Control principles of complex systems

Yang-Yu Liu

Channing Division of Network Medicine, Brigham and Women's Hospital, Harvard Medical School, Boston, Massachusetts 02115, USA and Center for Cancer Systems Biology, Dana-Farber Cancer Institute, Boston, Massachusetts 02115, USA

Albert-László Barabási

Center for Complex Network Research and Departments of Physics, Computer Science and Biology, Northeastern University, Boston, Massachusetts 02115, USA, Department of Medicine, Brigham and Women's Hospital, Harvard Medical School, Boston, Massachusetts 02115, USA, and Center for Network Science, Central European University, Budapest 1052, Hungary

(published 6 September 2016)

A reflection of our ultimate understanding of a complex system is our ability to control its behavior. Typically, control has multiple prerequisites: it requires an accurate map of the network that governs the interactions between the system's components, a quantitative description of the dynamical laws that govern the temporal behavior of each component, and an ability to influence the state and temporal behavior of a selected subset of the components. With deep roots in dynamical systems and control theory, notions of control and controllability have taken a new life recently in the study of complex networks, inspiring several fundamental questions: What are the control principles of complex systems? How do networks organize themselves to balance control with functionality? To address these questions here recent advances on the controllability and the control of complex networks are reviewed, exploring the intricate interplay between the network topology and dynamical laws. The pertinent mathematical results are matched with empirical findings and applications. Uncovering the control principles of complex systems can help us explore and ultimately understand the fundamental laws that govern their behavior.

DOI: 10.1103/RevModPhys.88.035006

CONTENTS

I. INTRODUCTION

To understand the mechanisms governing the behavior of a complex system, we must be able to measure its state variables and to mathematically model the dynamics of each of the system's components. Consequently, the traditional theory of complex systems has predominantly focused on the measurement and the modeling problem. Recently, however, questions pertaining to the control of complex networks became an important research topic in statistical physics (Liu, Slotine, and Barabási, 2011a; Nepusz and Vicsek, 2012; Yan *et al.*, 2012; Cornelius, Kath, and Motter, 2013; Sun and Motter, 2013; Ruths and Ruths, 2014). This interest is driven by the challenge to understand the fundamental control principles of an arbitrary self-organized system. Indeed, there is an increasing realization that the design principles of many complex systems are genuinely determined by the need to control their behavior. For example, we cannot divorce the understanding of subcellular networks from questions on how the activity or the concentrations of genes, proteins, and other biomolecules are controlled. Similarly, the structure and the daily activity of an organization is deeply determined by governance and leadership principles. Finally, to maintain the functionality of large technological systems, like the power grid or the Internet, and to adapt their functions to the shifting needs of the users, we must solve a host of control questions. These and many similar applications have led to a burst of research activity, aiming to uncover to what degree the topology of a real network behind a complex system encodes our ability to control it.

The current advances in controlling complex systems were facilitated by progress in network science, offering a quantitative framework to understand the design principles of complex networks (Watts and Strogatz, 1998; Barabási and Albert, 1999; Albert and Barabási, 2002; Milo *et al.*, 2002; Newman, 2006; Dorogovtsev, Goltsev, and Mendes, 2008; Barabási, 2016). On one end, these advances have shown that the topologies of most real systems share numerous universal characteristics. Equally important was the realization that these universal topological features are the result of the common dynamical principles that govern their emergence and growth. At the same time we learned that the topology fundamentally affects the dynamical processes taking place on these networks, from epidemic spreading (Cohen *et al.*, 2000; Pastor-Satorras and Vespignani, 2001) to transport and flow processing (Toroczkai and Bassler, 2004), and synchronization (Nishikawa *et al.*, 2003; Wang and Slotine, 2005). Hence, it is fair to expect that the network topology of a system also affects our ability to control it.

While the term "control" is frequently used in numerous disciplines with rather diverse meanings, here we employ it in the strict mathematical sense of control theory, a highly developed interdisciplinary branch of engineering and mathematics. Control theory asks how to influence the behavior of a dynamical system with appropriately chosen inputs so that the system's output follows a desired trajectory or final state. A key notion in control theory is the *feedback* process: The difference between the actual and desired output is applied as feedback to the system's input, forcing the system's output to converge to the desired output. Feedback control has deep roots in physics and engineering. For example, the centrifugal governor, one of the first practical control devices, has been used to regulate the pressure and distance between millstones in windmills since the 17th century and was used by James Watt to maintain the steady velocity of a steam engine. The feedback mechanism relies on a system of balls rotating around an axis, with a velocity proportional to the engine velocity. When the rotational velocity increases, the centrifugal force pushes the balls farther from the axis, opening valves to let the vapor escape. This lowers the pressure inside the boiler, slowing the engine (Fig. 1). The first definitive mathematical description of the centrifugal governor used in Watt's steam engine was provided by James Maxwell in 1867, proposing some of the best-known feedback control mechanisms in use today (Maxwell, 1867).

The subsequent need to design well-controlled engineered systems has resulted in a mathematically sophisticated array of control theoretical tools, which are today widely applied in the design of electric circuits, manufacturing processes, communication systems, airplanes, spacecrafts, and robots. Furthermore, since issues of regulation and control are central to the study of biological and biochemical systems, the

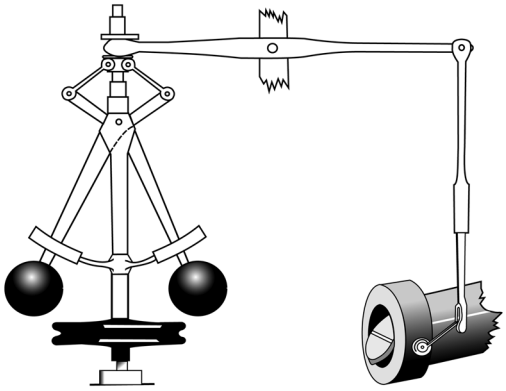

FIG. 1. Feedback control. A centrifugal governor represents a practical realization of a feedback process designed to control the speed of an engine. It uses velocity-dependent centrifugal force to regulate the release of fuel (or working fluid), maintaining a near-constant speed of the engine. It has been frequently used in steam engines, regulating the admission of steam into the cylinder(s).

concepts and tools developed in control theory have proven useful in the study of biological mechanisms and disease treatment (Sontag, 2004; Iglesias and Ingalls, 2009). For example, feedback control by transcranial electrical stimulation has been used to restore the aberrant brain activity during epileptic seizures (Berényi *et al.*, 2012).

Modern control theory heavily relies on the state space representation (also known as the "time-domain approach"), where a control system is described by a set of inputs, outputs, and state variables connected by a set of differential (or difference) equations. The concept of *state*, introduced into control theory by Rudolf Kalman in the 1960s, is a mathematical entity that mediates between the inputs and the outputs of a dynamical system, while emphasizing the notions of causality and internal structure. Any state of a dynamical system can then be represented as a vector in the state space whose axes are the state variables. The concept of the state space was inspired by the *phase space* concept used in physics, developed in the late 19th century by Ludwig Boltzmann, Henri Poincaré, and Willard Gibbs.

For a nonlinear dynamical system, we can write the state space model as

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t); \Theta), \tag{1a}$$

$$\mathbf{y}(t) = \mathbf{h}(t, \mathbf{x}(t), \mathbf{u}(t); \Theta), \tag{1b}$$

where the state vector $\mathbf{x}(t) \in \mathbb{R}^N$ represents the internal state of the system at time t, the input vector $\mathbf{u}(t) \in \mathbb{R}^M$ captures the known input signals, and the output vector $\mathbf{y}(t) \in \mathbb{R}^R$ captures the set of experimentally measured variables. The functions $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ are generally nonlinear, and Θ collects the system's parameters. Equations (1a) and (1b) are called the state and output equations, respectively, and describe the dynamics of a wide range of complex systems. For example, in metabolic networks the state vector $\mathbf{x}(t)$ represents the concentrations of all metabolites in a cell, the inputs $\mathbf{u}(t)$ represent regulatory signals modulated through enzyme abundance, and the outputs $\mathbf{y}(t)$ are experimental assays capturing the concentrations of a particular set of secreted species or the fluxes of a group of reactions of interest. In communication systems $\mathbf{x}(t)$ is the amount of information processed by a node and $\mathbf{y}(t)$ is the measurable traffic on selected links or nodes.

A significant body of work in control theory focuses on linear systems (Kailath, 1980), described by

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t), \tag{2a}$$

$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t), \tag{2b}$$

where Eqs. (2a) and (2b) represent so-called linear time-varying systems. Here $\mathbf{A}(t) \in \mathbb{R}^{N\times N}$ is the state or system matrix, telling us which components interact with each other and the strength or the nature of those interactions; $\mathbf{B}(t) \in \mathbb{R}^{N\times M}$ is the input matrix, $\mathbf{C}(t) \in \mathbb{R}^{R\times N}$ is the output matrix, and $\mathbf{D}(t) \in \mathbb{R}^{R\times M}$ is the feedthrough or feedforward matrix. In case $\mathbf{A}(t)$, $\mathbf{B}(t)$, $\mathbf{C}(t)$, and $\mathbf{D}(t)$ are constant matrices, Eqs. (2a) and (2b) represent a linear time-invariant (LTI) system, which is the starting point of most control theoretical approaches. Note that since we typically know $\mathbf{u}(t)$ and $\mathbf{D}(t)$, we can simply define a new output vector $\tilde{\mathbf{y}}(t) \equiv \mathbf{y}(t) - \mathbf{D}(t)\mathbf{u}(t) = \mathbf{C}(t)\mathbf{x}(t)$, allowing us to ignore the $\mathbf{D}(t)\mathbf{u}(t)$ term.

Many nonlinear systems like Eqs. (1a) and (1b) can be linearized around their equilibrium points, resulting in an LTI system. For example, in stick balancing, a prototypical control problem (Luenberger, 1979), our goal is to balance (or control) the stick in the upright position using the horizontal position of the hand as the control input $u(t)$. This mechanical system has a natural state space representation derived from Newton's second law of motion. Consider a stick of length L whose mass M is concentrated at the top.[1] Denote the angle between the stick and the vertical direction with $\theta(t)$. The hand and the top of the stick have horizontal displacement $u(t)$ and $x(t)$, respectively [Fig. 2(a)]. The nonlinear equation of motion for this system is

$$L\ddot{\theta}(t) = g\sin\theta(t) - \ddot{u}(t)\cos\theta(t), \tag{3}$$

where g is the gravitational constant and

$$x(t) = u(t) + L\sin\theta(t). \tag{4}$$

When the stick is nearly at rest in the upright vertical position ($\theta = 0$, which is an equilibrium point), θ is small; hence we can linearize Eqs. (3) and (4), obtaining

$$\ddot{x}(t) = \frac{g}{L}[x(t) - u(t)]. \tag{5}$$

[1]For a more realistic case, treating the stick as a rigid body of uniform density, see Stépán and Kollár (2000).

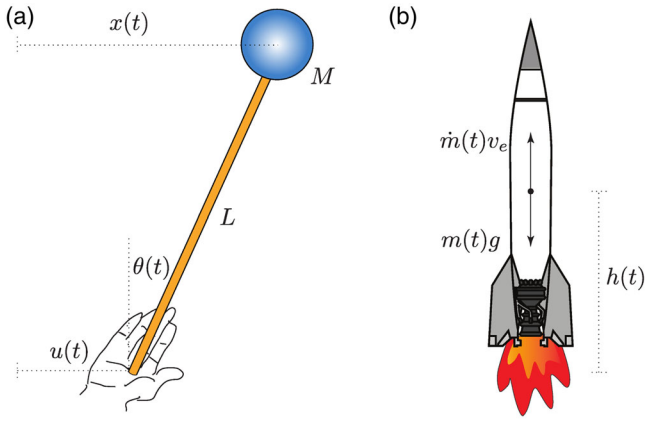

FIG. 2. Two mechanical systems whose natural state space representation with linear time-invariant (LTI) dynamics can be derived from Newton's laws of motion. (a) The goal of stick balancing, a simple but much-studied control problem (also known as the inverted perdulum problem), is to balance a stick on a palm. Adapted from Luenberger, 1979. (b) A rocket being thrust upward. The rocket ascends from the surface of the Earth with thrust force guaranteed by the ejection of mass. Adapted from Rugh, 1993.

Using the state vector $\mathbf{x}(t) = (x(t), v(t))^{\mathrm{T}}$ with velocity $v(t) = \dot{x}(t)$, and assuming $y(t) = x(t)$, we can rewrite the state and output equations in the form of an LTI system

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ \frac{g}{L} & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ -\frac{g}{L} \end{bmatrix} u(t), \tag{6a}$$

$$y(t) = [\,1 \quad 0\,]\mathbf{x}(t). \tag{6b}$$

This form allows us to perform a linear controllability analysis. Indeed, as shown in Sec. II.B, the linearized system (6a) is controllable, in line with our experience that we can balance a stick on our palm.

