt}} \in \arg\max_{\Delta\in\mathcal{H}'} \alpha(A + \Delta). \quad (10)$$

Let $\lambda_{\mathrm{opt}}(A + \Delta_{\mathrm{opt}})$ be a rightmost eigenvalue of $A + \Delta_{\mathrm{opt}}$ and suppose it is simple. Then Δ_{opt} must satisfy

$$\Delta_{\mathrm{opt}} \in \arg\max_{\Delta\in\mathcal{H}'} \langle \Delta, \mathrm{Re}(yx^*)\rangle, \quad (11)$$

where $x, y \in \mathbb{C}^n$ are the RP-compatible right and left eigenvectors associated with $\lambda_{\mathrm{opt}}(A + \Delta_{\mathrm{opt}})$.

Proof: To study the necessary condition of optimality, we consider the set $\mathcal{F}$ of feasible directions at $\Delta \in \mathcal{H}'$,

$$\mathcal{F} := \{C \in \mathbb{C}^{n\times n} : \exists \tau > 0 \text{ s.t. } \Delta + tC \in \mathcal{H}',\ \forall t \in [0,\tau]\}.$$

The necessary condition for optimality states that, if $\Delta_{\rm opt}$ is a local maximizer, then $\frac{d}{dt}\alpha(A+\Delta_{\rm opt}+tC)\big|_{t=0} \le 0$, for all $C \in \mathcal{F}$. Applying Lemma II.1, we deduce

$$\begin{aligned} 0 &\ge \mathrm{Re}\left(\frac{d\lambda_{\rm opt}(A+\Delta_{\rm opt}+tC)}{dt}\Big|_{t=0}\right) \\ &= \mathrm{Re}\left(\frac{y^*Cx}{y^*x}\right) = \frac{\mathrm{Re}(y^*Cx)}{y^*x} = \frac{\langle C, \mathrm{Re}(yx^*)\rangle}{y^*x}, \end{aligned}$$

where we have used the RP-compatibility for the second last equality and the identity (1) for the last equality. Now, one can see that $\langle C, \mathrm{Re}(yx^*)\rangle \le 0$ for all $C \in \mathcal{F}$ corresponds indeed to the necessary condition of optimality for the problem (11). Moreover, since this problem is convex, the condition is also sufficient to characterize an optimizer. Thus, satisfying the condition of (11) is a necessary condition for being a maximizer of (10). ∎

It is worth pointing out that, since (11) is a necessary condition, it needs to be satisfied by any worst-case perturbation. In this section we first design an iterative algorithm to estimate a structured perturbation satisfying (11). We then prove that when the initial guess to the algorithm is close enough to an optimizer of (10), then the output of this algorithm will in fact asymptotically converge to this optimizer.

Comparing with (4), we observe that (10) with $\mathcal{H}' = \mathcal{H}\cap\mathbb{B}_\epsilon$ corresponds exactly to a worst-case structured perturbation of energy ϵ (in fact, $\mathcal{H}$ as defined in (8) is closed and convex, and hence $\mathcal{H}'$ is compact and convex). The optimization problem in (10) is nonconvex and hence difficult to solve in general. Instead, Theorem IV.1 facilitates finding the structured pseudospectral abscissa by providing a characterization (11) of the worst-case structured perturbations.

For known $x, y \in \mathbb{C}^n$, the optimization in (11) is a convex problem of the form

$$\begin{aligned} \text{maximize} \quad & \langle \Delta, M\rangle \\ \text{subject to} \quad & \Delta \in \mathcal{H}\cap\mathbb{B}_\epsilon, \end{aligned} \tag{12}$$

when we set $M = [m_{ij}] := \mathrm{Re}(yx^*)$, and can therefore be solved efficiently. However, we should note that the vectors x and y in (11) are unknown, since they are the eigenvectors of $A+\Delta_{\rm opt}$, making equation (11) implicit in $\Delta_{\rm opt}$. Before we address this issue, we finish the exposition here describing the properties of the solution of (12) for a given known M.

We make the next assumption regarding the worst-case perturbation.

Assumption 1. *(Non-saturation at optimizers).* Given $\Delta \in \mathcal{H}$, let $\mathcal{S}_{\rm ns}(\Delta) := \{(i,j) \in \mathcal{E}_p : (\Delta)_{ij} \in (\underline{\Delta}_{ij}, \overline{\Delta}_{ij})\}$. Then, for any optimizer $\Delta_{\rm opt}$ of (12), we have $\mathcal{S}_{\rm ns}(\Delta_{\rm opt}) \neq \emptyset$ and there exists $(i,j) \in \mathcal{S}_{\rm ns}(\Delta_{\rm opt})$ such that $m_{ij} \neq 0$.

Assumption 1 is reasonable: in case it does not hold, i.e., a worst-case perturbation is fully saturated, then it must be at a vertex of $\mathcal{H}$ and since this constraint set has finitely many vertices, the worst-case perturbation can be found by exhaustion. In addition, if $m_{ij} = 0$ for all $(i,j) \in \mathcal{S}_{\rm ns}$, then the values of those $(\Delta_{\rm opt})_{ij}$ do not affect the optimal value and, consequently, one can construct other optimizers that are fully saturated. We observe that since (12) has a non-trivial linear objective function with a convex constraint set, the optimum is achieved on its boundary and hence, if the maximizer $\Delta_{\rm opt}$ is not fully saturated, it must verify $\|\Delta_{\rm opt}\|_F = \epsilon$. In addition, by employing the KKT conditions for optimality, we are able to characterize the solution of (12).

Proposition IV.2. *(Characterization of solution of* (12)*). Let $M \in \mathbb{C}^{n\times n}$ and $\epsilon > 0$. Under Assumption 1, the optimization* (12) *has a unique optimizer $\Delta_{\rm opt}$ given by*

$$(\Delta_{\rm opt})_{ij} = \begin{cases} m_{ij}\theta_{\rm opt}, & \text{if } (i,j) \in \mathcal{E}_p\backslash(\overline{\mathcal{S}}\cup\underline{\mathcal{S}}), \\ \overline{\Delta}_{ij}, & \text{if } (i,j) \in \overline{\mathcal{S}}, \\ \underline{\Delta}_{ij}, & \text{if } (i,j) \in \underline{\mathcal{S}}, \\ 0, & \text{if } (i,j) \notin \mathcal{E}_p \end{cases} \tag{13}$$

where $\overline{\mathcal{S}} := \overline{\mathcal{S}}(\epsilon, M), \underline{\mathcal{S}} := \underline{\mathcal{S}}(\epsilon, M)$ are the unique subsets of $\mathcal{E}_p$ such that

$$m_{ij}\theta_{\rm opt} \in (\underline{\Delta}_{ij}, \overline{\Delta}_{ij}) \quad \forall (i,j) \in \mathcal{E}_p\backslash(\overline{\mathcal{S}}\cup\underline{\mathcal{S}}), \tag{14a}$$

$$m_{ij}\theta_{\rm opt} \ge \overline{\Delta}_{ij} \quad \forall (i,j) \in \overline{\mathcal{S}}, \tag{14b}$$

$$m_{ij}\theta_{\rm opt} \le \underline{\Delta}_{ij} \quad \forall (i,j) \in \underline{\mathcal{S}} \tag{14c}$$

and $\theta_{\rm opt}$ is shorthand notation for $\theta_{\rm opt}(\epsilon, M) := \theta(\epsilon, M, \overline{\mathcal{S}}(\epsilon, M), \underline{\mathcal{S}}(\epsilon, M))$, where the function θ is

$$\theta(\epsilon, M, \overline{\mathcal{S}}, \underline{\mathcal{S}}) := \sqrt{\frac{\epsilon^2 - \sum_{(i,j)\in\overline{\mathcal{S}}} \overline{\Delta}_{ij}^2 - \sum_{(i,j)\in\underline{\mathcal{S}}} \underline{\Delta}_{ij}^2}{\sum_{(i,j)\in\mathcal{E}_p\backslash(\overline{\mathcal{S}}\cup\underline{\mathcal{S}})} m_{ij}^2}}. \tag{15}$$

In addition, there is a neighborhood D around $(\epsilon, M) \in \mathbb{R}_{\ge 0}\times\mathbb{R}^{n\times n}$ such that $\theta_{\rm opt}$ is Lipschitz on D.

Due to space constraint, we refer the readers to the online version [35] for the proof of Proposition IV.2. According to this result, the element of the optimizer $\Delta_{\rm opt}$ corresponding to $(i,j) \in \mathcal{E}_p$ is either saturated or proportional to m_{ij}, with ratio given by $\theta_{\rm opt}$. We refer to $\overline{\mathcal{S}}, \underline{\mathcal{S}}$ as the *index sets of saturation* as $(\Delta_{\rm opt})_{ij}$ attains either of its boundary values for all $(i,j) \in \overline{\mathcal{S}}\cup\underline{\mathcal{S}}$. Note that, by Assumption 1, $\overline{\mathcal{S}}\cup\underline{\mathcal{S}} \subsetneq \mathcal{E}_p$ and the right-hand side of (15) has a non-zero denominator since there exists $(i,j) \in \mathcal{E}_p\backslash(\overline{\mathcal{S}}\cup\underline{\mathcal{S}})$ such that $m_{ij} \neq 0$.

Remark IV.3. *(Comparison with the literature).* Proposition IV.2 is a generalization of the results available in the literature [15], [32]. When there are neither sparsity constraints nor saturation constraints, $\mathcal{H} = \mathbb{R}^{n\times n}$ and $\overline{\mathcal{S}} = \underline{\mathcal{S}} = \emptyset$. Then it follows from (15) that

$$\theta_{\rm opt} = \sqrt{\frac{\epsilon^2}{\sum_{(i,j)\in\mathcal{V}^2} m_{ij}^2}} = \frac{\epsilon}{\|M\|_F}, \tag{16}$$

so $\Delta_{\rm opt} = \frac{\epsilon\,\mathrm{Re}(yx^*)}{\|\mathrm{Re}(yx^*)\|_F}$, as stated in [15, Theorem 2.2]. On the other hand, when $\mathcal{H}$ contains sparsity constraints but no saturation constraints, we deduce there exists $c \ge 0$ such that

$$(\Delta_{\rm opt})_{ij} = \begin{cases} c\,\mathrm{Re}(yx^*), & \text{if } (i,j) \in \mathcal{E}_p, \\ 0, & \text{otherwise,} \end{cases}$$

as stated in [32, Theorem 3.2]. ∎

We note that $\theta_{\rm opt}$ and $\overline{\mathcal{S}}$, $\underline{\mathcal{S}}$ are inter-dependent, which means that Proposition IV.2 does not provide an explicit expression of $\Delta_{\rm opt}$. However, the result provides the basis for a simple method, which we summarize in Algorithm 1, to find the solution of (12) by growing the index sets of saturation $\overline{\mathcal{S}}, \underline{\mathcal{S}}$ if they do not meet the conditions (14a)–(14c) for the corresponding value of $\theta_{\rm opt}$ determined by (15). The next result shows that Algorithm 1 finds the solution of (12).

Algorithm 1 Incremental construction of index sets

Input: $\epsilon, M, \mathcal{H}$
Output: $\theta_{\rm opt}, \Delta_{\rm opt}$

$\overline{\mathcal{S}} \leftarrow \emptyset, \underline{\mathcal{S}} \leftarrow \emptyset$,
NotDone $\leftarrow$ false
if $\overline{\mathcal{S}} \cup \underline{\mathcal{S}} = \mathcal{E}_p$ **then** break with error "the optimizer is fully saturated, Assumption 1 is violated"
Compute $\theta_{\rm opt}$ as in (15)
for all $(i,j) \in \mathcal{E}_p \backslash (\overline{\mathcal{S}} \cup \underline{\mathcal{S}})$ **do**
  if $m_{ij}\theta_{\rm opt} \geq \overline{\Delta}_{ij}$ **then**
    $\overline{\mathcal{S}} \leftarrow \overline{\mathcal{S}} \cup \{(i,j)\}$
    $(\Delta_{\rm opt})_{ij} \leftarrow \overline{\Delta}_{ij}$
    NotDone $\leftarrow$ true
  else if $m_{ij}\theta_{\rm opt} \leq \underline{\Delta}_{ij}$ **then**
    $\underline{\mathcal{S}} \leftarrow \underline{\mathcal{S}} \cup \{(i,j)\}$
    $(\Delta_{\rm opt})_{ij} \leftarrow \underline{\Delta}_{ij}$
    NotDone $\leftarrow$ true
if NotDone = true **then** go back to Step 2
else $(\Delta_{\rm opt})_{ij} \leftarrow m_{ij}\theta_{\rm opt}$ for all $(i,j) \in \mathcal{E}_p \backslash (\overline{\mathcal{S}} \cup \underline{\mathcal{S}})$

Lemma IV.4. *(Algorithm 1 solves* (12)*). Under Assumption 1, Algorithm 1 finds the solution of the optimization* (12) *in a finite number of steps.*

The proof of Lemma IV.4 is provided in the Appendix. In contrast to generic convex optimization solvers, Algorithm 1 is tailored to problem (12) and takes advantage of the characterization (13) of its optimizer. We use later the ratio $\theta_{\rm opt}$ in Algorithm 1 to compute the structured stability radius.

We conclude this section by presenting a result which shows that the optimizer of (12) is locally Lipschitz when viewed as a function of the matrix defining the objective function. To establish this, we use the Frobenius norm and show that the Lipschitz constant is proportional to the energy of matrix perturbations. The proof is given in the Appendix.

Lemma IV.5. *(Sensitivity of the optimizer of* (12) *with respect to parameters). Let $\epsilon > 0, M_1, M_2 \in \mathbb{R}^{n\times n}$ and suppose Δ_k are the optimizers of* (12) *with parameters $(\epsilon, M_k), k = 1, 2$. Also assume that Assumption 1 holds for (ϵ, M_1). Then, there exist $\delta = \delta(M_1) > 0, \ell = \ell(M_1) > 0$ such that*

$$\|\Delta_1 - \Delta_2\|_F \leq \ell\epsilon\|M_1 - M_2\|_F$$

as long as $\|M_1 - M_2\|_F \leq \delta$.

B. Iterative computation of structured pseudospectral abscissa

What we have unveiled about the optimization problem (12) and the structure of its solution in Section IV-A is not directly applicable to the determination of the worst-case structured perturbation and the structured pseudospectral abscissa. This is because, as we mentioned earlier, the characterization (11) is implicit in $\Delta_{\rm opt}$, i.e., the matrix $M = \mathrm{Re}(yx^*)$ required to set up (12) is not a priori known, and in fact depends on the optimizer itself. To address this obstacle, we propose in Algorithm 2 an iterative strategy that proceeds by repeatedly solving instances of problem (12), in each case taking the right and left eigenvectors corresponding to the previous iterate.

Algorithm 2 Computation of worst-case perturbation

Input: $A, \epsilon, \mathcal{H}, \Delta_0, \mathrm{tol}_\Delta$
Output: $\Delta, \alpha, x, y, \theta$

$\mathcal{H}' \leftarrow \mathcal{H} \cap \mathbb{B}_\epsilon$
$A_0 \leftarrow A + \Delta_0$
repeat $k = 0, 1, \cdots$
  Compute a rightmost eigenvalue λ_k of A_k and its corresponding RP-compatible right and left eigenvectors x_k, y_k
  $M \leftarrow \mathrm{Re}(y_k x_k^*)$
  Run Algorithm 1 with inputs $\epsilon, M, \mathcal{H}$, set the outputs $\theta_{k+1}, \Delta_{k+1}$
  $A_{k+1} \leftarrow A + \Delta_{k+1}$
until $\|\Delta_{k+1} - \Delta_k\|_2 \leq \mathrm{tol}_\Delta$
$(\Delta, \alpha, x, y, \theta) \leftarrow (\Delta_{k+1}, \mathrm{Re}\,\lambda_{k+1}, x_{k+1}, y_{k+1}, \theta_{k+1})$

The logic of Algorithm 2 can be described as follows. At each step, we consider a candidate worst-case perturbation Δ_k, followed by computing the RP-compatible right and left eigenvalues x_k, y_k of a rightmost eigenvalue of $A + \Delta_k$. We then solve the optimization problem (12) using $M = \mathrm{Re}(y_k x_k^*)$, and set the new optimizer to be Δ_{k+1}. This process is repeated until the sequence of possible worst-case perturbations converges. From its design, it is clear that a fixed point of Algorithm 2 is a solution to the maximization problem (11) (and hence, by Theorem IV.1, satisfies the necessary condition for being the maximizer of (10)).

Remark IV.6. *(Continuous-time version of Algorithm 2).* Consider the matrix-valued differential equation $\dot{\Delta}(t) = \Delta^\circ(t) - \Delta(t)$, where $\Delta^\circ(t) \in \arg\max_{\Delta' \in \mathcal{H}\cap\mathbb{B}_\epsilon} \langle \Delta', \mathrm{Re}\,(y(t)x(t)^*)\rangle$ and $x(t), y(t)$ are the RP-compatible right and left eigenvectors associated with the rightmost eigenvalue of $A + \Delta(t)$. Then, it follows from Lemma II.1 that, for almost all $t \geq 0$,

$$\begin{aligned}\frac{d}{dt}\alpha(A+\Delta(t)) &= \mathrm{Re}\left(\frac{y(t)^*\dot{\Delta}(t)x(t)}{y(t)^*x(t)}\right)\\ &= \frac{\langle \dot{\Delta}(t), \mathrm{Re}(y(t)x(t)^*)\rangle}{y(t)^*x(t)}\\ &= \frac{\langle \Delta^\circ(t), \mathrm{Re}(y(t)x(t)^*)\rangle - \langle \Delta(t), \mathrm{Re}(y(t)x(t)^*)\rangle}{y(t)^*x(t)} \geq 0.\end{aligned}$$

In addition, the inequality becomes an equality if and only if $\Delta(t) \in \arg\max_{\Delta' \in \mathcal{H}\cap\mathbb{B}_\epsilon} \langle \Delta', \mathrm{Re}\,(y(t)x(t)^*)\rangle$, that is, when $\Delta(t)$ satisfies the necessary condition for maximality stated by Theorem IV.1. Hence the solution of the above differential inclusion gives a monotonically increasing function $t \mapsto \alpha(A + \Delta(t))$, which converges to a solution of (11). It is not hard to see that Algorithm 2 is a discrete-time implementation

of this matrix-valued differential inclusion with stepsize 1. In the next result, we show that by choosing this fixed stepsize, and when the initial guess is close enough to an optimizer, Algorithm 2 is guaranteed to converge to this optimizer at a linear rate. This is a stronger property than what is guaranteed for the differential inclusion, which is convergence to a point satisfying the necessary condition of being optimal. ■

Theorem IV.7 (Local convergence of Algorithm 2)**.** *Let $\Delta_{\rm opt}$ be a worst-case structured perturbation (i.e., $\Delta_{\rm opt}$ satisfies (10) with $\mathcal{H}' = \mathcal{H} \cap \mathbb{B}_\epsilon$) and assume the rightmost eigenvalue λ of $A + \Delta_{\rm opt}$ is simple, with RP-compatible right and left eigenvector pair x, y. Let Assumption 1 hold for $\Delta_{\rm opt}$ and $M = \mathrm{Re}(yx^*)$. Define*

$$r = \frac{4\sqrt{n}\ell\epsilon}{\sigma_{n-1}(A-\lambda I)(y^*x)^2}, \tag{17}$$

where $\ell = \ell(\mathrm{Re}(yx^*))$ is given in Lemma IV.5 and σ_{n-1} denotes the second smallest singular value. Let $\Delta_k, \mathrm{Re}\,\lambda_k$ be the sequences generated by Algorithm 2. If $r < 1$, $r^\dagger \in (r, 1)$ is arbitrary and $\|\Delta_0 - \Delta_{\rm opt}\|_2$ is sufficiently small, then

$$\|\Delta_k - \Delta_{\rm opt}\|_2 \le (r^\dagger)^k \|\Delta_0 - \Delta_{\rm opt}\|_2 \tag{18}$$

for all $k = 0, 1, \ldots$ and $\mathrm{Re}\,\lambda_k$ converges to the structured pseudospectral abscissa $\alpha_{\epsilon,\mathcal{H}}(A)$. In addition, the output of Algorithm 2 satisfies

$$|\alpha - \alpha_{\epsilon,\mathcal{H}}(A)| = \frac{\sqrt{n} r^\dagger \|\mathrm{Re}(yx^*)\|_F}{(1-r^\dagger) y^*x}\,\mathrm{tol}_\Delta + O(\mathrm{tol}_\Delta^2). \tag{19}$$