Linearization of a nonlinear system around its nominal trajectory $\{\mathbf{x}^*(t), \mathbf{u}^*(t)\}$ generally leads to a linear time-varying system. Consider the motion of a rocket thrust upward, following

$$m(t)\ddot{h}(t) = \dot{m}(t)v_e - m(t)g, \tag{7}$$

where $m(t)$ is the mass of the rocket at time t and $h(t)$ is its altitude. The thrust force $\dot{m}(t)v_e$ follows Newton's third law of motion, where $\dot{m}(t)$ denotes the mass flow rate and v_e is the assumed-constant exit velocity of the exhaust [Fig. 2(b)]. If we define the state vector $\mathbf{x}(t) = (h(t), v(t), m(t))^{\mathrm{T}}$ with velocity $v(t) = \dot{h}(t)$, the control input $u(t) = \dot{m}(t)$, and the output $y(t) = h(t)$, we have the state space representation

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ \frac{u(t)v_e}{x_3(t)} - g \\ u(t) \end{bmatrix}, \tag{8}$$

$$y(t) = x_1(t). \tag{9}$$

The state equation (8) is clearly nonlinear. Let us consider its linearization around a nominal trajectory that corresponds to a constant control input $u^*(t) = u_0 < 0$, i.e., a constant mass flow rate. This nominal trajectory follows $x_1^*(t) = v_e[(m_0/u_0 + t)\ln(1 + u_0 t/m_0)] - gt^2/2$, $x_2^*(t) = v_e \ln(1 + u_0 t/m_0) - gt$, and $x_3^*(t) = m_0 + u_0 t$, where m_0 is the initial mass of the rocket. By evaluating the partial derivatives $\partial\mathbf{f}(\mathbf{x}, u)/\partial\mathbf{x}$ and $\partial\mathbf{f}(\mathbf{x}, u)/\partial u$ at the nominal trajectory, we obtain the linearized state and output equations in the form of a linear time-varying system

$$\begin{aligned} \dot{\mathbf{x}}_\delta(t) &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{-u_0 v_e}{(m_0+u_0 t)^2} \\ 0 & 0 & 0 \end{bmatrix} \mathbf{x}_\delta(t) + \begin{bmatrix} 0 \\ \frac{v_e}{m_0+u_0 t} \\ 1 \end{bmatrix} u_\delta(t), \\ y_\delta(t) &= [\,1 \quad 0 \quad 0\,]\mathbf{x}_\delta(t), \end{aligned} \tag{10}$$

where the deviation variables $\mathbf{x}_\delta(t) = \mathbf{x}(t) - \mathbf{x}^*(t)$, $u_\delta(t) = u(t) - u^*(t)$, and $y_\delta(t) = y(t) - y^*(t) = \mathbf{x}_\delta(t)$.

Notwithstanding our ability to design such well-controlled systems as a car or an airplane, we continue to lack an understanding of the control principles that govern self-organized complex networked systems. Indeed, if given the wiring diagram of a cell, we do not understand the fundamental principles that govern its control, nor do we have tools to extract them. Until recently the degree of penetration of control theoretical tools in the study of complex systems was limited. The reason is that to extract the predictive power of Eqs. (1a) and (1b), we need (i) the accurate wiring diagram of the system, (ii) a description of the nonlinear dynamics that governs the interactions between the components, and (iii) a precise knowledge of the system parameters. For most complex systems we lack some of these prerequisites. For example, current estimates indicate that in human cells the available protein-protein interaction maps cover less than 20% of all potential protein-protein interactions (Sahni *et al.*, 2015). In communication systems we may be able to build an accurate wiring diagram, but we often lack the analytical form of the system dynamics $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t); \Theta)$. In biochemical reaction systems we have a good understanding of the underlying network and dynamics, but we lack the precise values of the system parameters, like the reaction rate constants. Although progress is made on all three fronts, offering increasingly accurate data on the network structure, dynamics, and the system parameters, accessing them all at once is still infeasible for most complex systems. Despite these difficulties, in the past decade we have seen significant advances pertaining to the control of complex systems. These advances indicate that many fundamental control problems can be addressed without knowing all the details of Eqs. (1a) and (1b). Hence, we do not have to wait for the description of complex systems to be complete and accurate to address and understand the control principles governing their behavior.

Graph-theoretical methods have been successfully applied to investigate the structural and qualitative properties of dynamical systems since the 1960s (Yamada and Foulds, 1990). This raises a question: Can the recent renaissance of interest in controlling networked systems offer a better

understanding of control principles than previous graph-theoretical methods? To answer this we must realize that the current interest in control in the area of complex systems is driven by the need to understand such large-scale complex networks as the Internet, the WWW, wireless communication networks, power grids, global transportation systems, genome-scale metabolic networks, protein interaction networks, and gene regulatory networks, to name only a few (Chen, 2014). Until the emergence of network science in the 21st century we lacked the mathematical tools to characterize the structure of these systems, not even mentioning their control principles. The nontrivial topology of real-world networks, uncovered and characterized in the past two decades, brings an intrinsic layer of complexity to most control problems, requiring us to rely on tools borrowed from many disciplines to address them. A typical example is the structural controllability problem of complex networks. Structural control theory developed in the 1970s offered sufficient and necessary conditions to check if any network with LTI dynamics is structurally controllable (Lin, 1974). Yet, it failed to offer an efficient algorithm to find the minimum set of driver nodes required to control the network, nor an analytical framework to estimate the fraction of driver nodes. Advances on this front became possible by mapping the control problem into well-studied network problems, like matching, and utilizing the notion of thermodynamic limit in statistical physics and the cavity method developed in spin glass theory, tools that were traditionally beyond the scope of control theory (Liu, Slotine, and Barabási, 2011a).

The goal of this article is to review the current advances in controlling complex systems, be they of biological, social, or technological in nature. To achieve this we discuss a series of topics that are essential to understand the control principles of networks, with emphasis on the impact of the network structure on control. The review is organized around several fundamental issues.

(i) Controllability: Before deciding how to control a system, we must make sure that it is possible to control it. Controllability, a key notion in modern control theory, quantifies our ability to steer a dynamical system to a desired final state in finite time. We discuss the impact of network topology on our ability to control complex networks and address some practical issues, like the energy or effort required for control.

(ii) Observability: As a dual concept of controllability, observability describes the possibility of inferring the initial state of a dynamical system by monitoring its time-dependent outputs. We discuss different methods to identify the sensor nodes, whose measurements over time enable us to infer the initial state of the whole system. We also explore a closely related concept, *identifiability*, representing our ability to determine the system's parameters through appropriate input and output measurements.

(iii) Steering complex systems to desired states or trajectories: The ultimate goal of control is to drive a complex system from its current state or trajectory to some desired final state or trajectory. This problem has applications from ecosystem management to cell reprogramming. For example, we want to design interventions that can move a cell from a disease (undesired) to a healthy (desired) state. We discuss different ways of achieving such control: (a) by applying small perturbations to a set of physically or experimentally feasible parameters, (b) via compensatory perturbations of state variables that exploit the basin of attraction of the desired final state, or (c) by mapping the control problem into a combinatorial optimization problem on the underlying network.

(iv) Controlling collective behavior: Collective behavior, a much-studied topic in modern statistical physics, can result from the coordinated local activity of many interdependent components. Examples include the emergence of flocking in mobile agents or synchronization in coupled oscillators. Controlling such processes has numerous potential applications, from the design of flocking robots (Olfati-Saber, 2006) to the treatment of Parkinson's disease (Tass *et al.*, 1998). We review a broad spectrum of methods to determine the conditions for the emergence of collective behavior and discuss pinning control as an effective control strategy.

Control problems are ubiquitous, with direct relevance to many natural, social, and technological phenomena. Hence the advances reviewed here probe our fundamental understanding of the complexity of the world surrounding us, potentially inspiring advances in numerous disciplines. Consequently, our focus is on conceptual advances and tools pertaining to control that apply to a wide range of problems emerging in physical, technological, biological, and social systems. It is this diversity of applications that makes control increasingly unavoidable in most disciplines.

II. CONTROLLABILITY OF LINEAR SYSTEMS

A system is *controllable* if we can drive it from any initial state to any desired final state in finite time (Kalman, 1963). Many mechanical problems can be formalized as controllability problems (Fig. 2). Consider, for example, the control of a rocket thrust upward. The rocket is controllable if we can find a continuous control input (thrust force) that can move the rocket from a given initial state (altitude and velocity) to a desired final state. Another example is the balancing of a stick on our hand. We know from our experience that this is possible, suggesting that the system must be controllable (Luenberger, 1979). The scientific challenge is to decide for an arbitrary dynamical system if it is controllable or not, given a set of inputs.

The current interest in the control of complex networked systems was induced by recent advances in the controllability of complex networks (Liu, Slotine, and Barabási, 2011a, 2012; Jia *et al.*, 2013; Pósfai *et al.*, 2013; Gao, Liu *et al.*, 2014), offering mathematical tools to identify the driver nodes, a subset of nodes whose direct control with appropriate signals can control the state of the full system. In general controllability is a prerequisite of control, hence understanding the topological factors of the underlying network that determine a system's controllability offers numerous insights

into the control principles of complex networked systems. As we will discuss below, thanks to a convergence of tools from control theory, network science and statistical physics, our understanding of network controllability has advanced considerably recently.

A. Linear time-invariant systems

The starting point of most control theoretical approaches is the LTI control system $(\mathbf{A}, \mathbf{B})$

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t). \tag{11}$$

Many mechanical systems can be naturally described by LTI dynamics, where the state vector captures the position and velocity of objects and the LTI dynamics is either directly derived from Newton's second law or represents some reasonable linearization of the underlying nonlinear problem, as illustrated by the stick balancing problem Eq. (5).

A significant fraction of the control theory literature deals exclusively with linear systems. There are multiple reasons for this. First, linear systems offer an accurate model for some real problems, such as consensus or agreement formation in multiagent networks, where the state of each agent captures its opinion (Tanner, 2004; Liu *et al.*, 2008; Rahmani *et al.*, 2009; Mesbahi and Egerstedt, 2010). Second, while many complex systems are characterized by nonlinear interactions between the components, the first step in any control challenge is to establish the controllability of the locally linearized system (Slotine and Li, 1991). Furthermore, for systems near their equilibrium points the linearized dynamics can actually characterize the underlying nonlinear controllability problem. Third, the nontrivial network topology of real-world complex systems brings a new layer of complexity to controllability. Before we can explore the fully nonlinear dynamical setting, which is mathematically much harder, we must understand the impact of the topological characteristics on linear controllability, serving as a prerequisite of nonlinear controllability.

Consider the LTI dynamics (11) on a directed weighted network $G(\mathbf{A})$ of N nodes (Fig. 3). The state variable $x_i(t)$ can denote the amount of traffic that passes through a node i on a communication network (Pastor-Satorras and Vespignani, 2004), or transcription factor concentration in a gene regulatory network (Lezon *et al.*, 2006). The state matrix $\mathbf{A} \coloneqq (a_{ij})_{N\times N}$ represents the weighted wiring diagram of the underlying network, where a_{ij} is the strength or weight with which node j affects or influences node i: a positive (or negative) a_{ij} means the link $(j \to i)$ is excitatory (or inhibitory), and $a_{ij} = 0$ if node j has no direct influence on node i. Consider M independent control signals $\{u_1, \ldots, u_M\}$ applied to the network. The input matrix $\mathbf{B} \coloneqq (b_{im})_{N\times M}$ identifies the nodes that are directly controlled, where b_{im} represents the strength of an external control signal $u_m(t)$ injected into node i.

The input signal $\mathbf{u}(t) = (u_1(t), \ldots, u_M(t))^{\mathrm{T}} \in \mathbb{R}^M$ can be imposed on all nodes or only a preselected subset of the nodes. In general the same signal $u_m(t)$ can drive multiple nodes. The nodes directly controlled by $\mathbf{u}(t)$ are called actuator nodes or simply *actuators*, such as nodes x_1, x_2, and x_5 in Fig. 3. The number of actuators is given by the number of nonzero elements in $\mathbf{B}$. The actuators that do not share input signals, e.g., nodes x_1 and x_2 in Fig. 3, are called driver nodes or simply *drivers*. The number of driver nodes equals the number of columns in $\mathbf{B}$.

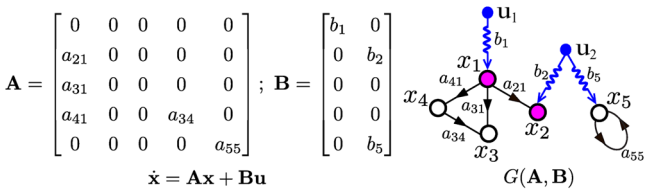

FIG. 3. Graphical representation of a linear time-invariant system (11). The state matrix $\mathbf{A}$ represents the weighted wiring diagram of the network that describes which components interact with each other and the direction of the signal or information flow for each link; the input matrix $\mathbf{B}$ identifies the nodes (state variables) that are controlled by an outside controller. The network shown in the figure is controlled by an input vector $\mathbf{u} = (u_1(t), u_2(t))^{\mathrm{T}}$ with two independent signals $u_1(t)$ and $u_2(t)$. The three actuator nodes (x_1, x_2, and x_5) are the nodes directly controlled by $\mathbf{u}(t)$. These actuator nodes correspond to the three nonzero elements in $\mathbf{B}$. The two driver nodes (x_1 and x_2), representing nodes that do not share input signals, correspond to the two columns of $\mathbf{B}$. Note that node x_5 is an actuator node, but not a driver node.

Controllability, the ability to steer a system into an arbitrary final state in a finite time, implies that we can move the state variable of each node of a network to a predefined value, corresponding to the system's desired position in the state space. Our ability to do so is largely determined by the network topology. For example, if the network structure is such that a signal cannot get from our driver nodes to a particular node, that node, and hence the system as a whole, is uncontrollable. Our challenge is to decide when control is possible and when it is not. The answer is given by controllability tests described next.

B. Kalman's criterion of controllability

Controllability tests allow us to check if an LTI system is controllable from a given set of inputs. The best known is Kalman's rank condition (Kalman, 1963), stating that the LTI system $(\mathbf{A}, \mathbf{B})$ is controllable if and only if the $N \times NM$ controllability matrix

$$\mathcal{C} \equiv [\mathbf{B}, \mathbf{A}\mathbf{B}, \mathbf{A}^2\mathbf{B}, \ldots, \mathbf{A}^{N-1}\mathbf{B}] \tag{12}$$

has full rank, i.e.,

$$\mathrm{rank}\mathcal{C} = N. \tag{13}$$

To understand the origin of Eq. (12), we consider the formal solution of Eq. (11) with $\mathbf{x}(0) = \mathbf{0}$, i.e.,

$$\mathbf{x}(t) = \int_0^t \exp[\mathbf{A}(t-\tau)]\mathbf{B}\mathbf{u}(\tau)d\tau. \tag{14}$$

If we expand $\exp[\mathbf{A}(t-\tau)]$ in series, we realize that $\mathbf{x}(t)$ is actually a linear combination of the columns in the matrices $\{\mathbf{B}, \mathbf{AB}, \mathbf{A}^2\mathbf{B}, \ldots\}$. Note that for any $N' \geq N$, we have $\text{rank}[\mathbf{B}, \mathbf{AB}, \mathbf{A}^2\mathbf{B}, \ldots, \mathbf{A}^{N'-1}\mathbf{B}] = \text{rank}\mathcal{C}$. So if $\text{rank}\mathcal{C} < N$, then even the infinite series of $\{\mathbf{B}, \mathbf{AB}, \mathbf{A}^2\mathbf{B}, \ldots\}$ will not contain a full basis to span the entire N-dimensional state space. In other words, we cannot fully explore the state space, regardless of $\mathbf{u}(t)$, indicating that given our inputs the system is stuck in a particular subspace, unable to reach an arbitrary point in the state space (Fig. 4). If, however, $\text{rank}\mathcal{C} = N$, then we can find an appropriate input vector $\mathbf{u}(t)$ to steer the system from $\mathbf{x}(0)$ to an arbitrary $\mathbf{x}(t)$. Hence, the system is controllable.

One can check that in the stick balancing problem (6a), the controllability matrix has full rank ($\text{rank}\mathcal{C} = N = 2$), indicating that the system is controllable. In the network control problem of Fig. 4(a) the controllability matrix

$$\mathcal{C} = \begin{bmatrix} b_1 & 0 & 0 \\ 0 & a_{21}b_1 & 0 \\ 0 & a_{31}b_1 & 0 \end{bmatrix} \tag{15}$$

is always rank deficient. Hence, the system is uncontrollable. By contrast, for Fig. 4(c) we have

$$\mathcal{C} = \begin{bmatrix} b_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & b_2 & a_{21}b_1 & 0 & 0 & 0 \\ 0 & 0 & a_{31}b_1 & 0 & 0 & 0 \end{bmatrix}, \tag{16}$$

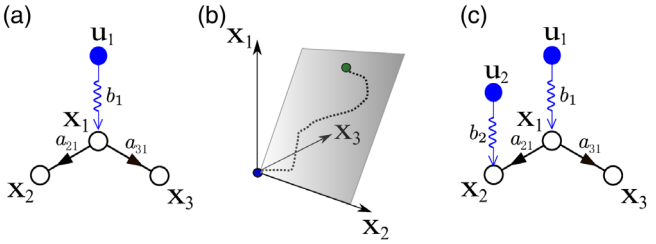

FIG. 4. Controlling star networks. (a) Controlling the central node of a directed star does not assure controllability of the whole network, as shown in Eq. (15). (b) Indeed, the system is stuck in the plane $a_{31}x_2(t) = a_{21}x_3(t)$; hence no signal $u_1(t)$ can make the system leave this plane and explore the whole state space. The reason is simple: if we change $u_1(t)$, $x_2(t)$ and $x_3(t)$ always evolve in a correlated fashion, indicating that we are unable to control the two nodes independently of each other. Note that while the system is not controllable in the whole state space, it remains controllable within the plane. It is natural that ensuring controllability within a restricted subspace will require fewer driver nodes than ensuring controllability within the whole state space (Liu, Slotine, and Barabási, 2011b; Müller and Schuppert, 2011). (c) To ensure controllability, we must inject an additional signal u_2 to either x_2 or x_3, in which case, according to Eq. (16), the network becomes controllable. From Liu, Slotine, and Barabási, 2011b.

which has full rank, as long as the parameters b_1, b_2, a_{21}, and a_{31} are nonzero. Hence the system is controllable.

The example of Fig. 4 implies that the topology of the *controlled network*, which consists of both the network itself and the control signals applied to some nodes, imposes some inherent limits on the controllability matrix: some configurations are controllable [Fig. 4(c)], while others are not [Fig. 4(a)]. Thanks to the Kalman criterion, controllability can be easily tested when the dimension of the controllability matrix is small and its rank test can be done even without knowing the detailed values of its nonzero matrix elements. For large real networks the controllability test (13) is difficult to perform, however. Indeed, there is no scalable algorithm to numerically determine the rank of the controllability matrix $\mathcal{C}$, which has dimension $N \times NM$. Equally important, executing an accurate rank test is ill conditioned and is very sensitive to round-off errors and uncertainties in the matrix elements. For example, if we plug the numerical values of b_i and a_{ij} into Eq. (12), we may obtain extremely large or small matrix elements, such as a_{ij}^{N-1}, which for large N are rather sensitive to numeric precision. Hence, for large complex systems we need to determine the system's controllability without numerically calculating the rank of the controllability matrix. As we discuss in the next section, this can be achieved in the context of structural control theory.

C. Structural controllability

For many complex networks the system parameters (e.g., the elements in $\mathbf{A}$) are not precisely known. Indeed, we are often unable to measure the weights of the links, knowing only whether there is a link or not. In other cases the links are time dependent, like the traffic on an internet cable or the flux of a chemical reaction. Hence, it is hard, if not conceptually impossible, to numerically verify Kalman's rank condition using fixed weights. Structural control, introduced by C.-T. Lin in the 1970s, offers a framework to systematically avoid this limitation (Lin, 1974).

1. The power of structural controllability

An LTI system $(\mathbf{A}, \mathbf{B})$ is a *structured system* if the elements in $\mathbf{A}$ and $\mathbf{B}$ are either fixed zeros or independent free parameters. The corresponding matrices $\mathbf{A}$ and $\mathbf{B}$ are called *structured matrices*. The system $(\mathbf{A}, \mathbf{B})$ is *structurally controllable* if we can set the nonzero elements in $\mathbf{A}$ and $\mathbf{B}$ such that the resulting system is controllable in the usual sense (i.e., $\text{rank}\mathcal{C} = N$).

The power of structural controllability comes from the fact that if a system is structurally controllable then it is controllable for almost all possible parameter realizations (Lin, 1974; Glover and Silverman, 1976; Shields and Pearson, 1976; Davison, 1977; Hosoe and Matsumoto, 1979; Mayeda, 1981; Linnemann, 1986; Reinschke, 1988; Dion, Commault, and van der Woude, 2003). To see this, denote with $\mathcal{S}$ the set of all possible LTI systems that share the same zero-nonzero connectivity pattern as a structurally controllable system $(\mathbf{A}, \mathbf{B})$. It has been shown that almost all systems that belong

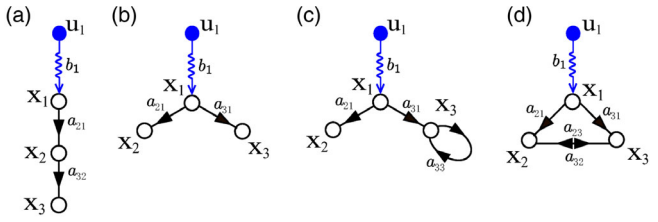

FIG. 5. Controllability, structural controllability, and strong structural controllability. (a) A directed path can be controlled by controlling the starting node only. The controllability is independent of the detailed (nonzero) values of b_1, a_{21}, and a_{32}, so the system is strongly structurally controllable. (b) A directed star can never be controlled by controlling the central hub (node x_1) only. (c) The network obtained by adding a self-edge to a leaf node x_3 in (b) can be controlled by controlling x_1 only. The controllability is independent of the detailed (nonzero) values of b_1, a_{21}, a_{31}, and a_{33}, so the system is strongly structurally controllable. (d) This network is controllable for almost all weight combinations. It will be uncontrollable only in some pathological cases, for example, when the weights satisfy the constraint $a_{32}a_{21}^2 = a_{23}a_{31}^2$ exactly. Hence, the system is structurally controllable but does not display strong structural controllability.

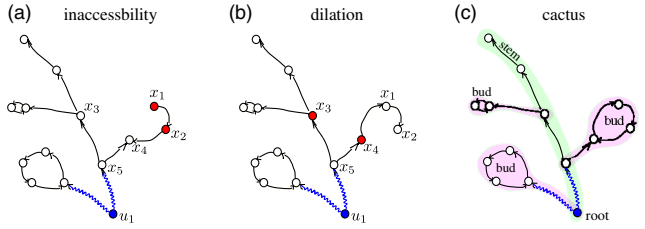

FIG. 6. Inaccessibility, dilations, and cactus. (a) The shaded (red) nodes x_1 and x_2 are inaccessible from the input node u_1 [shaded (blue)], as variations in u_1 do not influence the states of x_1 and x_2. (b) The shaded (red) nodes in the set $S = \{x_3, x_4\}$ cause a dilation. Indeed, their neighborhood set $T(S) = \{x_5\}$ contains only one node, implying that a single node in $T(S)$ aims to control two nodes in S. As shown in Eq. (15) and Fig. 4(a), this is not possible. (c) A cactus contains neither inaccessible nodes nor dilations. Note that in the cactus structure $T(S) = \{x_2, x_5\}$, hence there is no dilation. There is only one stem (green) in one cactus. There could be multiple buds (purple) in the same cactus. A cactus is a minimal structure for structural controllability.

to the set $\mathcal{S}$ are controllable except for some pathological cases with Lebesgue measure zero (Lin, 1974; Shields and Pearson, 1976). This is rooted in the fact that if a system $(\mathbf{A}_0, \mathbf{B}_0) \in \mathcal{S}$ is uncontrollable, then for every $\epsilon > 0$ there exists a controllable system $(\mathbf{A}, \mathbf{B})$ with $\|\mathbf{A} - \mathbf{A}_0\| < \epsilon$ and $\|\mathbf{B} - \mathbf{B}_0\| < \epsilon$, where $\|\cdot\|$ denotes matrix norm (Lee and Markus, 1968; Lin, 1974). In other words, an uncontrollable system in $\mathcal{S}$ becomes controllable if we slightly alter some of the link weights. For example, the system shown in Fig. 5(d) is controllable for almost all parameter realizations, except when the edge weights satisfy the constraint $a_{32}a_{21}^2 = a_{23}a_{23}^2$. But these pathological cases can be easily avoided by slightly changing one of the edge weights, hence this system is structurally controllable.

Taken together, structural control tells us that we can decide a network's controllability even if we do not know the precise weight of each edge. All we have to make sure is that we have an accurate map of the system's wiring diagram, i.e., know which components are linked and which are not. As we demonstrate in the next section, this framework considerably expands the practical applicability of control tools to real systems.

2. Graphical interpretation

Structural control theory allows us to check if a controlled network is structurally controllable by simply inspecting its topology, avoiding expensive matrix operations. This is possible thanks to the graphical interpretation[2] of Lin's structural controllability theorem, discussed next.

[2]The structural controllability theorem also has a pure algebraic meaning (Shields and Pearson, 1976), which plays an important role in the characterization of strong structural controllability (Mayeda and Yamada, 1979).

Consider an LTI system $(\mathbf{A}, \mathbf{B})$ represented by a digraph $G(\mathbf{A}, \mathbf{B}) = (V, E)$ (Fig. 3). The vertex set $V = V_A \cup V_B$ includes both the *state* vertices $V_A = \{x_1, \ldots, x_N\} \equiv \{v_1, \ldots, v_N\}$, corresponding to the N nodes of the network, and the *input* vertices $V_B = \{u_1, \ldots, u_M\} \equiv \{v_{N+1}, \ldots, v_{N+M}\}$, corresponding to the M input signals that are called the origins or roots of the digraph $G(\mathbf{A}, \mathbf{B})$. The edge set $E = E_A \cup E_B$ includes both the edges among state vertices $E_A = \{(x_j, x_i) | a_{ij} \neq 0\}$, corresponding to the links of network $\mathbf{A}$, and the edges connecting input vertices to state vertices $E_B = \{(u_m, x_i) | b_{im} \neq 0\}$. These definitions allow us to formulate a useful statement: The system $(\mathbf{A}, \mathbf{B})$ is not structurally controllable if and only if it has *inaccessible nodes* or *dilations* (Lin, 1974).

Let us consider these two cases separately. A state vertex x_i is inaccessible if there are no directed paths reaching x_i from the input vertices [Fig. 6(a)]. Consequently, an inaccessible node cannot be influenced by input signals applied to the driver nodes, making the whole network uncontrollable.

The digraph $G(\mathbf{A}, \mathbf{B})$ contains a dilation if there is a subset of nodes $S \subset V_A$ such that the neighborhood set of S, denoted as $T(S)$, has fewer nodes than S itself [Fig. 6(b)]. Here $T(S)$ is the set of vertices v_j for which there is a directed edge from v_j to some other vertex in S. Note that the input vertices are not allowed to belong to S but may belong to $T(S)$. Roughly speaking, dilations are subgraphs in which a small subset of nodes attempts to rule a larger subset of nodes. In other words, there are more "subordinates" than "superiors." A controlled network containing dilations is uncontrollable. For example, in a directed star configuration, where we wish to control all the leaves via a central node, any two leaf nodes form a dilation with the central hub. If we control the central hub only, the system remains uncontrollable because we cannot independently control the difference between the two leaf nodes' states (Fig. 4). In other words, we cannot independently control two subordinates if they share the same superior.