Proof: Let $L := yx^*$ and $L_k := y_k x_k^*$ for $k = 0, 1, \ldots$, where x_k, y_k come from Step 4 in the k-th iteration of Algorithm 2. In addition, define $E_k := \Delta_k - \Delta_{\rm opt}$ and $F_k := L_k - L$. We first find a relation between $\|F_k\|_2$ and $\|E_k\|_2$. Consider the matrix trajectory

$$C(t) = A + \Delta_{\rm opt} + t\frac{E_k}{\|E_k\|_2}. \tag{20}$$

Let $x(t)$, $y(t)$ be RP-compatible right and left eigenvectors of $C(t)$ associated with its rightmost eigenvalue $\lambda(t)$ such that $\lambda(0) = 0$, $x(0) = x$ and $y(0) = y$. Define $Q(t) := y(t)x(t)^*$. Invoking Theorem II.2 and taking the conjugate of (6),

$$\frac{dQ(t)}{dt}\Big|_{t=0} = \mathrm{Re}(\beta+\gamma)Q(0) - Q(0)C_1^*G^* - G^*C_1^*Q(0),$$

where $G = (A + \Delta_{\rm opt} - \lambda I)^\#$, $\beta = x^*GC_1x$ with $C_1 = E_k/\|E_k\|_2$ and $\gamma = y^*C_1Gy$. Note that, since $C(0) = A + \Delta_{\rm opt}$ and $C(\|E_k\|_2) = A + \Delta_k$, we have $Q(0) = L$ and $Q(\|E_k\|_2) = L_k$. Therefore using Taylor's expansion, we have

$$\begin{aligned} F_k = Q(\|E_k\|_2) - Q(0) &= \frac{dQ(t)}{dt}\Big|_{t=0}\|E_k\|_2 + R(\|E_k\|_2) \\ &= \mathrm{Re}(x^*GE_kx + y^*E_kGy)L - LE_k^*G^* - G^*E_k^*L \\ &\quad + R(\|E_k\|_2), \end{aligned} \tag{21}$$

where $R(\|E_k\|_2)$ is the Taylor remainder of order 1 such that $\|R(t)\|_2 = O(t^2)$, which implies the existence of a class $\mathcal{K}$ function ξ such that $\|R(t)\|_2 \le st$ for any $s > 0$ and $t \in [0, \xi(s)]$. Now notice that because x, y are RP-compatible and $L = yx^*$, $|x| = |y| = 1$ and $\|L\|_2 \le \|L\|_F = 1$. In addition, from [19, Theorem 5.5], $\|G\|_2 \le \frac{1}{\sigma_{n-1}(A-\lambda I)(y^*x)^2}$. Hence it follows from (21) that

$$\begin{aligned} \|F_k\|_2 &\le 4\|G\|_2\|E_k\|_2\|L\|_2 + \|R(\|E_k\|_2)\|_2 \\ &\le \frac{4\|E_k\|_2}{\sigma_{n-1}(A-\lambda I)(y^*x)^2} + \|R(\|E_k\|_2)\|_2 \\ &= \frac{r}{\sqrt{n}\ell\epsilon}\|E_k\|_2 + \|R(\|E_k\|_2)\|_2 \end{aligned} \tag{22}$$

where we have used the definition (17) of r in the last equality.

We are ready to establish (18) by induction. Define $r_1 := \frac{r^\dagger - r}{\sqrt{n}\ell\epsilon}$ and let $\|E_0\|_2 < \min\left\{\frac{\sqrt{n}\ell\epsilon}{r^\dagger}\delta, \xi(r_1)\right\}$, where δ is given in Lemma IV.5. The base case $k = 0$ for (18) is trivially true. Since $r^\dagger < 1$, the induction assumption for index k implies $\|E_k\|_2 < \min\left\{\frac{\sqrt{n}\ell\epsilon}{r^\dagger}\delta, \xi(r_1)\right\}$ as well. Hence, from (22),

$$\|F_k\|_2 \le \big(\frac{r}{\sqrt{n}\ell\epsilon} + r_1\big)\|E_k\|_2 = \frac{r^\dagger}{\sqrt{n}\ell\epsilon}\|E_k\|_2 < \delta. \tag{23}$$

Using Lemma IV.5, we deduce

$$\begin{aligned} &\|E_{k+1}\|_2 \le \|E_{k+1}\|_F = \|\Delta_{k+1} - \Delta_{\rm opt}\|_F \\ &\le \ell\epsilon\|\mathrm{Re}(y_kx_k^*) - \mathrm{Re}(yx^*)\|_F \le \ell\epsilon\|F_k\|_F \le \sqrt{n}\ell\epsilon\|F_k\|_2, \end{aligned}$$

where we have used $\|M\|_2 \le \|M\|_F \le \sqrt{n}\|M\|_2$. Combining this bound with the first inequality in (23) and using the induction hypothesis,

$$\|E_{k+1}\|_2 \le r^\dagger\|E_k\|_2 \le (r^\dagger)^{k+1}\|E_0\|_2, \tag{24}$$

establishing (18). It follows that $\alpha(A + \Delta_k) = \mathrm{Re}\,\lambda_k$ converges to $\alpha(A + \Delta_{\rm opt}) = \alpha_{\epsilon,\mathcal{H}}(A)$. To prove (19), we apply Lemma II.1 to the matrix trajectory (20) and conclude

$$\frac{d}{dt}\alpha(C(t))\Big|_{t=0} = \frac{\langle E_k, \mathrm{Re}(yx^*)\rangle}{\|E_k\|_2 y^*x}.$$

Again because $C(0) = A + \Delta_{\rm opt}$ and $C(\|E_k\|_2) = A + \Delta_k$, we conclude from the Taylor expansion that

$$\begin{aligned} |\alpha(A+\Delta_k) - \alpha_{\epsilon,\mathcal{H}}(A)| &= \frac{\langle E_k, \mathrm{Re}(yx^*)\rangle}{y^*x} + O(\|E_k\|_2^2) \\ &\le \frac{\sqrt{n}\|E_k\|_2\|\mathrm{Re}(yx^*)\|_F}{y^*x} + O(\|E_k\|_2^2). \end{aligned}$$

On the other hand, when the loop in Algorithm 2 terminates at the k-th iteration,

$$\begin{aligned} (1-r^\dagger)\|E_{k+1}\|_2 &\le r^\dagger(\|E_k\|_2 - \|E_{k+1}\|_2) \\ &\le r^\dagger(\|\Delta_{k+1} - \Delta_k\|_2) \le r^\dagger\,\mathrm{tol}_\Delta, \end{aligned}$$

where we use the first inequality in (24), the triangle inequality and the stopping criteria of Algorithm 2. Hence $\|E_{k+1}\|_2 \le \frac{r^\dagger}{1-r^\dagger}\,\mathrm{tol}_\Delta$, completing the proof. ■

According to Theorem IV.7, the algorithm's convergence requires r to be smaller than 1: the smaller r is, the smaller $r^\dagger$ can be, leading to faster convergence of Algorithm 2 and a smaller approximation error $|\alpha - \alpha_{\epsilon,H}(A)|$. The value of r depends on various system parameters: it is proportional to the square root of the dimension of A, inversely proportional to the second smallest singular value of $A - \lambda I$, and increases as the left and right eigenvectors of $A + \Delta_{\rm opt}$ associated with λ get closer to being orthogonal. Using arguments similar to

those in the proof of [19, Theorem 5.7], we can conclude that $r = O(\epsilon)$ and hence, as long as the energy of the structured perturbation is small enough, $r < 1$ is ensured. A smaller value of tol_Δ results in a more accurate approximation of the structured pseudospectral abscissa, cf. (19), at the cost of more iterations in Algorithm 2.

V. Measuring network resilience: structured stability radius

Here we introduce an iterative algorithm to find the structured stability radius $r_{\mathcal{H}}(A)$, providing a metric of network resilience, corresponding to question 3) of our problem statement. Our strategy makes repeated use of Algorithm 2 to find the structured pseudospectral abscissa $\alpha_{\epsilon,\mathcal{H}}(A)$ for a given energy ϵ, since the zero-crossing of this function corresponds to the structured stability radius.

A. Structured pseudospectral abscissa as a function of perturbation energy

From the definition (3), we observe that $\epsilon \mapsto \alpha_{\epsilon,\mathcal{H}}(A)$ is an increasing function and $r_{\mathcal{H}}(A)$ is its zero-crossing. We claim that the map $\epsilon \mapsto \alpha_{\epsilon,\mathcal{H}}(A)$ is locally Lipschitz and hence differentiable almost everywhere by Rademacher's theorem [36]. To show Lipschitzness, we note that eigenvalues are Lipschitz functions with respect to perturbations of matrix entries, cf. [37] (and in fact differentiable when the eigenvalue is simple). Then, in view of definition (3) and because the maximum of Lipschitz functions is Lipschitz, we conclude that $\epsilon \mapsto \alpha_{\epsilon,\mathcal{H}}(A)$ is locally Lipschitz. The following result is useful in computing derivative of $\epsilon \mapsto \alpha_{\epsilon,\mathcal{H}}(A)$ when it exists.