Taken together, Lin's structural controllability theorem states that an LTI system $(\mathbf{A}, \mathbf{B})$ is structurally controllable if and only if the digraph $G(\mathbf{A}, \mathbf{B})$ does not contain inaccessible nodes or dilations. These two conditions can be accurately checked by inspecting the topology of the digraph $G(\mathbf{A}, \mathbf{B})$ without dealing with any floating-point operations. Hence, this bypasses the numerical issues involved in evaluating Kalman's controllability rank test and also our lack of detailed knowledge on the edge weights in $G(\mathbf{A}, \mathbf{B})$.

An alternative graph-theoretical formulation of Lin's structural controllability theorem is often useful in practice. A general graph is *covered* or *spanned* by a subgraph if the subgraph and the graph have the same vertex set. Typically the spanning subgraph has only a subset of links of the original graph. For a digraph, a sequence of oriented edges $\{(v_1 \to v_2), \ldots, (v_{k-1} \to v_k)\}$, where the vertices $\{v_1, v_2, \ldots, v_k\}$ are distinct, is called an elementary path. When v_k coincides with v_1, the sequence of edges is called an elementary cycle. For the digraph $G(\mathbf{A}, \mathbf{B})$, we define the following subgraphs [Fig. 6(c)]: (i) a *stem* is an elementary path originating from an input vertex; (ii) a *bud* is an elementary cycle C with an additional edge e that ends, but does not begin, in a vertex of the cycle; (iii) a *cactus* is defined recursively: A stem is a cactus. Let C, O, and e be, respectively, a cactus, an elementary cycle that is disjoint with C, and a directed edge that connects C to O in $G(\mathbf{A}, \mathbf{B})$. Then $C \cup \{e\} \cup O$ is also a cactus. $G(\mathbf{A}, \mathbf{B})$ is spanned by cacti if there exists a set of disjoint cacti that cover all state vertices.

Note that a cactus is a minimal structure that contains neither inaccessible nodes nor dilations. That is, for a given cactus, the removal of any edge will result in either inaccessibility or dilation, hence the controllability of the cactus is lost (Fig. 6). We can now formulate Lin's structural controllability theorem as follows: An LTI system $(\mathbf{A}, \mathbf{B})$ is structurally controllable if and only if $G(\mathbf{A}, \mathbf{B})$ is spanned by cacti (Lin, 1974). Later we show that this formulation helps us design an efficient algorithm to identify a minimum set of inputs that guarantee structural controllability.

3. Strong structural controllability

The fundamental assumption of structural control is that the entries of the matrices $\mathbf{A}$ and $\mathbf{B}$ are either zeros or independent free parameters. Therefore structural control does not require knowledge of the exact values of parameters, and by avoiding floating-point operations, it is not subject to any numerical errors. However, some systems have interdependent parameters, making it uncontrollable despite the fact that it is structurally controllable. For example, Fig. 5(d) displays an LTI system that is structurally controllable, but becomes uncontrollable when the parameters satisfy the constraint $a_{32}a_{21}^2 = a_{23}a_{31}^2$. This leads to the notion of strong structural controllability: A system is strongly structurally controllable if it remains controllable for any value (other than zero) of the indeterminate parameters (Mayeda and Yamada, 1979). In other words, there is no combination of nonzero link weights that violates Kalman's criterion (13). For example, the LTI systems shown in Figs. 5(a) and 5(c) are strongly structurally controllable.

Both graph-theoretic (Mayeda and Yamada, 1979; Jarczyk, Svaricek, and Alt, 2011) and algebraic conditions (Reinschke, Svaricek, and Wend, 1992) for strong structural controllability have been studied. Unfortunately, those conditions do not lead to efficient algorithms. Recently, necessary and sufficient graph-theoretical conditions involving constrained matchings were derived (Chapman and Mesbahi, 2013). These conditions can be applied to check if an input set leads to a strongly structurally controllable network of size N with time complexity $\mathcal{O}(N^2)$. Although finding a minimum cardinality input set is proven to be nondeterministic polynomial time (NP) complete, a greedy $\mathcal{O}(N^2)$ algorithm has been developed to provide a strongly structural controllable input set, which is not necessarily minimal (Chapman and Mesbahi, 2013).

D. Minimum inputs problem

If we want to control a networked system, we first need to identify the set of driver nodes that, if driven by different signals, can offer full control over the network. Any system is fully controllable if we control each node individually. Yet, such full control is costly and typically impractical. Hence, we are particularly interested in identifying a minimum driver node set (MDNS), whose control is sufficient to make the whole system controllable. In other words, we want to control a system with minimum inputs.

1. Solution based on structural control theory

Kalman's rank condition does not offer us the MDNS—it tells us only if we can control a system through a given set of potential driver nodes that we must guess or select. Furthermore, to numerically check Kalman's rank condition, we have to know all the entries in $\mathbf{A}$ and $\mathbf{B}$, which are often unknown for complex networks. Even if we know all the weights (parameters) exactly, a brute-force search for the MDNS would require us to compute the rank of almost 2^N distinct controllability matrices, a combinatorially prohibitive task for any network of reasonable size. Yet, as we show next, we can identify the MDNS by mapping the control problem into a purely graph-theoretical problem called maximum matching (Yamada and Foulds, 1990; Commault, Dion, and van der Woude, 2002; Murota, 2009; Liu, Slotine, and Barabási, 2011a).

Matching is a widely studied problem in graph theory, with many practical applications (Lovász and Plummer, 2009). On undirected graphs, where it was originally defined, a matching represents a set of edges without common vertices [gray (red) edges in Fig. 7(g)]. Maximum matching is a matching of the largest size. For most graphs we can find multiple maximum matchings [Figs. 7(h1)–7(h3)]. The end vertices of a matching edge are called *matched*, and the remaining vertices are *unmatched*. If all vertices are matched, then the matching is called *perfect* [Fig. 7(g)].

Many real-world problems can be formalized as a maximum matching problem on bipartite graphs [Fig. 7(c)]. Consider, for example, M job applicants applying for N openings. Each applicant is interested in a subset of the openings. Each opening can accept only one applicant and

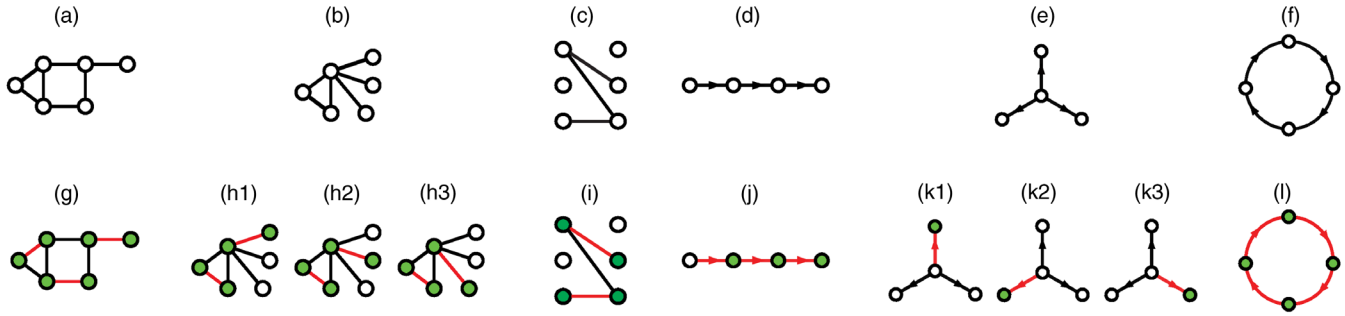

FIG. 7. Matching. The maximum matchings of (a), (b) undirected graphs, (c) a bipartite graph, and (d)–(f) digraphs. For undirected or bipartite graphs, a matching represents a set of edges without common vertices. For digraphs, a matching is a set of directed edges that do not share the common start or end vertices. Maximum matching is a matching with the largest number of edges. (g)–(l) Edges in the matching are colored in gray (red). Matched (or unmatched) nodes are shown as shaded (green) (or white), respectively.

an applicant can accept only one job offer. Finding an assignment of openings to applicants such that as many applicants as possible get a job is a classical maximum matching problem.

In structural control theory, the role of matching is well studied and matching was originally defined in the bipartite representation of a digraph (Yamada and Foulds, 1990; Commault, Dion, and van der Woude, 2002; Murota, 2009). The extended definition of matching on a digraph connects more naturally to the cactus structure (Fig. 8), which is a fundamental notion in structural control theory. In a directed graph (digraph), a matching is defined to be a set of directed edges that do not share common start or end vertices [gray (red) edges in Figs. 7(j)] (Liu, Slotine, and Barabási, 2011a). A vertex is matched if it is the end vertex of a matching edge. Otherwise, it is unmatched. For example, in a directed path, all but the starting vertex are matched [Figs. 7(d) and 7(j)]. A matching of maximum size is called a maximum matching. A maximum matching is called perfect if all vertices are matched, as in a directed elementary cycle [Figs. 7(f) and 7(l)]. We can prove that a matching of a digraph can be decomposed into a set of directed paths and/or directed cycles [Fig. 8(b)]. Note that directed paths and cycles are also the basic elements of the cactus structure [Fig. 8(d). Hence, matching in digraphs connects naturally to the cactus structure.

The usefulness of matching in network control comes from a theorem that provides the minimum number of driver nodes in a network (Liu, Slotine, and Barabási, 2011a).

a. Minimum inputs theorem

To fully control a directed network $G(\mathbf{A})$, the minimum number of inputs, or equivalently the minimum number of driver nodes, is

$$N_{\mathrm{D}} = \max\{N - |M^*|, 1\}, \tag{17}$$

where $|M^*|$ is the size of the maximum matching in $G(\mathbf{A})$. In other words, the driver nodes correspond to the unmatched nodes. If all nodes are matched ($|M^*| = N$), we need at least one input to control the network, hence $N_{\mathrm{D}} = 1$. We can choose any node as our driver node in this case.

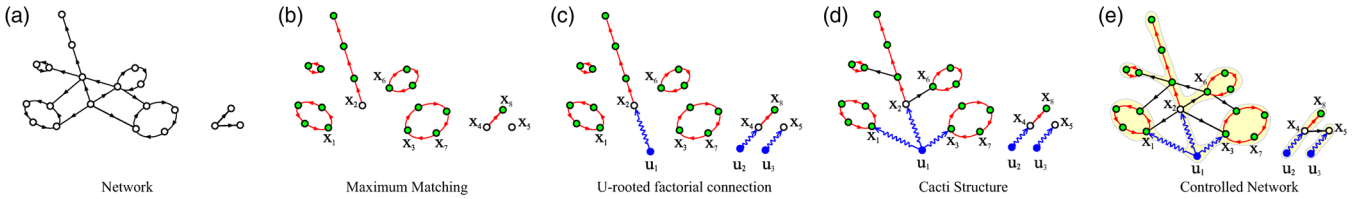

FIG. 8. Graph-theoretic proof of the minimum inputs theorem. (a) A directed network. (b) The maximum matching represents the largest set of edges without common heads or tails. All maximum matchings can be decomposed into a set of vertex-disjoint directed paths and directed cycles, shown in gray (red). If a node is the head of a matching edge, then this node is matched [shaded (green)]. Otherwise, it is unmatched (white). The unmatched nodes must be directly controlled to control the whole network, hence they are the driver nodes. (c) By injecting signals into driver nodes, we get a set of directed paths whose starting points are the input nodes. The resulting paths are called stems and the resulting digraph is called a U-rooted factorial connection. (d) By "grafting" the directed cycles to those stems, we get buds. The resulting digraph is called cactus or cacti. A cactus is a minimal structure for structural controllability as removing any of its edges will cause either inaccessible nodes or dilations. (e) According to the structural controllability theorem, since there is a cacti structure [light shading (yellow)] underlying the controlled network, the system is structurally controllable. Note that (a)–(d) also suggest an efficient method to identify the *minimal cacti*, i.e., the cacti structure with the minimum number of roots. The minimal cacti structure serves as the control skeleton that maintains the structural controllability of the system.

The minimum inputs theorem maps an inherently dynamical problem, i.e., our ability to control a network from a given subset of nodes, into a purely graph-theoretical problem of finding the maximum matching of a diagraph. Most importantly, it bypasses the need to search all node combinations for a minimum driver node set, as the driver nodes are provided by the solution of the underlying matching problem.

b. Maximum matching: Algorithmic solution

The mapping of the MDNS problem to a matching problem via Eq. (17) seems to map a problem of high computational complexity—an exhaustive search for the MDNS—into another just as complicated problem, that of finding the maximum matching for a digraph. The real value of this mapping, however, comes from the fact that the maximum matching problem in a digraph is not NP hard, but can be solved in polynomial time. Indeed, the maximum matching for a digraph can be identified by mapping the digraph to its bipartite representation, as illustrated in Fig. 9. Consider a digraph $G(\mathbf{A})$, whose bipartite representation is $H(\mathbf{A}) \equiv (V_A^+ \cup V_A^-, \Gamma)$. Here $V_A^+ = \{x_1^+, \ldots, x_N^+\}$ and $V_A^- = \{x_1^-, \ldots, x_N^-\}$ are the set of vertices corresponding to the N columns and rows of the state matrix $\mathbf{A}$, respectively. The edge set of this bipartite graph is $\Gamma = \{(x_j^+, x_i^-) | a_{ij} \neq 0\}$. In other words, we split each node x_i of the original digraph into two "nodes" x_i^+ and x_i^-. We then place an edge (x_j^+, x_i^-) in the bipartite graph if there is a directed edge $(x_j \to x_i)$ in the original digraph. Note that since we allow self-loops $(x_i \to x_i)$ in the original digraph, there can be edges of this type (x_i^+, x_i^-) in the bipartite graph. A maximum matching of a bipartite graph can be found efficiently using the Hopcroft-Karp algorithm, which runs in $O(\sqrt{V}E)$ time (Hopcroft and Karp, 1973). After running the algorithm, we can map the maximum matching in the bipartite representation, e.g., $(x_1^+, x_2^-), (x_3^+, x_3^-)$ in Fig. 9(b), back to the maximum matching in the original diagraph, e.g., $(x_1, x_2), (x_3, x_3)$ in Fig. 9(a), obtaining the desired maximum matching and hence the corresponding MDNS.