Lemma V.1 (Derivative of optimal value of (12) with respect to energy of the perturbation)**.** *Let $\epsilon > 0$ and $M \in \mathbb{R}^{n\times n}$. Suppose Assumption 1 holds for $\Delta_{\mathrm{opt}}(\epsilon) = [(\Delta_{\mathrm{opt}}(\epsilon))_{ij}] \in \mathcal{H}\cap\mathbb{B}_\epsilon$ and let $\eta(\epsilon, M)$ be the optimal value of the optimization problem* (12)*. Let $\overline{\mathcal{S}} := \overline{\mathcal{S}}(\epsilon, M)$, $\underline{\mathcal{S}} := \underline{\mathcal{S}}(\epsilon, M)$ be the index sets of saturation as in Proposition IV.2. Then $\epsilon \mapsto \eta(\epsilon, M)$ is Lipschitz and wherever it is differentiable, its derivative is*

$$\frac{\partial}{\partial\epsilon}\eta(\epsilon, M) = \frac{\epsilon}{\theta_{\mathrm{opt}}(\epsilon, M)},$$

where θ_{opt} is given by (15)*.*

The proof is in the Appendix. In the case when the perturbation is unstructured and without saturation constraints, θ_{opt} has the explict form given by (16) in Remark IV.3. Hence $\frac{\partial}{\partial\epsilon}\eta(\epsilon, M) = \|M\|_F$, which recovers [22, Lemma 4]. Using Lemma II.1, when $\Lambda_{\epsilon,\mathcal{H}}(A)$ has a simple, unique rightmost eigenvalue associated with the worst-case perturbation $\Delta_{\mathrm{opt}}(\epsilon)$ with RP-compatible right and left eigenvectors $x(\epsilon), y(\epsilon)$,

$$\begin{aligned}\frac{d}{d\epsilon}\alpha_{\epsilon,\mathcal{H}}(A) &= \frac{d}{dt}\alpha(A + \Delta_{\mathrm{opt}}(\epsilon + t))\big|_{t=0}\\ &= \mathrm{Re}\left(\frac{y(\epsilon)^*(\frac{d}{dt}\Delta_{\mathrm{opt}}(t)|_{t=\epsilon})x(\epsilon)}{y(\epsilon)^*x(\epsilon)}\right)\\ &= \frac{1}{y(\epsilon)^*x(\epsilon)}\langle\frac{d}{dt}\Delta_{\mathrm{opt}}(t)|_{t=\epsilon}, \mathrm{Re}(y(\epsilon)x(\epsilon)^*)\rangle\\ &= \frac{1}{y(\epsilon)^*x(\epsilon)}\frac{d}{dt}\langle\Delta_{\mathrm{opt}}(t), \mathrm{Re}(y(\epsilon)x(\epsilon)^*)\rangle|_{t=\epsilon}.\end{aligned}$$

From Theorem IV.1, $\Delta_{\mathrm{opt}}(\epsilon)$ is the maximizer of (12) with $M = \mathrm{Re}(y(\epsilon)x(\epsilon)^*)$. Thus, using Lemma V.1, we conclude

$$\frac{d}{d\epsilon}\alpha_{\epsilon,\mathcal{H}}(A) = \frac{1}{y(\epsilon)^*x(\epsilon)}\frac{\epsilon}{\theta_{\mathrm{opt}}(\epsilon, \mathrm{Re}(y(\epsilon)x(\epsilon)^*))}. \tag{25}$$

B. Iterative computation of structured stability radius

We use Newton's method [22]

$$\epsilon_{l+1} = \epsilon_l - \frac{\alpha_{\epsilon_l,\mathcal{H}}(A)}{\frac{d}{d\epsilon}\alpha_{\epsilon,\mathcal{H}}(A)\big|_{\epsilon=\epsilon_l}} \tag{26}$$

to find the zero-crossing of $\epsilon \mapsto \alpha_{\epsilon,\mathcal{H}}(A)$, which by definition (5) is the stability radius $r_{\mathcal{H}}(A)$. Note from our discussion above that $\frac{d}{d\epsilon}\alpha_{\epsilon,\mathcal{H}}(A)$ is not well defined when $\Lambda_{\epsilon,\mathcal{H}}(A)$ has multiple rightmost eigenvalues. In this case, we compute the value of the right-hand of (25) for each rightmost eigenvalue and take the minimum to be the "gradient" (which is in fact the subgradient of $\epsilon \mapsto \alpha_{\epsilon,\mathcal{H}}(A)$ with smallest norm). Consequently, substituting (25) into (26), we update ϵ using

$$\epsilon_{l+1} = \epsilon_l - \frac{(y_l^*x_l)\theta_{\mathrm{opt}}(\epsilon_l, \mathrm{Re}(y_lx_l^*))\alpha_{\epsilon_l,\mathcal{H}}(A)}{\epsilon_l} \tag{27}$$

where x_l, y_l denote the right and left eigenvectors associated with the rightmost eigenvalue in $\Lambda_{\epsilon,\mathcal{H}}(A)$ giving the smallest value of right-hand side of (25). We also observe from (27) that a fixed point of the iteration corresponds to either $y_l^*x_l = 0$, which is ruled out if the corresponding rightmost eigenvalue is simple, or $\alpha_{\epsilon_l,\mathcal{H}}(A) = 0$, in which case the iteration has found the structured stability radius. Algorithm 3 summarizes the procedure written in pseudocode. Notice that the matrix

Algorithm 3 Computation of structured stability radius

Input: $A, \epsilon_0, \mathcal{H}, \mathrm{tol}_\Delta, \mathrm{tol}_\alpha, \zeta$
Output: r

1: **repeat** $l = 0, 1, \cdots$
2: Pick $\Delta_{\mathrm{init}} \in \mathcal{H} \cap \mathbb{B}_{\epsilon_l}$
3: Run Algorithm 2 with inputs A, ϵ_l, $\mathcal{H}$, Δ_{init}, tol_α and set the outputs $\Delta_l, \alpha_l, x_l, y_l, \theta_l$
4: $\epsilon_{l+1} \leftarrow \max\left\{\epsilon_l - \frac{(y_l^*x_l)\theta_l\alpha_l}{\epsilon_l}, \zeta\epsilon_l\right\}$
5: **until** $|\alpha_l| \le \mathrm{tol}_\alpha$
6: $r \leftarrow \epsilon_l$

$\Delta_{\mathrm{init}} \in \mathcal{H} \cap \mathbb{B}_{\epsilon_l}$ selected in Step 2 is used as the initial guess of the worst-case perturbation for Algorithm 2 (cf. Step 3). In our simulations, cf. Section VI, we use either the zero matrix, a random matrix taken from the constraint set or the result from the previous algorithm iteration. In addition, in Step 4, we let ϵ_{l+1} be lower bounded by $\zeta\epsilon_l$, for some $\zeta \in (0, 1)$, in order to prevent ϵ_{l+1} from becoming negative.

The next result characterizes the output of Algorithm 3.

Theorem V.2 (Error bound for Algorithm 3)**.** *Let λ be a simple rightmost eigenvalue for the structured pseudospectrum corresponding to the structured stability radius, with RP-compatible right and left eigenvector pair x, y. Let $r^\dagger \in (0, 1)$ and suppose that for each iteration of Algorithm 3, it holds that $r < r^\dagger$, where the parameter r is defined by* (17)*. Further assume that for every iteration l, the rightmost eigenvalue of*

$\Lambda_{\epsilon_l,\mathcal{H}}(A)$ is simple and the initial guess Δ_{init} in Step 2 is close enough to $\Delta_{\text{opt}}(\epsilon_l)$ so that it falls in the region of convergence of Algorithm 2. Then, if Algorithm 3 terminates at the l_f-th iteration, it holds that

$$|\epsilon_{l_f} - r_{\mathcal{H}}(A)| = O\left(\frac{\sqrt{n}r^\dagger \|\operatorname{Re}(yx^*)\|_F}{(1-r^\dagger)y^*x}\operatorname{tol}_\Delta + \operatorname{tol}_\alpha\right). \quad (28)$$

Proof: Since the rightmost eigenvalue is simple, $\frac{d}{d\epsilon}\alpha_{\epsilon,\mathcal{H}}(A)\big|_{\epsilon=r_{\mathcal{H}}(A)}$ exists and is given by (25). Hence, using a first-order approximation on the inverse map of $\epsilon \mapsto \alpha_{\epsilon,\mathcal{H}}(A)$,

$$\epsilon_{l_f} = r_{\mathcal{H}}(A) + \frac{y^*x\theta^\dagger \alpha_{\epsilon_{l_f},\mathcal{H}}(A)}{r_{\mathcal{H}}(A)} + O(\alpha_{\epsilon_{l_f},\mathcal{H}}(A)^2), \quad (29)$$

where $\theta^\dagger := \theta_{\text{opt}}(r_{\mathcal{H}}(A), \operatorname{Re}(yx^*))$ is given by (15). In addition, $|\alpha_{\epsilon_{l_f},\mathcal{H}}(A)| \le |\alpha_{\epsilon_{l_f},\mathcal{H}}(A) - \alpha_{l_f}| + |\alpha_{l_f}| = \frac{\sqrt{n}r^\dagger\|\operatorname{Re}(yx^*)\|_F}{(1-r^\dagger)y^*x}\operatorname{tol}_\Delta + O(\operatorname{tol}_\Delta^2) + \operatorname{tol}_\alpha$, where we have used the bound (19) from Theorem IV.7. The result follows from using this fact in equation (29). ∎

Notice that l_f does not appear on the right-hand side of (28), which indicates that the final error does not depend on the total number of iterations. In other words, the error introduced by using the approximate solution computed by Algorithm 2 at each iteration does not accumulate. We also make a final remark here that while in most cases Newton's method has quadratic convergence rate, for some particular initial guesses it may yield cyclic orbits and not converge. Nevertheless, such cyclic orbits are unstable and in all our simulations we observe Algorithm 3 terminates after a finite number of timesteps.

VI. Examples

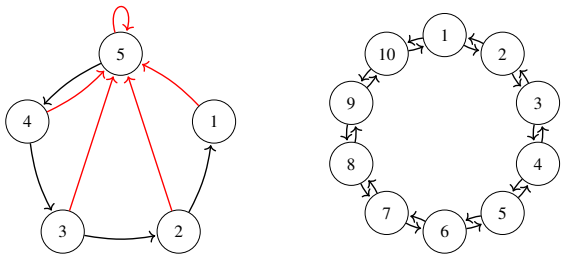
We illustrate here the use of the proposed algorithms to find the structured pseudospectral abscissa and structured stability radius. In all examples, we use the parameters $\epsilon_0 = 1$, $\operatorname{tol}_\Delta = \operatorname{tol}_\alpha = 10^{-3}$, and $\zeta = 0.1$. Our algorithms are implemented in `MATLAB` on a personal computer with 4 cores at 2.71GHz.

A. Perturbation to edges of single node of sparse network

Consider a 5-node network system with graph given in Fig. 1a. Let system matrix in (7) be given by

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -150 & -260 & -187 & -69 & -13 \end{pmatrix},$$

whose eigenvalues are $-2 \pm i, -3, -3 \pm i$. We consider additive perturbations to the edges of the node 5, i.e., $\mathcal{E}_p = \{(5,i) : i = 1,\dots,5\}$. All rows of the perturbation matrix are then zero, except for the last one which has entries $\delta = (\delta_1,\dots,\delta_5) := (\Delta_{51},\cdots,\Delta_{55})$. We consider three scenarios: (i) the entries δ_i are unconstrained, (ii) $\delta_i \le 0$ for all $i = 1,\dots,5$; and (iii) $\delta_i \ge 0$, for all $i = 1,\dots,5$. Since A is in controllable canonical form, its eigenvalues are the roots of the 5-degree polynomial $p_a(x) = x^5 + a_5x^4 + a_4x^3 + a_3x^2 + a_3x^1 + a_1$, with $a = (a_1,\cdots,a_5) = (150,\ 260,\ 187,\ 69,\ 13)$. Hence, the problem of determining the structured stability radius of A is equivalent to finding the smallest perturbation on the non-leading coefficients of $p_a(x)$ such that $p_{a-\delta}(x)$ becomes non-Hurwitz. Even though the roots of a high-degree polynomial may be sensitive to its coefficients (cf. the Wilkinson's polynomial studied in [38]), our algorithms can efficiently produce accurate structured pseudospectral abscissa and stability radius for this matrix A.



(a) (b)

Fig. 1: Network graph of (a) Section VI-A (only red edges are subject to perturbation) and (b) Section VI-B (without self-loops displayed).

1) No constraints on perturbation: Consider the case when $\mathcal{H} = \{\Delta \in \mathbb{R}^{n\times n} : \Delta_{ij} = 0 \text{ if } (i,j) \notin \mathcal{E}_p\}$. Algorithm 3, using $\Delta_{\text{init}} = 0^{5\times5}$ in Step 2, finds 10.1465 as the structured stability radius, with worst-case perturbation $\delta_{\text{opt}} = (0.0024,\ -0.0074,\ -0.1571,\ 0.4743,\ 10.1341)$. Fig. 2c shows the corresponding structured ϵ-pseudospectrum of A using random samples from $\mathcal{H} \cap \mathbb{B}_\epsilon$. Note that structured pseudospectrum is indeed touching the imaginary axis, showing that the computed value is the true structured stability radius. Since the last element $(\delta_{\text{opt}})_5$ has the largest magnitude in the worst-case perturbation, we suspect $A_{5,5}$ to be the most critical one for preserving network stability (equivalently, the coefficient in front of x^4 in $p_a(x)$ is most critical for preserving the Hurwitzness of the polynomial). Indeed if we eliminate the possibility of perturbing $A_{5,5}$, the algorithm computes the new structured stability radius to be 94.3512, which is significantly larger than those obtained by not allowing to perturb any of the other elements in the bottom row of A. Fig. 2a shows the locally Lipschitzness nature of $\epsilon \mapsto \alpha_{\epsilon,\mathcal{H}}(A)$. The non-smooth corners in $\epsilon \to \alpha_{\epsilon,\mathcal{H}}(A)$ correspond to the case when $\Lambda_{\epsilon,\mathcal{H}}(A)$ has multiple rightmost eigenvalues, see Figs. 2b through 2d.

2) Non-positive constraints on perturbation: We next consider the case when $\mathcal{H} = \{\Delta \in \mathbb{R}^{n\times n} : \Delta_{ij} = 0 \text{ if } (i,j) \notin \mathcal{E}_p \text{ and } \Delta_{5j} \le 0, j \in \{1,\dots,5\}\}$. This corresponds to increasing the value of the coefficients of $p_a(x)$. Algorithm 3 finds 24.1733 as the structured stability radius, with a worst-case perturbation $\delta_{\text{opt}} = (-1.4657,\ -2.6928,\ 0,\ 0,\ -23.9583)$. Fig. 3b shows the corresponding structured ϵ-pseudospectrum of A using random samples from $\mathcal{H} \cap \mathbb{B}_\epsilon$. In this case, we observe that using $\Delta_{\text{init}} = 0^{5\times5}$ in Step 2 of Algorithm 3 does not always lead to convergence to a global optimizer of problem (11) when executing Algorithm 2 (cf. blue curve in Fig. 3). Instead, we make the selection

$$\Delta_{\text{init}} = \min_{\Delta \in \mathcal{H}\cap\mathbb{B}_{\epsilon_l}} \|\Delta - \Delta_{l-1}\|_F, \quad (30)$$

which takes the previous estimated worst-case perturbation as an initial guess corresponding to the updated value of structured pseudospectral abscissa. In Fig. 3, we observe that

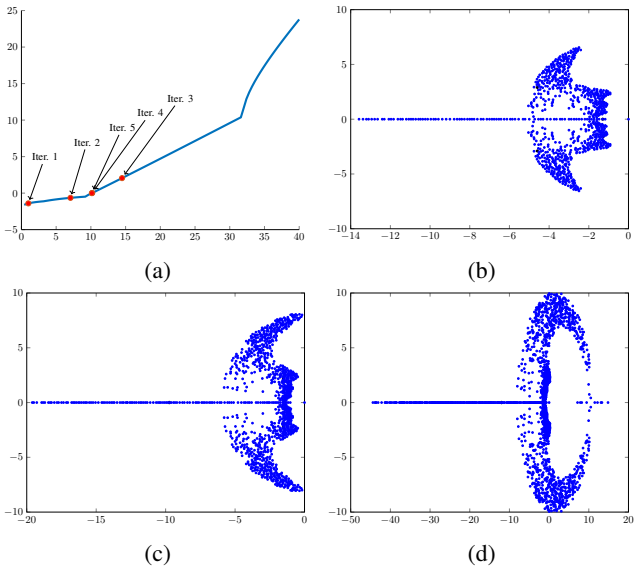


Fig. 2: Structured stability radius without constraints on perturbation. (a) Plot of $\epsilon \mapsto \alpha_{\epsilon,\mathcal{H}}(A)$ generated by uniformly sampling ϵ in $[0, 40]$ with stepsize 0.1 and running Algorithm 2 with $\Delta_{\text{init}} = 0^{5\times 5}$ for each value. The red dots correspond to the values α_l computed by Algorithm 3, which terminates after 5 iterations. Generated with 1000 random samples of $\Delta \in \mathcal{H} \cap \mathbb{B}_\epsilon$ in each case, the figures (b) (c) and (d) are the estimated $\Lambda_{\epsilon,\mathcal{H}}(A)$ with $\epsilon = 5, 10.1465$ (the structured stability radius computed by Algorithm 3) and 35, resp.

with the selection (30), the first and second iterations of Algorithm 3 correspond to local rightmost eigenvalues, but to global ones from the 3rd iteration onwards, thereby allowing the algorithm to determine the structured stability radius.

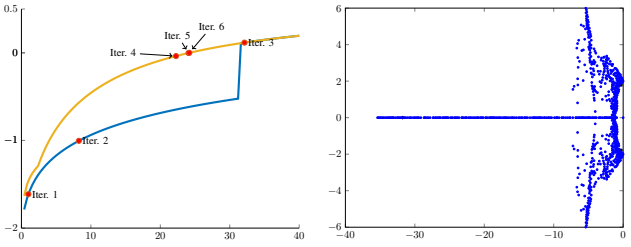


Fig. 3: Structured stability radius with non-positive constraints on perturbation. (a) The blue (resp., yellow) curve corresponds to the output of Algorithm 2 with uniform samples of ϵ in $[0, 40]$ with stepsize 0.1 and the initial guess $\Delta_{\text{init}} = 0^{5\times 5}$ (resp., 1000 random $\Delta_{\text{init}} \in \mathcal{H} \cap \mathbb{B}_\epsilon$ and taking the maximum). The red dots correspond to the values α_l of the stability radius computed by Algorithm 3, which terminates after 6 iterations; (b) Estimated $\Lambda_{\epsilon,\mathcal{H}}(A)$ with $\epsilon = 24.1733$, generated with 1000 random samples of $\Delta \in \mathcal{H} \cap \mathbb{B}_\epsilon$.

3) Non-negative constraints on perturbation: Lastly, the case when $\mathcal{H} = \{\Delta \in \mathbb{R}^{n\times n} : \Delta_{ij} = 0 \text{ if } (i,j) \notin \mathcal{E}_p \text{ and } \Delta_{5j} \geq 0\,, j \in \{1,\ldots,5\}\}$ yields almost identical result as the scenario without constraints, with 10.1478 as the structured stability radius and $\delta_{\text{opt}} = (0.0024,\ 0,\ 0,\ 0.474,\ 10.1367)$ as worst-case perturbation.

B. Circulant network

Consider a network system of 10 agents with circulant graph given in Fig. 1b. Each agent's state is 1-dimensional and obeys the simple dynamics $\dot{x}_i = u_i$, where u_i is the control input. Here, we consider

$$u_i = x_{(i+1)} - x_{(i-1)} - 0.1x_i, \tag{31}$$

where we abuse notation by identifying $(i+1) = 1$ if $i = 10$ and $(i-1) = 10$ if $i = 1$. The closed-loop system is then

$$\dot{x}_i = -0.1x_i + x_{(i+1)} - x_{(i-1)},$$

which is a typical formation control problem of agents running in a circle when the state x_i corresponds to the angle of agent i with respect to the center of rotation. The corresponding 10×10 circulant matrix is

$$A = \begin{pmatrix} -0.1 & 1 & & & & & & & & -1 \\ -1 & -0.1 & 1 & & & & & & & \\ & \ddots & \ddots & \ddots & & & & & & \\ & & & -1 & -0.1 & 1 & & & & \\ & & & & & \ddots & \ddots & \ddots & & \\ & & & & & & -1 & -0.1 & 1 & \\ 1 & & & & & & & -1 & -0.1 & \end{pmatrix}.$$

and the network graph is given in Fig. 1b.