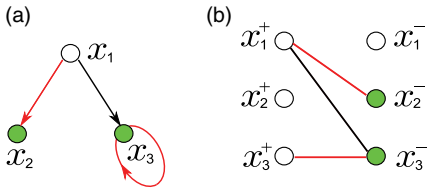

FIG. 9. Maximum matching calculation. The maximum matching of the digraph (a) can be computed from its bipartite representation (b), which is obtained by splitting each node x_i into two nodes (x_i^+ and x_i^-) and placing an edge (x_j^+, x_i^-) in the bipartite graph if there is a directed edge $(x_j \to x_i)$ in the original digraph. The maximum matching of any bipartite graph can be identified in polynomial time using the Hopcroft-Karp algorithm. Mapped back to the digraph, we obtain the maximum matching of the original digraph and the driver nodes of the corresponding control problem.

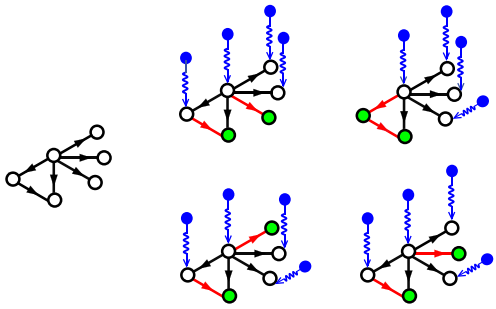

FIG. 10. Identifying the driver nodes. For a general directed network, such as the one shown in the left panel, there could be multiple maximum matchings, shown in gray (red) on the right panels. Hence, we can identify multiple MDNSs (white nodes). To each driver node we must add a unique control signal [shaded (blue)] necessary to ensure structural controllability.

Taken together, the maximum matching algorithm allows the efficient identification of the MDNS using the following steps (Fig. 10): (i) Given the directed network we want to control, we generate its bipartite representation (Fig. 9). Next identify a maximum matching on the underlying bipartite graph using the Hopcroft-Karp algorithm. (ii) To each unmatched node add a unique control signal, as unmatched nodes represent the driver nodes. (iii) As there could be multiple maximum matchings for a general digraph, multiple MDNSs exist, with the same size N_D.

Recently, several algorithmic approaches have been developed to optimize the network controllability (in the sense of decreasing N_D) via minimal structural perturbations, such as adding a minimum number of edges at judiciously chosen locations in the network (Wang *et al.*, 2012), rewiring redundant edges (Hou *et al.*, 2013), and assigning the direction of edges (Hou *et al.*, 2012; Xiao *et al.*, 2014).

c. Maximum matching: Analytical solution

While the maximum matching allows us to efficiently identify the MDNS, the algorithmic approach provides no physical insights about the impact of the network topology on N_D. For example, what network characteristics influence N_D, and how does N_D depend on them? Which networks are easier to control and which are harder? To answer these questions we turn to the cavity method, a versatile tool of statistical physics (Mézard and Parisi, 2001; Zhou and Ou-Yang, 2003; Zdeborová and Mézard, 2006). We illustrate this approach by analytically calculating $\bar{n}_D$, representing the fraction of driver nodes n_D ($\equiv N_D/N$) averaged over all network realizations compatible with the network's degree distribution $P(k_{in}, k_{out})$ (Liu, Slotine, and Barabási, 2011a). We start by describing a matching M in a digraph $G = \{V(G), E(G)\}$ by the binary variables $s_a = s_{(i\to j)} \in \{0, 1\}$ assigned to each directed edge $a = (i \to j) \in E(G)$ with $s_a = 1$ if a belongs to the matching M and $s_a = 0$ otherwise. According to the definition of matching in a digraph, matching edges do not share start or end nodes, formally resulting in two constraints for each vertex $i \in V(G)$: (i) $\sum_{j\in\partial^+ i} s_{(i\to j)} \leq 1$; (ii) $\sum_{k\in\partial^- i} s_{(k\to i)} \leq 1$

with $\partial^- i$ and $\partial^+ i$ indicating the sets of nodes that point to i or are pointed by i, respectively.

The quantity $\mathcal{E}_i(\{s\}) = 1 - \sum_{k\in\partial^- i} s_{(k\to i)}$ tells us the state of each vertex: vertex i is matched if $\mathcal{E}_i(\{s\}) = 0$ and unmatched if $\mathcal{E}_i(\{s\}) = 1$. Consequently, the cost (or energy) function gives for each matching $M = \{s\}$ the number of unmatched vertices

$$\mathcal{E}_G(\{s\}) = \sum_{i\in V(G)} \mathcal{E}_i(\{s\}) = N - |M|. \tag{18}$$

We define the Boltzmann probability in the space of matchings as

$$\mathcal{P}_G(\{s\}) = \frac{e^{-\beta\mathcal{E}_G(\{s\})}}{\mathcal{Z}_G(\beta)}, \tag{19}$$

where β is the inverse temperature and $\mathcal{Z}_G(\beta)$ is the partition function

$$\mathcal{Z}_G(\beta) = \sum_{\{s\}} e^{-\beta\mathcal{E}_G(\{s\})}. \tag{20}$$

In the limit $\beta\to\infty$ (i.e., the zero temperature limit), the internal energy $\mathcal{E}_G(\beta)$ and the entropy $\mathcal{S}_G(\beta)$ provide the ground state properties, i.e., the properties of the maximum matchings. In particular, $\mathcal{E}_G(\infty)$ represents the number of unmatched vertices (with respect to any maximum matching), and the entropy $\mathcal{S}_G(\infty)$ yields the logarithm of the number of maximum matchings.

In the zero temperature limit, the average fraction of driver nodes is given by

$$\begin{aligned}\bar{n}_D = \frac{1}{2}\Big\{&[G(\hat{w}_2) + G(1-\hat{w}_1) - 1] \\ &+ [\hat{G}(w_2) + \hat{G}(1-w_1) - 1] \\ &+ \frac{z}{2}[\hat{w}_1(1-w_2) + w_1(1-\hat{w}_2)]\Big\},\end{aligned} \tag{21}$$

where $w_1, w_2, w_3, \hat{w}_1, \hat{w}_2, \hat{w}_3$ satisfy the set of self-consistent equations

$$\begin{aligned} &w_1 = H(\hat{w}_2), \quad w_2 = 1 - H(1-\hat{w}_1), \quad w_3 = 1 - w_2 - w_1, \\ &\hat{w}_1 = \hat{H}(w_2), \quad \hat{w}_2 = 1 - \hat{H}(1-w_1), \quad \hat{w}_3 = 1 - \hat{w}_2 - \hat{w}_1, \end{aligned} \tag{22}$$

and

$$\begin{aligned} &G(x) \equiv \sum_{k_{\rm out}=0}^{\infty} P(k_{\rm out}) x^{k_{\rm out}}, \quad \hat{G}(x) \equiv \sum_{k_{\rm in}=0}^{\infty} \hat{P}(k_{\rm in}) x^{k_{\rm in}}, \\ &H(x) \equiv \sum_{k_{\rm out}=0}^{\infty} Q(k_{\rm out}+1) x^{k_{\rm out}}, \quad \hat{H}(x) \equiv \sum_{k_{\rm in}=0}^{\infty} \hat{Q}(k_{\rm in}+1) x^{k_{\rm in}} \end{aligned} \tag{23}$$

are the generating functions, and

$$Q(k_{\rm out}) \equiv \frac{k_{\rm out} P(k_{\rm out})}{\langle k_{\rm out}\rangle}, \qquad \hat{Q}(k_{\rm in}) \equiv \frac{k_{\rm in}\hat{P}(k_{\rm in})}{\langle k_{\rm in}\rangle}$$

are the out- and in-degree distributions of node i when one selects uniformly at random a directed edge $(i\to j)$ from the digraph.

While the cavity method does not offer a closed-form solution, Eq. (21) allows us to systematically study the impact of key network characteristics, such as the average degree $\langle k\rangle$ or the degree exponent γ of the underlying network, on $\bar{n}_D$ in the thermodynamic limit ($N\to\infty$) (see Fig. 11). For example, for directed Erdős-Rényi (ER) random networks (Erdős and Rényi, 1960; Bollobás, 2001), both $P(k_{\rm in})$ and $P(k_{\rm out})$ follow a Poisson distribution, i.e., $e^{-\langle k\rangle/2}(\langle k\rangle/2)^k/k!$. In the large $\langle k\rangle$ limit we have

$$n_D \sim e^{-\langle k\rangle/2}. \tag{24}$$

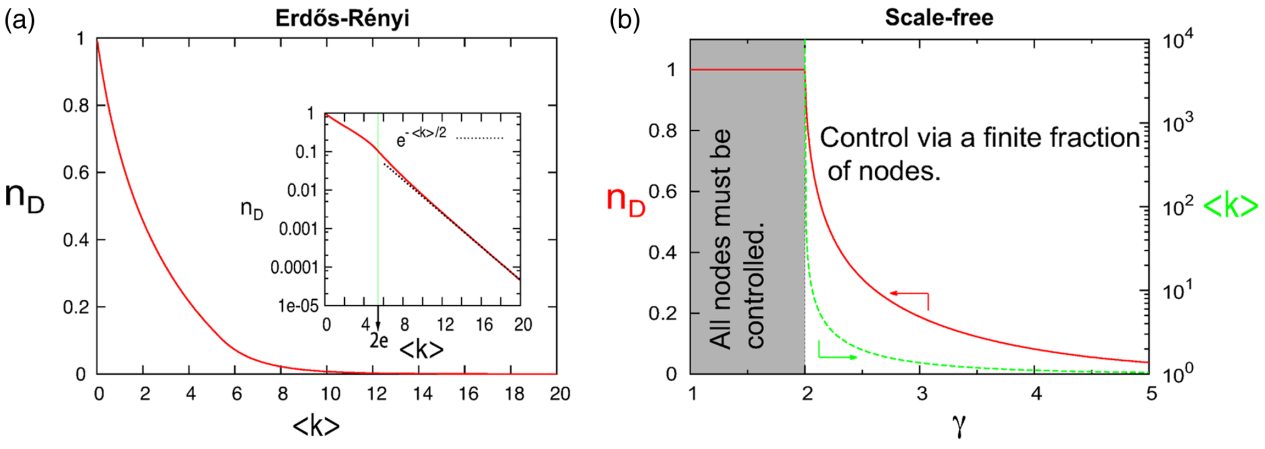

FIG. 11. Analytical results on the fraction of driver nodes ($n_D = N_D/N$) for canonical model networks. (a) For directed Erdős-Rényi random networks, n_D decays exponentially for large $\langle k\rangle$. (b) For directed scale-free networks n_D approaches 1 as the degree exponent γ approaches 2, indicating that in such networks all nodes need to be controlled.

For directed scale-free (SF) networks, we assume that $P(k_{\rm in})$ and $P(k_{\rm out})$ have the same functional form, following a power law with degree exponent γ and exponential cutoff, i.e., $P(k_{\rm in}) = Ck_{\rm in}^{-\gamma}e^{-k/\kappa}$, $P(k_{\rm out}) = Ck_{\rm out}^{-\gamma}e^{-k/\kappa}$. Here the normalization constant is $C = [\mathrm{Li}_\gamma(e^{-1/\kappa})]^{-1}$, where $\mathrm{Li}_n(x)$ is the nth polylogarithm of x. Because of the exponential cutoff $e^{-k/\kappa}$, the distribution is normalizable for any γ. One can show that as $\gamma \to 2$, we have $n_{\rm D} \to 1$. This means one has to control almost all the nodes to achieve full control over the network. Therefore $\gamma = 2$ is the critical value for the controllability of scale-free networks, as only for $\gamma > 2$ can we obtain full controllability by controlling only a subset of the nodes. Note that for $\gamma \to 2$ superhubs emerge that connect to almost all nodes in the network (Albert and Barabási, 2002; Barabási, 2016). We know that for a starlike digraph with one central hub and $N-1$ leaves, one has to control $N_{\rm D} = N-1$ nodes (the central hub and any $N-2$ leaves). In the large N limit, $N_{\rm D} \approx N$, which explains intuitively why we have to control almost all nodes when $\gamma \to 2$.

For SF networks with degree exponent $\gamma_{\rm in} = \gamma_{\rm out} = \gamma$ generated from the static model (Goh, Kahng, and Kim, 2001), the parameters $\langle k\rangle$ and γ are independent. In the thermodynamic limit the degree distribution is

$$P(k) = \frac{[m(1-\alpha)]^{1/\alpha}}{\alpha}\frac{\Gamma(k-1/\alpha, m[1-\alpha])}{\Gamma(k+1)},$$

where $\Gamma(s)$ is the gamma function and $\Gamma(s,x)$ is the upper incomplete gamma function. In the large k limit, $P(k) \sim k^{-(1+1/\alpha)} = k^{-\gamma}$, where $\gamma = 1 + 1/\alpha$. The asymptotic behavior of $n_{\rm D}(\langle k\rangle, \gamma)$ for large $\langle k\rangle$ is

$$n_{\rm D} \sim e^{-(1/2)[1-1/(\gamma-1)]\langle k\rangle}. \tag{25}$$

If $\gamma_{\rm in} \neq \gamma_{\rm out}$, the smaller of the two exponents, i.e., $\min[\gamma_{\rm in}, \gamma_{\rm out}]$ determines the asymptotic behavior of $n_{\rm D}$. Equation (25) indicates that as $\gamma \to 2$, $n_{\rm D} \to 1$, which is consistent with the result that $\gamma_{\rm c} = 2$ for a purely SF network.