Now suppose the first agent is compromised by an adversary so that instead of (31), it implements $u_1 = (1+\delta_1)x_2 - (1-\delta_2)x_{10} - (0.1-\delta_3)x_1$. This corresponds to perturbations in the non-zero elements of the first row of A. If $\delta = (\delta_1, \delta_2, \delta_3)$ is unconstrained, Algorithm 3 initialized at $\Delta_{\text{init}} = 0^{10\times 10}$ finds the structured stability radius 0.4727 and a worst-case perturbation $\delta_{\text{opt}} = (-0.0889, 0.0889, 0.4556)$. Fig. 4a shows the structured pseudospectrum. Note that although the estimated pseudospectrum touches the imaginary axis at multiple points in Fig. 4a, this does not violate the assumption of simple rightmost eigenvalue at the optimal point. In fact, multiple touching points on the imaginary axis mean multiple worst-case structured perturbations. For each worst-case structured perturbation Δ_{opt}, the rightmost eigenvalue of $A + \Delta_{\text{opt}}$ is still simple.

If the perturbation is constrained with $\overline{\Delta}_{1,1} = 0.1$, $\underline{\Delta}_{1,2} = -1$, and $\overline{\Delta}_{1,10} = 1$, the sign of each term in the control law (31) is preserved and Algorithm 3 initialized with $\Delta_{\text{init}} = 0^{10\times 10}$ finds the structured stability radius 1.4177 and a worst-case perturbation $\delta_{\text{opt}} = (0.1, -1, 1)$. With this worst-case perturbation, the control u_1 vanishes and the network becomes marginally stable. This can also be seen in Fig. 4b, where the structured pseudospectrum touches the imaginary axis.

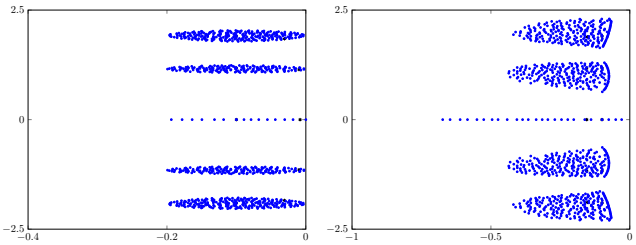
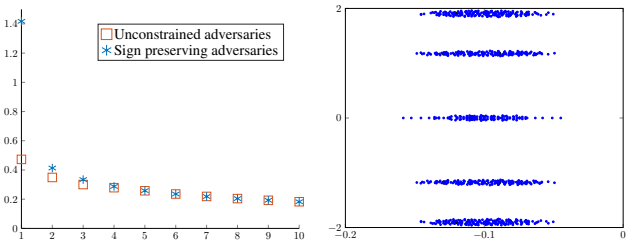


Fig. 4: Structured pseudospectra of the circulant network when one agent is subject to adversaries, generated by 2000 random sampling of $\Delta \in \mathcal{H} \cap \mathbb{B}_\epsilon$. (a) $\epsilon = 0.4727$ and the perturbation is unconstrained. (b) $\epsilon = 1.4177$ and the perturbation is sign preserving.

In scenarios where several, instead of just one, consecutive agents are compromised, Fig. 5a shows the stability radius as a function of the number of compromised agents. As this

number grows, the set $\mathcal{H}$ becomes larger and, for a fixed ϵ, $\alpha_{\epsilon,\mathcal{H}}(A)$ increases. As a result, $r_{\mathcal{H}}(A)$ decreases, which is correctly captured in Fig. 5a. Notice also that when 5 or more consecutive agents are subject to adversaries, the structured stability radii are the same independently of whether the perturbation is unconstrained or sign preserving. This is because the worst-case perturbation is distributed over all compromised edges, with entries on each edge small enough that they do not violate the boundary constraints. It is also worth mentioning that when all agents are subject to adversaries, the estimated pseudospectrum generated by randomly sampling of perturbations in $\mathcal{H} \cap \mathbb{B}_\epsilon$ does not accurately reflect the true pseudospectrum, cf. Fig. 5b, because of the sensitivity of the rightmost eigenvalue with respect to the perturbation. Nevertheless, our iterative algorithms are still able to correctly find the pseudospectral abscissa and stability radius.



(a) (b)

Fig. 5: (a) Stability radius vs. number of consecutive compromised agents. (b) Estimated $\Lambda_{\epsilon,\mathcal{H}}(A)$ when all agents are subject to unconstrained adversaries, generated by 2000 random sampling of $\Delta \in \mathcal{H} \cap \mathbb{B}_\epsilon$ with $\epsilon = 0.1826$.

C. Tolosa networks

Lastly we consider large-scale systems where the network matrix in (7) is given by a Tolosa matrix [39]. These matrices are sparse, asymmetric, and Hurwitz, and here we consider the cases of dimension $n = 90$, 340, and 1090. For each network, we consider additive perturbations to the entries of the 19th to the 36th rows ($\mathcal{E}_p = \{(i,j) : 19 \leq i \leq 36\}$), since these are the ones with the most non-zero elements. We consider two scenarios for the constraints corresponding, resp., to non-negative and non-positive perturbations, in both cases bounded in magnitude by 10,

$$\mathcal{H}_1 = \{\Delta \in [0,10]^{n\times n} : \Delta_{ij} = 0 \ \forall (i,j) \notin \mathcal{E}_p\},$$
$$\mathcal{H}_2 = \{\Delta \in [-10,0]^{n\times n} : \Delta_{ij} = 0 \ \forall (i,j) \notin \mathcal{E}_p\}.$$

We run Algorithm 3, where at each iteration we take the initial condition $\Delta_{\text{init}} = 0^{n\times n}$ and 10 random initial guesses $\Delta_{\text{init}} \in \mathcal{H} \cap \mathbb{B}_{\epsilon_l}$, and select the one that results in the maximum structured pseudospectral abscissa. Tables I and II summarize the results for each scenario.

Matrix	Dimension	$r_{\mathcal{H}}$	Iterations	Time (secs)
Tolosa90	90	0.50251	2	0.915406
Tolosa340	340	0.066404	13	58.732560
Tolosa1090	1090	0.15919	7	312.649106

TABLE I: Structured stability radii under perturbations in $\mathcal{H}_1$ obtained by Algorithm 3. The columns, from left to right, correspond to the matrix name, its dimension, the structured stability radius, the number of algorithm iterations, and the total computation time.

Matrix	Dimension	$r_{\mathcal{H}}$	Iterations	Time (secs)
Tolosa90	90	4.6737	3	2.896226
Tolosa340	340	0.35407	17	45.508739
Tolosa1090	1090	0.19163	7	213.961736

TABLE II: Structured stability radii under perturbations in $\mathcal{H}_2$ obtained by Algorithm 3. The description of the columns is the same as in Table I.

In our simulations, we notice the worst-case perturbations have a structure that is even sparser than the one specified by $\mathcal{E}_p$. For example, for the case of Tolosa90 with perturbation in $\mathcal{H}_1$, the worst-case perturbation Δ_{opt} only has 8 elements with magnitude larger than 0.001, and the largest element is $(\Delta_{\text{opt}})_{21,21} = 0.5021$, which takes about 99.84% of the total energy ($r_{\mathcal{H}}(A) = 0.50251$) of Δ_{opt}, suggesting the importance of protecting edge $(21,21) \in \mathcal{E}_p$ against structured additive topological perturbations. We also observe that for these large-scale systems, around 90% of the total computation time corresponds to the computation of the left and right eigenvectors using the `MATLAB` command `eigs`. This points out to significant room for improvement by optimizing the computation of eigenvectors of large-scale sparse matrices.

VII. Conclusions

We have studied the stability of linear dynamical systems against additive perturbations of the system matrix. We have formalized questions about whether an adversary can destabilize the network with perturbations of a given energy and determining what is the maximum amount of perturbation energy the network can withstand without becoming unstable using the concepts of structured pseudospectral abscissa and structured stability radius. We have proposed iterative algorithms that asymptotically compute both quantities for a given network along with the corresponding worst-case structured perturbations. The results of the paper open up interesting avenues of research pertaining to the identification of the most critical edges for network protection and attack. Future work will study the global convergence of the algorithms and their extension to consider arbitrary values of the perturbation energy, develop distributed strategies for the computation of the structured pseudospectral abscissa and the corresponding worst-case perturbation to help individual agents assess their relative value in ensuring network stability, and examine resilience in scenarios where adversaries only have partial knowledge of the network structure.

Appendix

Proof of Lemma IV.4: Note that Algorithm 1 terminates in a finite number of steps since the sequence of index sets of saturation $\overline{\mathcal{S}}$, $\underline{\mathcal{S}}$ is monotonically non-decreasing and uniformly bounded. In fact, the algorithm stops if the sets $\overline{\mathcal{S}}$, $\underline{\mathcal{S}}$ are unchanged or their union satisfies $\overline{\mathcal{S}} \cup \underline{\mathcal{S}} = \mathcal{E}_p$. Under Assumption 1, Algorithm 1 terminates if and only if NotDone = false, i.e., when (14a) holds. In addition, when Algorithm 1 terminates, Δ_{opt} is given by (13) for the computed $\overline{\mathcal{S}}$, $\underline{\mathcal{S}}$. Thus, to prove the statement, we are left to show that (14b) and (14c) hold. Let $\theta_{\text{opt},1}, \theta_{\text{opt},2}, \cdots, \theta_{\text{opt},k}$ be the sequence of θ_{opt} generated by Algorithm 1. From (15), it is clear that these variables are non-negative. We next show that the sequence

is non-decreasing. Take any consecutive terms $\theta_{\mathrm{opt},l}$, $\theta_{\mathrm{opt},l+1}$ and let $\delta\overline{\mathcal{S}}$ (resp. $\delta\underline{\mathcal{S}}$) be the difference between the set $\overline{\mathcal{S}}$ (resp. $\underline{\mathcal{S}}$) computed in the l-th iteration and the one computed in the next iteration. In other words, $m_{ij}\theta_{\mathrm{opt},l} \geq \overline{\Delta}_{ij}$ for all $(i,j) \in \delta\overline{\mathcal{S}}$ and $m_{ij}\theta_{\mathrm{opt},l} \leq \underline{\Delta}_{ij}$ for all $(i,j) \in \delta\underline{\mathcal{S}}$. To simplify the presentation, for the l-th iteration, let

$$\begin{aligned}
a &:= \sum_{(i,j)\in\overline{\mathcal{S}}} \overline{\Delta}_{ij}^2 + \sum_{(i,j)\in\underline{\mathcal{S}}} \underline{\Delta}_{ij}^2,\\
b &:= \sum_{(i,j)\in\delta\overline{\mathcal{S}}} \overline{\Delta}_{ij}^2 + \sum_{(i,j)\in\delta\underline{\mathcal{S}}} \underline{\Delta}_{ij}^2,\ c := \sum_{(i,j)\in\mathcal{E}_p} m_{ij}^2,\\
d &:= \sum_{(i,j)\in\overline{\mathcal{S}}} m_{ij}^2 + \sum_{(i,j)\in\underline{\mathcal{S}}} m_{ij}^2,\\
e &:= \sum_{(i,j)\in\delta\overline{\mathcal{S}}} m_{ij}^2 + \sum_{(i,j)\in\delta\underline{\mathcal{S}}} m_{ij}^2.
\end{aligned}$$

Note that (15) implies

$$\theta_{\mathrm{opt},l}^2 = \frac{\epsilon^2 - a}{c - d}, \quad \theta_{\mathrm{opt},l+1}^2 = \frac{\epsilon^2 - a - b}{c - d - e}.$$

Plugging $\theta_{\mathrm{opt},l}^2$ into $b \leq \theta_{\mathrm{opt},l}^2 e$, we have $b(c-d) \leq (\epsilon^2 - a)e$. Subtracting the expression for $\theta_{\mathrm{opt},l}^2$ from $\theta_{\mathrm{opt},l+1}^2$,

$$\theta_{\mathrm{opt},l+1}^2 - \theta_{\mathrm{opt},l}^2 = \frac{(\epsilon^2 - a)e - b(c-d)}{(c-d-e)(c-d)} \geq 0,$$

as claimed. Observe that in the execution of Algorithm 1, an edge $(i,j) \in \overline{\mathcal{S}}$ (resp. $\underline{\mathcal{S}}$) is added to the index set of saturation at some iteration $l \leq k$, when $m_{ij}\theta_{\mathrm{opt},l} \geq \overline{\Delta}_{ij} \geq 0$ (resp. $m_{ij}\theta_{\mathrm{opt},l} \leq \underline{\Delta}_{ij} \leq 0$). Since $\{\theta_{\mathrm{opt},l}\}$ is non-decreasing, we deduce $m_{ij}\theta_{\mathrm{opt},k} \geq \overline{\Delta}_{ij}$ (resp. $m_{ij}\theta_{\mathrm{opt},k} \leq \underline{\Delta}_{ij}$), thereby verifying (14b) and (14c). ∎

Proof of Lemma IV.5: Let $\theta_k = \theta_{\mathrm{opt}}(\epsilon, M_k), \quad k = 1, 2$. Notice that for some $(i,j) \in \mathcal{E}_p$, if (i,j) belongs to neither index sets of saturation for the two optimization problems, then $(\Delta_k)_{ij} = (M_k)_{ij}\theta_k$ for both $k = 1, 2$ so

$$|(\Delta_1)_{ij} - (\Delta_2)_{ij}| \leq |(M_1)_{ij}\theta_1 - (M_2)_{ij}\theta_2| \qquad (32)$$

holds with equality. If (i,j) only belongs to one index set of saturation, say $(i,j) \in \overline{\mathcal{S}}(\epsilon, M_1)$ but $(i,j) \notin \overline{\mathcal{S}}(\epsilon, M_2)$, then $(M_1)_{ij}\theta_1 \geq (\Delta_1)_{ij} = \overline{\Delta}_{ij} \geq (\Delta_2)_{ij} = (M_2)_{ij}\theta_2$ and hence again the inequality (32) holds. This is also true if (i,j) only belongs to one of the other index sets of saturation. If (i,j) belongs to the index sets of saturation for both optimization problems, then $(\Delta_1)_{ij} = (\Delta_2)_{ij}$ and (32) holds again.

Let $\delta > 0$ be the one picked in the proof of Proposition IV.2 (see the online version [35]) for defining the neighborhood $D \ni (\epsilon, M_1)$. We have $(\epsilon, M_2) \in D$ as well and using the Lipschitzness of θ_{opt},

$$\begin{aligned}
|(\Delta_1)_{ij} - (\Delta_2)_{ij}| &\leq |(M_1)_{ij}\theta_1 - (M_2)_{ij}\theta_2|\\
&\leq |(M_1)_{ij}||\theta_1 - \theta_2| + |(M_1)_{ij} - (M_2)_{ij}|\theta_2\\
&\leq \kappa\epsilon\|M_1 - M_2\|_F + \frac{\epsilon}{\underline{m}}|(M_1)ij - (M_2)_{ij}|,
\end{aligned}$$

where $\kappa = \max\{\frac{\overline{m}^2}{\underline{m}^4}, \frac{1}{(\min\{m_f, m_g\})^2}\}$ and $\overline{m}, \underline{m}, m_f, m_g$ come from the proof for the Lipschitzness of θ_{opt} in Proposition IV.2 and only depend on δ, M_1. As a result,

$$\begin{aligned}
\|\Delta_1 - \Delta_2\|_F &= \sqrt{\sum_{(i,j)\in\mathcal{E}_p} |(\Delta_1)_{ij} - (\Delta_2)_{ij}|^2}\\
&\leq \left(\sqrt{|\mathcal{E}_p|}\kappa + \frac{1}{\underline{m}}\right)\epsilon\|M_1 - M_2\|_F
\end{aligned}$$

and hence the statement holds with $\ell := \left(\sqrt{|\mathcal{E}_p|}\kappa + \frac{1}{\underline{m}}\right)$. ∎

Proof of Lemma V.1: For fixed $M \in \mathbb{R}^{n\times n}$, $\eta(\epsilon, M) = \langle \Delta_{\mathrm{opt}}, M\rangle$ is linear in Δ_{opt}, and hence Lipschitz. In addition, $\theta_{\mathrm{opt}}(\epsilon, M) \mapsto \Delta_{\mathrm{opt}}$, given by (13)-(14) is also Lipschitz. Lastly, $\epsilon \mapsto \theta_{\mathrm{opt}}(\epsilon, M)$ is Lipschitz by Proposition IV.2. Hence, the composition $\epsilon \mapsto \eta(\epsilon, M)$ is locally Lipschitz.

To find the derivative of $\epsilon \mapsto \eta(\epsilon, M)$ when it exists, we first conclude from the continuity of $\epsilon \mapsto \theta_{\mathrm{opt}}(\epsilon, M)$ and the criteria for index sets of saturation (14) that for $\delta \in \mathbb{R}$ with sufficiently small $|\delta|$, $\overline{\mathcal{S}}(\epsilon + \delta, M) \subseteq \overline{\mathcal{S}}(\epsilon, M)$ and $\underline{\mathcal{S}}(\epsilon + \delta, M) \subseteq \underline{\mathcal{S}}(\epsilon, M)$. Meanwhile, the linearity of the objective function (12) implies that when ϵ grows, the saturated elements in the optimizer remain saturated. In other words, if $\delta \geq 0$, then $\overline{\mathcal{S}}(\epsilon + \delta, M) \supseteq \overline{\mathcal{S}}(\epsilon, M)$ and $\underline{\mathcal{S}}(\epsilon + \delta, M) \supseteq \underline{\mathcal{S}}(\epsilon, M)$ Therefore, the index sets of saturation are the same for sufficiently small $\delta > 0$; i.e., $\overline{\mathcal{S}}(\epsilon+\delta, M) = \overline{\mathcal{S}}(\epsilon, M) =: \overline{\mathcal{S}}$ and $\underline{\mathcal{S}}(\epsilon + \delta, M) = \underline{\mathcal{S}}(\epsilon, M) =: \underline{\mathcal{S}}$. Thus

$$\begin{aligned}
&\eta(\epsilon + \delta, M) - \eta(\epsilon, M)\\
&\quad = \big(\theta_{\mathrm{opt}}(\epsilon, M) - \theta_{\mathrm{opt}}(\epsilon + \delta, M)\big) \sum_{(i,j)\in\mathcal{E}_p\setminus(\overline{\mathcal{S}}\cup\underline{\mathcal{S}})} m_{ij}^2.
\end{aligned}$$

This equation is useful for computing the right one-sided derivative of $\epsilon \mapsto \eta(\epsilon, M)$, which equals to the derivative of this map when it exists,

$$\begin{aligned}
\frac{d}{dt}\eta(t, M)|_{t=\epsilon} &= \lim_{\delta\to 0^+} \frac{\eta(\epsilon + \delta, M) - \eta(\epsilon, M)}{\delta}\\
&= \lim_{\delta\to 0^+} \frac{\theta_{\mathrm{opt}}(\epsilon + \delta, M) - \theta_{\mathrm{opt}}(\epsilon, M)}{\delta} \sum_{(i,j)\in\mathcal{E}_p\setminus(\overline{\mathcal{S}}\cup\underline{\mathcal{S}})} m_{ij}^2\\
&= \frac{d}{dt}\theta_{\mathrm{opt}}(t, M)\big|_{t=\epsilon} \cdot \sum_{(i,j)\in\mathcal{E}_p\setminus(\overline{\mathcal{S}}\cup\underline{\mathcal{S}})} m_{ij}^2\\
&= \epsilon\sqrt{\frac{\sum_{(i,j)\in\mathcal{E}_p\setminus(\overline{\mathcal{S}}\cup\underline{\mathcal{S}})} m_{ij}^2}{\epsilon^2 - \sum_{(i,j)\in\overline{\mathcal{S}}} \overline{\Delta}_{ij}^2 - \sum_{(i,j)\in\underline{\mathcal{S}}} \underline{\Delta}_{ij}^2}} = \frac{\epsilon}{\theta_{\mathrm{opt}}}.
\end{aligned}$$

∎

REFERENCES

[1] C. P. Pfleeger and S. L. Pfleeger, *Security in Computing*. Englewood Cliffs, NJ: Prentice Hall, 2003.