The systematic dependence of $n_{\rm D}$ on $\langle k\rangle$ and γ prompts us to ask: How do other network characteristics, such as degree correlations, clustering, modularity, or the fraction of low degree nodes, influence $n_{\rm D}$ (Pósfai *et al.*, 2013; Menichetti, Dall'Asta, and Bianconi, 2014)? A combination of analytical and numerical results indicates that the clustering coefficient and modularity have no discernible effect on $n_{\rm D}$. At the same time the symmetries of the underlying matching problem generate linear, quadratic, or no dependence on degree correlation coefficients, depending on the nature of the underlying degree correlations (Pósfai *et al.*, 2013).

For uncorrelated directed networks, the density of nodes with $k_{\rm in}, k_{\rm out} = 1$ or 2 determine the size of maximum matchings (Menichetti, Dall'Asta, and Bianconi, 2014). This suggests that uncorrelated random networks whose minimum $k_{\rm in}$ and $k_{\rm out}$ are greater than 2 typically have perfect matchings and hence can be fully controlled via a single control input (i.e., $N_{\rm D} = 1$), regardless of the other properties of the degree distribution.

2. Solution based on Popov-Belevitch-Hautus controllability test

In structural control theory we assume that the system parameters, such as the link weights in $G(\mathbf{A}, \mathbf{B})$, are either fixed zeros or independent free parameters. This framework is ideal for many systems for which we know only the underlying wiring diagram (i.e., zero or nonzero values, indicating the absence or presence of physical connections) but not the link characteristics, such as their weights. Yet the independent free parameter assumption is very strong, and can be violated in some systems, such as in undirected networks, where the state matrix $\mathbf{A}$ is symmetric, or unweighted networks, where all link weights are the same. In such cases structural control theory could yield misleading results on the minimum number of driver nodes $N_{\rm D}$. Hence, it is important to move beyond structural control as we explore the controllability and other control related issues.

For LTI systems with exactly known system parameters the minimum inputs problem can be efficiently solved using the Popov-Belevitch-Hautus (PBH) controllability test. The PBH controllability test states that the system $(\mathbf{A}, \mathbf{B})$ is controllable if and only if (Hautus, 1969)

$$\mathrm{rank}[s\mathbf{I} - \mathbf{A}, \mathbf{B}] = N, \quad \forall\ s \in \mathbb{C}. \tag{26}$$

Since the first $N \times N$ block of the $N \times (N+M)$ matrix $[s\mathbf{I} - \mathbf{A}, \mathbf{B}]$ has full rank whenever s is not an eigenvalue of $\mathbf{A}$, we need only to check each eigenvalue of $\mathbf{A}$, i.e., $s \in \lambda(\mathbf{A})$, when running the PBH test.

Note that the PBH test (26) and Kalman's rank condition (13) are equivalent. Yet the advantage of the PBH test comes from the fact that it connects the controllability of $(\mathbf{A}, \mathbf{B})$ to the eigenvalues and eigenvectors of the state matrix $\mathbf{A}$. This can be used to solve the minimum inputs problem exactly. Indeed, the PBH controllability test suggests that $(\mathbf{A}, \mathbf{B})$ is controllable if and only if there is no left eigenvector of $\mathbf{A}$ orthogonal to all the columns of $\mathbf{B}$. In other words, the columns of $\mathbf{B}$ must have a component in each eigendirection of $\mathbf{A}$. Recall that for an eigenvalue $\lambda_0 \in \lambda(\mathbf{A})$, its *algebraic* multiplicity is the multiplicity of λ_0 as a root of the characteristic polynomial $p(\lambda) = \det(\mathbf{A} - \lambda\mathbf{I})$. Its *geometric* multiplicity is the maximal number of linearly independent eigenvectors corresponding to it. Hence, the number of control inputs must be greater than or equal to the largest geometric multiplicity of the eigenvalues of $\mathbf{A}$ (Antsaklis and Michel, 1997; Sontag, 1998; Yuan *et al.*, 2013). In other words, the minimum number of control inputs (or equivalently the minimum number of driver nodes) is determined by the maximum geometric multiplicity of the eigenvalues of $\mathbf{A}$, i.e.,

$$N_{\rm D} = \max_i\{\mu(\lambda_i)\}, \tag{27}$$

where $\mu(\lambda_i) = \dim V_{\lambda_i} = N - \mathrm{rank}(\lambda_i \mathbf{I}_N - \mathbf{A})$ is the geometric multiplicity of $\mathbf{A}$'s eigenvalue λ_i, representing the

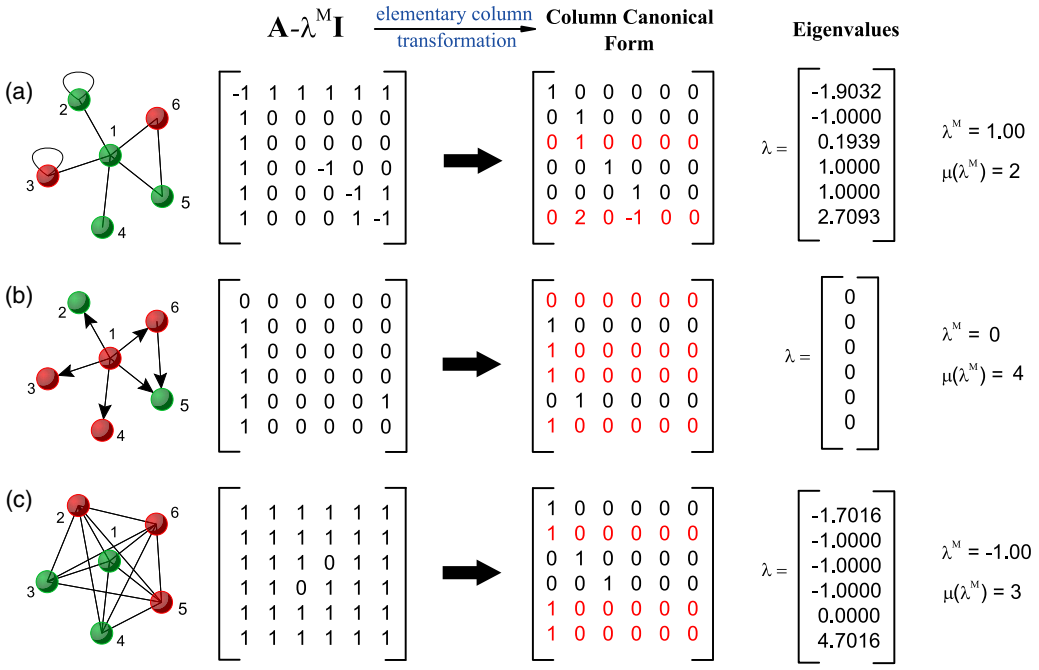

FIG. 12. Identifying a minimum set of driver nodes of small networks. For each network, we show the matrix $\mathbf{A}-\lambda^M\mathbf{I}$, its column canonical form, all eigenvalues λ of $\mathbf{A}$, and the eigenvalue λ^M with the largest geometric multiplicity. We highlight the rows that are linearly dependent on others in the column canonical form in gray (red). The corresponding nodes are the driver nodes [gray (red)] of the corresponding networks. For undirected networks in (a) and (c), $\mu(\lambda^M)$ is equal to the maximum algebraic multiplicity, that is, the multiplicity of λ^M. The configuration of driver nodes is not unique as it relies on the elementary column transformation, but the minimum number of drivers is uniquely determined by the maximum geometric multiplicity $\mu(\lambda^M)$ of matrix $\mathbf{A}$. From Yuan *et al.*, 2013.

dimension of its eigenspace. Note that the algebraic multiplicity of eigenvalue λ_i, denoted by $\delta(\lambda_i)$, is its multiplicity as a root of the characteristic polynomial. In general, $\delta(\lambda_i) \geq \mu(\lambda_i)$. But for symmetric $\mathbf{A}$, which is the case of undirected networks, we have $\delta(\lambda_i) = \mu(\lambda_i)$.

Based on Eq. (27), we can develop an efficient algorithm to identify the minimum set of driver nodes for arbitrary LTI systems (Fig. 12), allowing us to explore the impact of the network topology and link-weight distributions on N_D (Yuan *et al.*, 2013). For undirected and unweighted ER networks of connectivity probability p, the results indicate that for small p, n_D decreases with p, while for sufficiently large p, n_D increases to $(N-1)/N$, which is exact for $p=1$. For results of some special graphs, see Table I. This approach was recently extended to multiplex networks (Yuan *et al.*, 2014).

TABLE I. Eigenvalues and minimum number of driver nodes of some special graphs of N nodes. For an unweighted and undirected star and a complete graph, the table shows the algebraic multiplicity of eigenvalues in the parenthesis. From Yuan *et al.*, 2013.

Network	Eigenvalue	N_D
Chain	$2\cos\frac{q\pi}{N+1}$, $q=1,\ldots,N$	1
Ring	$2\cos\frac{2\pi(q-1)}{N}$, $q=1,\ldots,N$	2
Star	$0(N-2)$, $\pm\sqrt{N-1}(1)$	$N-2$
Complete graph	$N-1(1)$, $-1(N-1)$	$N-1$

E. Minimal controllability problems

Any networked system with LTI dynamics is fully controllable if we control each node individually with an independent signal, i.e., $M=N$. But this is costly and typically impractical for large complex systems. Hence, we are particularly interested in fully controlling a network with minimum number of nodes. Depending on the objective function and the way we "inject" input signals, we can formalize different types of *minimal controllability problems* (MCPs) (Olshevsky, 2014).

MCP0: We try to minimize the number of independent control signals, corresponding to the number of columns in the input matrix $\mathbf{B}$, or equivalently, the number of driver nodes (Liu, Slotine, and Barabási, 2011a) whose control is sufficient to fully control the system's dynamics [Fig. 13(a)]. This is nothing but the minimum inputs problem discussed in Sec. II.D.

MCP1: We assume dedicated inputs, i.e., each control input u_i can directly control only one node (state variable). In the matrix form, this amounts to finding a diagonal matrix $\mathbf{B}\in\mathbb{R}^{N\times N}$ that has as few nonzero entries as possible so that the LTI system $\dot{\mathbf{x}}=\mathbf{A}\mathbf{x}+\mathbf{B}\mathbf{u}$ is controllable [Fig. 13(b)].

MCP2: We set $u_i(t)=u(t)$ and aim to find a vector $\mathbf{b}$ that has as few nonzero entries as possible such that the system $\dot{\mathbf{x}}=\mathbf{A}\mathbf{x}+\mathbf{b}u$ is controllable [Fig. 13(c)].

Note that in solving MCP0, one signal can be applied to multiple nodes. The number of actuator nodes (corresponding

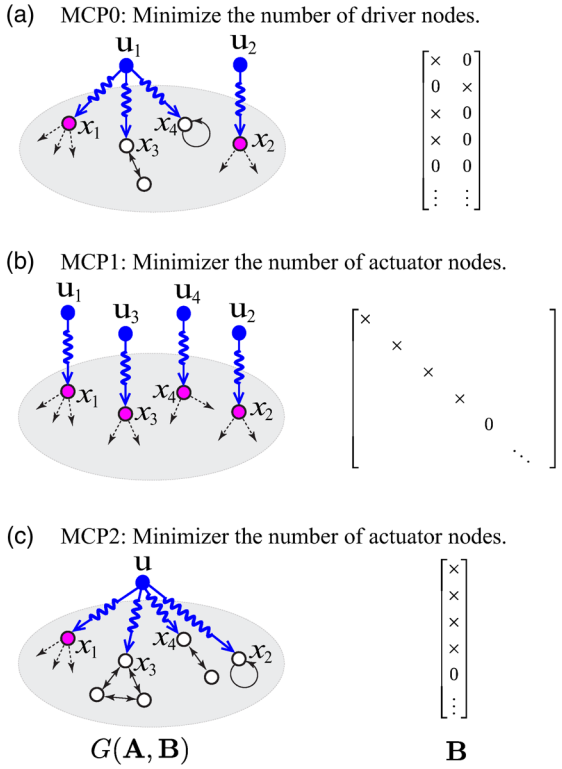

FIG. 13. Different minimal controllability problems (MCPs). For each MCP, we show the corresponding graph representation $G(\mathbf{A},\mathbf{B})$, and the input matrix $\mathbf{B}$ (where $\times$'s stand for nonzero elements. MCP0: We aim to minimize the number of driver nodes, or equivalently, the number of independent input signals. One signal can drive multiple nodes. MCP1: We aim to minimize the number of dedicated actuator nodes that receive independent input signals. One signal can drive only one actuator node. MCP2: We aim to minimize the number of actuator nodes with only one signal. This unique signal can drive multiple actuator nodes. In all cases, we assume there are four actuator nodes (x_1, x_2, x_3, and x_4). We colored the driver nodes in gray (pink).

to those nonzero entries in $\mathbf{B}$) is not necessarily minimized. In MCP1 $\mathbf{u}(t)$ is a vector of control inputs, i.e., we have multiple input signals, while in MCP2, $u(t)$ is a scalar, i.e., there is only one input signal. In both cases, we try to minimize the number of actuator nodes that are directly controlled by input signals.

Although MCP0 for a general LTI system is easy to solve, MCP1 and MCP2 are NP hard (Olshevsky, 2014). Yet, if we need to guarantee only structural controllability, MCP1 can be easily solved (Pequito, Kar, and Aguiar, 2013, 2015). For a directed network G with LTI dynamics the minimum number of dedicated inputs (or actuators) N_{da} required to assure structural controllability is

$$N_{\mathrm{da}} = N_{\mathrm{D}} + \beta - \alpha, \tag{28}$$

where N_{D} is the minimum number of driver nodes; β is the number of root strongly connected components (rSCCs), which have no incoming links from other SCCs; and α is the maximum assignability index of the bipartite representation $\mathcal{B}(G)$ of the directed network G. An rSCC is said to be a top assignable SCC if it contains at least one driver node with respect to a particular maximum matching M^*. The maximum assignability index of $\mathcal{B}(G)$ is the maximum number of top assignable SCCs that a maximum matching M^* may lead to. The minimum set of dedicated actuators can be found with polynomial-time complexity (Pequito, Kar, and Aguiar, 2013, 2015).

Consider, for example, the network shown in Fig. 3, which has in total two different maximum matchings $M_1 = \{(x_1 \to x_4), (x_4 \to x_3), (x_5 \to x_5)\}$, $M_2 = \{(x_1 \to x_2), (x_4 \to x_3), (x_5 \to x_5)\}$. Both have size 3; hence the number of driver nodes is $N_{\mathrm{D}} = \max\{N - |M^*|, 1\} = 2$, according to Eq. (17). Note that the two maximum matchings will yield two minimum sets of driver nodes, i.e., $\{x_1, x_2\}$ and $\{x_1, x_4\}$. The former is shown in Fig. 3. There are two rSCCs, $\{x_1\}$ and $\{x_5\}$, each containing a single node, hence $\beta = 2$. The rSCC $\{x_1\}$ is a top assignable SCC, because it contains one driver node with respect to either M_1 or M_2. The rSCC $\{x_5\}$ is not a top assignable SCC, because it contains no driver nodes. Hence the maximum assignability index of this system is $\alpha = 1$. Finally, the minimum number of dedicated actuators is $N_{\mathrm{da}} = N_{\mathrm{D}} + \beta - \alpha = 3$ and there are two minimum sets of actuators, i.e., $\{x_1, x_2, x_5\}$ and $\{x_1, x_4, x_5\}$.

F. Role of individual nodes and links

As seen in Sec. II.D.1, a system can be controlled by multiple driver node configurations, each corresponding to a different maximum matching (Fig. 10). Some links may appear more often in the maximum matchings than other links. This raises a fundamental question: What is the role of the individual node (or link) in control? Are some nodes (or links) more important for control than others? To answer these questions, in this section we discuss the classification of nodes and links based on their role and importance in the control of a given network (Liu, Slotine, and Barabási, 2011a; Jia *et al.*, 2013; Ruths and Ruths, 2014; Vinayagam *et al.*, 2016).

1. Link classification

In both natural and technological systems we need to quantify how robust is our ability to control a network under unavoidable link failure. To address this question, we can use structural controllability to classify each link into one of the following three categories: (1) a link is *critical* if in its absence we must increase the number of driver nodes to maintain full control over the system. In this case the link is part of all maximum matchings of the network; (2) a link is *redundant* if it can be removed without affecting the current set of driver nodes (i.e., it does not appear in any maximum matching); (3) a link is *ordinary* if it is neither critical nor redundant (it appears in some but not all maximum matchings). Note that this classification can be efficiently done with a polynomial time algorithm based on Berge's property (Régin, 1994), rather than enumerating all maximum matchings, which is infeasible for large networks.

We can compute the density of critical ($l_{\mathrm{c}} = L_{\mathrm{c}}/L$), redundant ($l_{\mathrm{r}} = L_{\mathrm{r}}/L$), and ordinary ($l_{\mathrm{o}} = L_{\mathrm{o}}/L$) links for a wide range of real-world networks. It turns out that most real networks have few or no critical links. Most links are ordinary, meaning that they play a role in some control configurations,

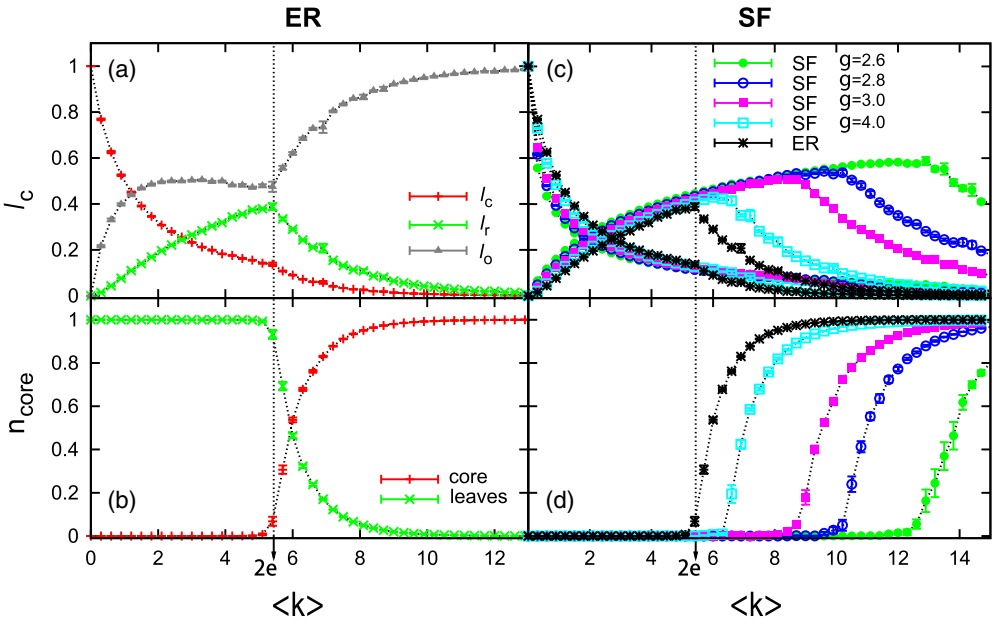

FIG. 14. Link classification and core percolation. (a) Dependence on $\langle k \rangle$ of the fraction of critical (red, l_c), redundant (green, l_r), and ordinary (gray, l_o) links for an Erdős-Rényi (ER) network: l_r peaks at $\langle k \rangle = \langle k \rangle_c = 2e$ and the derivative of l_c is discontinuous at $\langle k \rangle = \langle k \rangle_c$. (b) Core percolation for the ER network occurs at $\langle k \rangle = \langle k \rangle_c = 2e$, which explains the l_r peak. (c), (d) Same as in (a) and (b) but for scale-free networks constructed using the static model. The ER and SF networks have $N = 10^4$ nodes and the results are averaged over ten realizations with error bars defined as the standard error of the mean. Dotted lines are a guide for the eye. From Liu, Slotine, and Barabási, 2011a.

but the network can be still controlled in their absence (Liu, Slotine, and Barabási, 2011a).

For model networks (ER and SF), we can calculate l_c, l_r, and l_o as functions of $\langle k \rangle$ (Fig. 14). The behavior of l_c is easy to understand: for small $\langle k \rangle$ all links are essential for control ($l_c \approx 1$). As $\langle k \rangle$ increases the network's redundancy increases, decreasing l_c. The increasing redundancy suggests that the density of redundant links l_r should always increase with $\langle k \rangle$, but it does not: it reaches a maximum at $\langle k \rangle_c$, after which it decays. This nonmonotonic behavior results from a structural transition driven by core percolation (Liu, Slotine, and Barabási, 2011a). Here the core represents a compact cluster of nodes left in the network after applying a greedy leaf removal procedure: Recursively remove in-leaf (with $k_{in} = 1$) and out-leaf (with $k_{out} = 1$) nodes' neighbors' all outgoing (or incoming) links. The core emerges through a percolation transition [Figs. 14(b) and 14(d)]: for $k < \langle k \rangle_c$, $n_{core} = N_{core}/N = 0$, so the system consists of leaves only. At $\langle k \rangle_c$ a small core emerges, decreasing the number of leaves. For ER random networks, the analytical calculations predict $\langle k \rangle_c = 2e \approx 5.436\,564$, in agreement with the numerical result [Fig. 14(b)], a value that coincides with $\langle k \rangle$ where l_r reaches its maximum. Indeed, l_r starts decaying at $\langle k \rangle_c$ because after $\langle k \rangle_c$ the number of distinct maximum matchings increases exponentially, which can be confirmed by calculating the ground state entropy using the cavity method (Liu, Slotine, and Barabási, 2011a). Consequently, the chance that a link does not participate in any control configurations decreases. For SF networks we observe the same behavior, with the caveat that $\langle k \rangle_c$ decreases with γ [Figs. 14(c) and 14(d)].

2. Node classification

Given the existence of multiple driver node configurations, we can classify nodes based on their likelihood of being included in the MDNSs: a node is (1) *critical* if that node must always be controlled to control the system, implying that it is part of all MDNSs; (2) *redundant* if it is never required for control, implying that it never participates in an MDNS; and (3) *intermittent* if it is a driver node in some control configurations, but not in others (Jia *et al.*, 2013).

For model networks with symmetric in- and out-degree distributions, we find that the fraction of redundant nodes (n_r) undergoes a *bifurcation* at a critical mean degree $\langle k \rangle_c$: for low $\langle k \rangle$ the fraction of redundant nodes (n_r) is uniquely determined by $\langle k \rangle$, but beyond $\langle k \rangle_c$ two different solutions for n_r coexist, one with very high and the other with very low values, leading to a bimodal behavior [Fig. 15(a)]. Hence for large $\langle k \rangle$ (after the bifurcation) two control modes coexist (Jia *et al.*, 2013): (i) *Centralized control*: In networks that follow the upper branch of the bifurcation diagram most of the nodes are redundant, as in this case n_r is very high. This means that in these networks only a small fraction of the nodes are involved in control ($n_c + n_i$ is very low), hence control is guaranteed by a few nodes in the network. A good analogy would be a company involved in manufacturing whose leadership is concentrated in the hands of a few managers and the rest of the employees are only executors. (ii) *Distributed control*: In networks on the lower branch $n_c + n_i$ can exceed 90%. Hence, most nodes participate as driver nodes in some MDNSs, implying that one can engage most nodes in control. A good analogy would be an innovation-based horizontal organization, where any employee can take a leadership role, as the shifting tasks require.

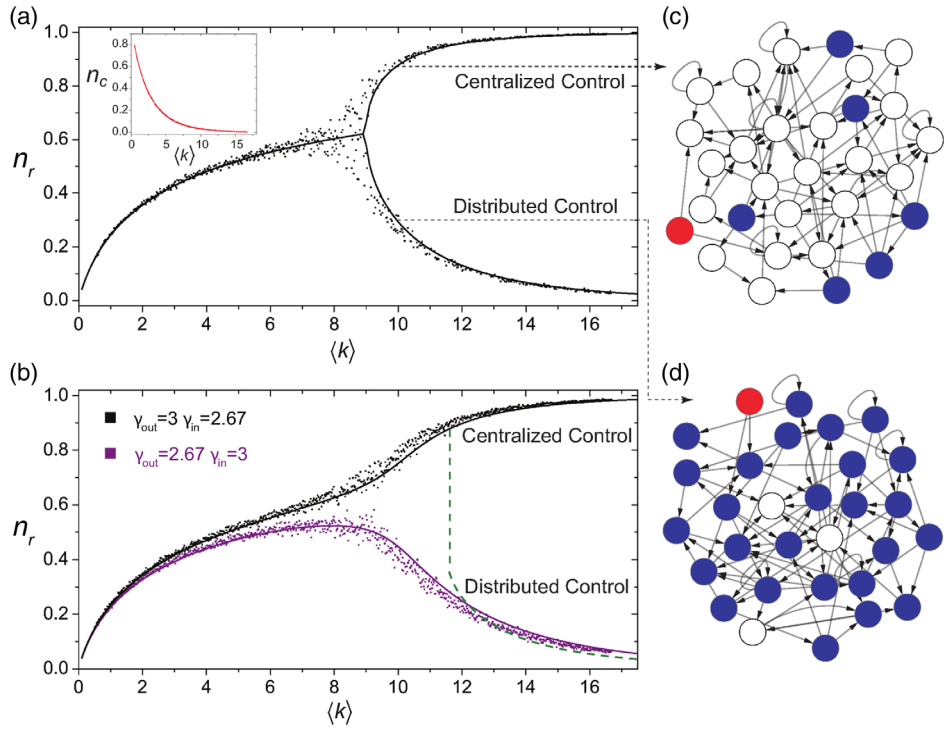

FIG. 15. Emergence of bimodality in controlling complex networks. (a) n_r and n_c (inset) vs $\langle k \rangle$ in scale-free networks with degree exponents $\gamma_{out} = \gamma_{in} = 3$, displaying the emergence of a bimodal behavior for high $\langle k \rangle$. (b) n_r in scale-free networks with asymmetric in- and out-degree distributions, i.e., $\gamma_{out} = 3$, $\gamma_{in} = 2.67$ (upper branch), and $\gamma_{out} = 2.67$, $\gamma_{in} = 3$ (lower branch). The control mode is predetermined by their degree asymmetry. (c), (d) Networks displaying centralized or distributed control. Both networks have $N_D = 4$ and $N_c = 1$ (red node), but they have a rather different number of redundant nodes (blue nodes), $N_r = 23$ in (c) and $N_r = 3$ in (d). From Jia *et al.*, 2013.

For ER random networks this bifurcation occurs at $\langle k \rangle_c = 2e$, corresponding to the core percolation threshold (Liu *et al.*, 2012).