[2] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. Boca Raton, FL: CRC Press, 2008.

[3] C. Kaufman, R. Perlman, and M. Speciner, *Network Security: Private Communication in a Public World*, 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 2002.

[4] R. Patton, P. M. Frank, and R. Clark, *Fault Diagnosis in Dynamic Systems: Theory and Applications*. Englewood Cliffs, NJ: Prentice Hall, 1989.

[5] M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes: Theory and Application*. Englewood Cliffs, NJ: Prentice Hall, 1993.

[6] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation via linear iterative strategies in the presence of malicious agents," *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1495–1508, 2011.

[7] F. Pasqualetti, A. Bicchi, and F. Bullo, "Consensus computation in unreliable networks: A system theoretic approach," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 90–104, 2012.

[8] H. Zhang, E. Fata, and S. Sundaram, "A notion of robustness in complex networks," *IEEE Transactions on Automatic Control*, vol. 2, no. 3, pp. 310–320, 2015.

[9] M. Zhu and S. Martínez, "On the performance analysis of resilient networked control systems under replay attacks," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 804–808, 2014.

[10] H. Fawzi, P. Tabuada, and S. Diggavi, "Secure estimation and control for cyber-physical systems under adversarial attacks," *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1454–1467, 2014.

[11] Q. Hu, D. Fooladivanda, Y. H. Chang, and C. J. Tomlin, "Secure state estimation and control for cyber security of the nonlinear power systems," *IEEE Transactions on Automatic Control*, vol. 5, no. 3, pp. 1310–1321, 2018.

[12] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2715–2729, 2013.

[13] T. Sarkar, M. Roozbehani, and M. Dahleh, "Asymptotic network robustness," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 2, pp. 812–821, 2019.

[14] L. N. Trefethen and M. Embree, *Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators*. Princeton, NJ: Princeton University Press, 2005.

[15] N. Guglielmi and C. Lubich, "Low-rank dynamics for computing extremal points of real pseudospectra," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 1, pp. 40–66, 2013.

[16] J. V. Burke, A. S. Lewis, and M. L. Overton, "Robust stability and a criss-cross algorithm for pseudospectra," *IMA Journal of Numerical Analysis*, vol. 23, no. 3, pp. 359–375, 2003.

[17] D. Lu and B. Vandereycken, "Criss-cross type algorithms for computing the real pseudospectral abscissa," *SIAM Journal on Matrix Analysis and Applications*, vol. 38, no. 3, pp. 891–923, 2017.

[18] R. Byers, "A bisection method for measuring the distance of a stable matrix to the unstable matrices," *SIAM Journal on Scientific and Statistical Computing*, vol. 9, pp. 875–881, 1988.

[19] N. Guglielmi and M. Overton, "Fast algorithms for the approximation of the pseudospectral abscissa and pseudospectral radius of a matrix," *SIAM Journal on Matrix Analysis and Applications*, vol. 32, no. 4, pp. 1166–1192, 2011.

[20] M. W. Rostami, "New algorithms for computing the real structured pseudospectral abscissa and the real stability radius of large and sparse matrices," *SIAM Journal on Scientific Computing*, vol. 37, no. 5, pp. S447–S471, 2015.

[21] J. Vanbiervliet, B. Vandereycken, W. Michiels, S. Vandewalle, and M. Diehl, "The smoothed spectral abscissa for robust stability optimization," *SIAM Journal on Optimization*, vol. 20, no. 1, pp. 156–171, 2009.

[22] N. Guglielmi, "On the method by Rostami for computing the real stability radius of large and sparse matrices," *SIAM Journal on Scientific Computing*, vol. 38, no. 3, pp. A1662–A1681, 2016.

[23] D. Hinrichsen and A. Pritchard, "Stability radii of linear systems," *Systems & Control Letters*, vol. 7, no. 1, pp. 1–10, 1986.

[24] L. Qiu, B. Bernhardsson, A. Rantzer, E. Davison, P. Young, and J. Doyle, "A formula for computation of the real stability radius," *Automatica*, vol. 31, no. 6, pp. 879–890, 1995.

[25] A. Packard and J. Doyle, "The complex structured singular value," *Automatica*, vol. 29, no. 1, pp. 71–109, 1993.

[26] S. Poljak and J. Rohn, "Checking robust nonsingularity is NP-hard," *Mathematics of Control, Signals and Systems*, vol. 6, pp. 1–9, 1993.

[27] S. M. Rump, "Eigenvalues, pseudospectrum and structured perturbations," *Linear Algebra and its Applications*, vol. 413, no. 2, pp. 567–593, 2006.

[28] M. Karow, E. Kokiopoulou, and D. Kressner, "On the computation of structured singular values and pseudospectra," *Systems & Control Letters*, vol. 59, no. 2, pp. 122–129, 2010.

[29] P. Buttà, N. Guglielmi, and S. Noschese, "Computing the structured pseudospectrum of a Toeplitz matrix and its extreme points," *SIAM Journal on Matrix Analysis and Applications*, vol. 33, no. 4, pp. 1300–1319, 2012.

[30] S. C. Johnson, M. Wicks, M. Žefran, and R. A. DeCarlo, "The structured distance to the nearest system without property $\mathcal{P}$," *IEEE Transactions on Automatic Control*, vol. 63, no. 9, pp. 2960–2975, 2018.

[31] N. Aliyev, "Subspace method for the estimation of large-scale structured real stability radius," *arXiv preprint arXiv:2105.01001*, 2021.

[32] V. Katewa and F. Pasqualetti, "On the real stability radius of sparse systems," *Automatica*, vol. 113, p. 108685, 2020.

[33] C. D. Meyer and G. W. Stewart, "Derivatives and perturbations of eigenvectors," *Numerical Analysis*, vol. 25, pp. 679–691, 1988.

[34] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.

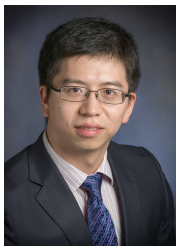
[35] S. Liu, S. Martínez, and J. Cortés, "Iterative algorithms for assessing network resilience against structured perturbations," *https://arxiv.org/abs/2105.07080*, 2021.

[36] F. H. Clarke, *Optimization and Nonsmooth Analysis*, ser. Canadian Mathematical Society Series of Monographs and Advanced Texts. Wiley, 1983.

[37] A. S. Lewis, "Nonsmooth analysis of eigenvalues," *Mathematical Programming*, vol. 84, pp. 1–24, 1999.

[38] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*. Englewood Cliffs, New Jersey: Prentice Hall, 1963.

[39] R. F. Boisvert, R. Pozo, K. Remington, R. F. Barrett, and J. J. Dongarra, "Matrix market: A web resource for test matrix collections," in *The Quality of Numerical Software: Assessment and Enhancement*. London: Chapman & Hall, 1997, pp. 125–137.



Shenyu Liu (S'16-M'20) received his B. Eng. degree in Mechanical Engineering and B.S. degree in Mathematics from the University of Singapore, Singapore, in 2014. He then received his M.S. degree in Mechanical Engineering from the University of Illinois, Urbana-Champaign in 2015, where he also received his Ph.D. degree in Electrical Engineering in 2020. He is currently a postdoctoral researcher in Department of Mechanical and Aerospace Engineering at University of California San Diego. His research interest includes matrix perturbation theory, Lyapunov methods, input-to-state stability theory, switched/hybrid systems and motion planning via geometric methods.

Sonia Martínez (M'02-SM'07-F'18) is a Professor of Mechanical and Aerospace Engineering at the University of California, San Diego, CA, USA. She received the Ph.D. degree in Engineering Mathematics from the Universidad Carlos III de Madrid, Spain, in May 2002. She was a Visiting Assistant Professor of Applied Mathematics at the Technical University of Catalonia, Spain (2002-2003) and a Postdoctoral Fulbright Fellowship at the Coordinated Science Laboratory of the University of Illinois, Urbana-Champaign (2003-2004) and the Center for Control, Dynamical systems and Computation of the University of California, Santa Barbara (2004-2005). Her research interests include the control of network systems, multi-agent systems, nonlinear control theory, and robotics. She received the Best Student Paper award at the 2002 IEEE Conference on Decision and Control for her work on the control of underactuated mechanical systems and was the recipient of a NSF CAREER Award in 2007. For the paper "Motion coordination with Distributed Information," co-authored with Jorge Cortés and Francesco Bullo, she received the 2008 Control Systems Magazine Outstanding Paper Award. She is the Editor in Chief of the recently launched Open Journal of Control Systems.

Jorge Cortés (M'02, SM'06, F'14) received the Licenciatura degree in mathematics from Universidad de Zaragoza, Zaragoza, Spain, in 1997, and the Ph.D. degree in engineering mathematics from Universidad Carlos III de Madrid, Madrid, Spain, in 2001. He held postdoctoral positions with the University of Twente, Twente, The Netherlands, and the University of Illinois at Urbana-Champaign, Urbana, IL, USA. He was an Assistant Professor with the Department of Applied Mathematics and Statistics, University of California, Santa Cruz, CA, USA, from 2004 to 2007. He is currently a Professor in the Department of Mechanical and Aerospace Engineering, University of California, San Diego, CA, USA. He is the author of Geometric, Control and Numerical Aspects of Nonholonomic Systems (Springer-Verlag, 2002) and co-author (together with F. Bullo and S. Martínez) of Distributed Control of Robotic Networks (Princeton University Press, 2009). He is a Fellow of IEEE and SIAM. His current research interests include distributed control and optimization, network science, nonsmooth analysis, reasoning and decision making under uncertainty, network neuroscience, and multi-agent coordination in robotic, power, and transportation networks.