Another way to assess a node's importance for control is to quantify the impact of its removal on controllability. Consider a network with minimum number of driver nodes N_D. After a node is removed (deleted), denote the minimum number of driver nodes with N'_D. Once again, each node can belong to one of three categories: (1) A node is *deletion critical* if in its absence we have to control more driver nodes, i.e., $N'_D > N_D$. For example, removing a node in the middle of a directed path will increase N_D. (2) A node is *deletion redundant* if in its absence we have $N'_D < N_D$. For example, removing a leaf node in a star will decrease N_D by 1. (3) A node is *deletion ordinary* if in its absence $N'_D = N_D$. For example, removing the central hub in a star will not change N_D. The above node classification has been applied to a directed human protein-protein interaction network, where the edge direction indicates signal flow (Vinayagam *et al.*, 2016). In this context critical nodes tend to correspond to disease genes, viral targets, through which a virus takes control over its host, and targets of FDA approved drugs, indicating that control-based classification can select biologically relevant proteins.

3. Driver node classification

To understand why a node is a driver node, we decompose the driver nodes (N_D) into three groups (Ruths and Ruths, 2014): (1) *source nodes* (N_s) that have no incoming links, hence they must be directly controlled, being always driver nodes; (2) *external dilations* (N_e) arise due to a surplus of sink nodes (N_t) that have no outgoing links. Since each source node can control one sink node, the number of external dilation is $N_e = \max(0, N_t - N_s)$; (3) *internal dilations* (N_i) occur when a path must branch into two or more paths in order to reach all nodes (or equivalently a subgraph has more outgoing links than incoming links). This classification leads to the control profile of a network defined as $(\eta_s, \eta_e, \eta_i) = (N_s/N, N_e/N, N_i/N)$, which quantifies the different proportions of control-inducing structures present in a network. The measurements indicate that random network models do not reproduce the control profiles of real-world networks and that the control profiles of real networks group into three well-defined clusters, dominated by external dilations, sources, or internal dilations (Ruths and Ruths, 2014).

These results offer insight into the high-level organization and function of complex networks (Fig. 16). For example, neural and social networks are source dominated, which allow relatively uncorrelated behavior across their agents and are thus suitable to distributed processing. Food webs and airport interconnectivity networks are internal dilation dominated. They are mostly closed systems and obey some type of conservation laws. In contrast, trust hierarchies and transcriptional systems are external dilation dominated. With their surplus sink nodes, these systems display correlated behavior across their agents that are downstream neighbors of a common source.

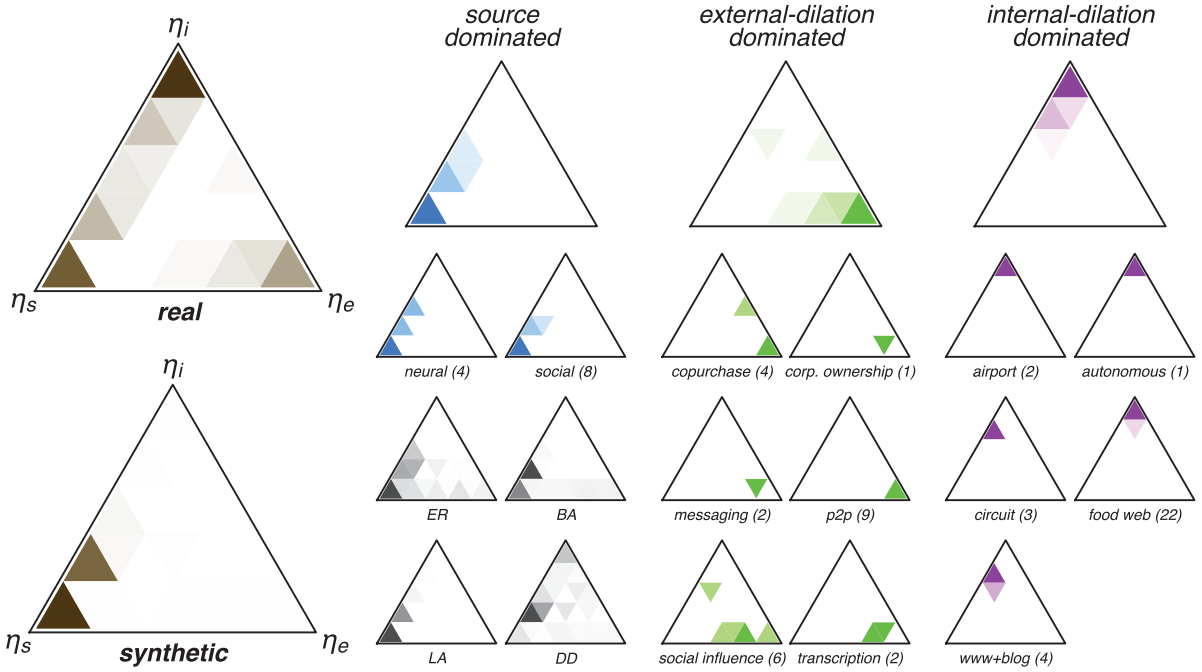

FIG. 16. Control profiles of real and model networks. The control profiles of real networks show a tendency to cluster around the three components (η_s, η_e, η_i) of the control profile, implying that real networks broadly fall into three distinct classes: external-dilation dominated, source dominated, and internal-dilation dominated. The coloring of each small heat map indicates the clustering observed in a wide range of real networks, with numbers in parentheses indicating the number of networks present in each heat map. Deeper shades of the heat map represent a greater density of networks with control profiles located in that region. From Ruths and Ruths, 2014.

G. Controllable subspace, control centrality, and structure permeability

Lin's structural controllability theorem can tell us whether an LTI system $(\mathbf{A}, \mathbf{B})$ is structurally controllable or not. If, however, the system is not structurally controllable, the theorem does not provide further information about controllability. Even if we are unable to make the system reach any point in the state space, we want to understand which region of the state space is accessible to it, i.e., what region of the state space can we control? For example, in the network of Fig. 4(a) the control input u_1 is applied to the central hub x_1 of the directed star with $N = 3$ nodes. The system is therefore stuck in the plane described by $a_{31}x_2(t) = a_{21}x_3(t)$, shaded in Fig. 4(b). Consequently, the network is not controllable in the whole state space, but it is controllable within the subspace defined by the plane.

When we control a single node i, the input matrix $\mathbf{B}$ reduces to a vector $\mathbf{b}(i)$ with a single nonzero entry, and the controllability matrix $\mathbf{C} \in \mathbb{R}^{N\times N}$ becomes $\mathbf{C}(i)$. We can use $\mathrm{rank}(\mathbf{C}(i))$ as a natural measure of node i's ability to control the system. If $\mathrm{rank}(\mathbf{C}(i)) = N$, then node i alone can control the whole system. Any $\mathrm{rank}(\mathbf{C}(i))$ less than N yields the dimension of the subspace i can control. For example, if $\mathrm{rank}(\mathbf{C}(i)) = 1$, then node i can control only itself.

In reality the system parameters (i.e., the entries of $\mathbf{A}$ and $\mathbf{B}$) are often not known precisely, except the zeros that mark the absence of connections, rendering the calculation of $\mathrm{rank}(\mathbf{C}(i))$ difficult. This difficulty can be again avoided using structural control theory. Assuming $\mathbf{A}$ and $\mathbf{B}$ are structured matrices, i.e., their elements are either fixed zeros or independent free parameters, then $\mathrm{rank}(\mathbf{C}(i))$ varies as a function of the free parameters of $\mathbf{A}$ and $\mathbf{B}$. However, it achieves its maximum for almost all sets of values of the free parameters except for some pathological cases with Lebesgue measure zero. This maximal value is called the generic rank (Johnston, Barton, and Brisk, 1984) of the controllability matrix $\mathbf{C}(i)$, denoted as $\mathrm{rank}_g(\mathbf{C}(i))$, which also represents the generic dimension of the controllable subspace.

We define the control capacity of a single node i, or control centrality, as the generic dimension of the controllable subspace (Liu, Slotine, and Barabási, 2012)

$$d_c(\mathbf{A}, \mathbf{b}(i)) = \mathrm{rank}_g(\mathbf{C}(i)). \tag{29}$$

This definition can also be extended to the case when we control via a group of nodes. The definition corresponds directly to our intuition of how powerful a single node is (or a group of nodes are) in controlling the whole network. For example, if the control centrality of a single node is N, then we can control the whole system through it.

The calculation of $d_c(\mathbf{A}, \mathbf{B})$ has a graph-theoretic interpretation (Hosoe, 1980). Consider a structured system $(\mathbf{A}, \mathbf{B})$ in which all state vertices are accessible, and let us denote with $\mathcal{G}$ the set of subgraphs of $G(\mathbf{A}, \mathbf{B})$ which can be spanned by a collection of vertex-disjoint cycles and stems. In this case, the generic dimension of the controllable subspace is

$$d_c(\mathbf{A}, \mathbf{B}) = \max_{G\in\mathcal{G}} |E(G)|, \tag{30}$$

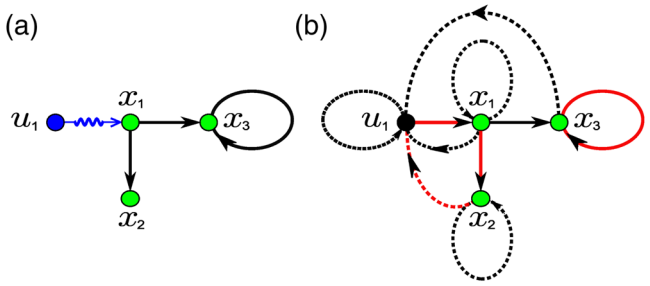

FIG. 17. Calculation of control centrality. (a) The original controlled system is represented by a digraph $G(\mathbf{A},\mathbf{B})$. (b) The modified digraph $G'(\mathbf{A},\mathbf{B})$ used in solving the linear programming. Dotted and solid lines are assigned with weight $w_{ij}=0$ and 1, respectively. The maximum-weight cycle partition is shown in red, which has weight 3, corresponding to the generic dimension of controllable subspace by controlling node x_1 or equivalently the control centrality of node x_1. From Liu, Slotine, and Barabási, 2012.

where $|E(G)|$ is the number of edges in the subgraph G. This is called Hosoe's controllable subspace theorem. Essentially, Hosoe's theorem tells us that to calculate the generic dimension of the controllable subspace we need to find the cactus that contains as many edges as possible. Note that Hosoe's theorem applies only to a structured system $(\mathbf{A},\mathbf{B})$ that has no inaccessible state vertices. In calculating $d_c(\mathbf{A},\mathbf{B})$ for a general system $(\mathbf{A},\mathbf{B})$, we should consider only the accessible part of the network.

For a digraph with no directed cycles Hosoe's theorem further simplifies: the controllability of any node equals its layer index $C_s(i)=l_i$. Here the layer index of a node is calculated from the unique hierarchical structure of the digraph following a recursive labeling procedure (Liu, Slotine, and Barabási, 2012). For general networks, we can use linear programming to calculate $d_c(\mathbf{A},\mathbf{B})$ (Poljak, 1990). We first get a new graph $G'(\mathbf{A},\mathbf{B})$ from $G(\mathbf{A},\mathbf{B})$ by adding to $G(\mathbf{A},\mathbf{B})$ the edges (v_i,v_{N+j}) for $i=1,\ldots,N$, $j=1,\ldots,M$; and the loops (v_i,v_i) for $i=1,\ldots,N+M$, if they do not exist in $G(\mathbf{A},\mathbf{B})$ (see Fig. 17). We associate the weight $w_e=1$ with every original edge e of $G(\mathbf{A},\mathbf{B})$ and the weight $w_e=0$ with every new edge. A collection of node-disjoint cycles in $G'(\mathbf{A},\mathbf{B})$ covering all nodes is called a cycle partition. It is easy to check that to calculate $\max_{G\in\mathcal{G}}|E(G)|$ is equivalent to calculating the maximum-weight cycle partition in $G'(\mathbf{A},\mathbf{B})$, which can then be solved by the following linear programming: $\max\sum_{e\in G'(\mathbf{A},\mathbf{B})}w_e x_e$ subject to (1) $\sum(x_e:e\,\text{leaves node}\,v_i)=1$ for every node $v_i\in G'(\mathbf{A},\mathbf{B})$; (2) $\sum(x_e:e\,\text{enters node}\,v_i)=1$ for every node $v_i\in G'(\mathbf{A},\mathbf{B})$; and (3) $x_e\in\{0,1\}$ for every edge $e\in G'(\mathbf{A},\mathbf{B})$.

Hosoe's theorem also allows us to address a problem complementary to the notion of control centrality: identify an optimal set of driver nodes of fixed cardinality M, denoted as $\Omega_D(M)$, for a network of size N such that the dimension of the controllable subspace, denoted as $|\mathcal{C}(M)|$, is maximized (Lo Iudice, Garofalo, and Sorrentino, 2015). If we solve this problem for each $M\in[1,N]$, we obtain a sequence of $|\mathcal{C}(M)|$. To quantify the readiness or propensity of a network to be controllable, we can calculate the so-called *network permeability* measure (Lo Iudice, Garofalo, and Sorrentino, 2015)

$$\mu=\frac{\int_0^N[|\mathcal{C}(M)|-M]dM}{\int_0^N(N-M)dM}. \tag{31}$$

Note that $\mu\in[0,1]$: 0 for N disconnected nodes, and 1 for networks that are completely controllable by one driver node. Generally, for a network with a high permeability, a large controllable subspace can be obtained with a reasonable small set of driver nodes.

H. Controlling edges

So far we focused on nodal dynamics, where we monitored and controlled the state of nodes. The sole purpose of the edges was to pass information or influence between the nodes. In social or communication networks nodes constantly process the information received from their upstream neighbors and make decisions that are communicated to their downstream neighbors. Most importantly, in these systems nodes can communicate different information along different edges. Hence the information received and passed on by a node can be best represented by state variables defined on the incoming and outgoing edges, respectively. In this section we ask how to control systems characterized by such edge dynamics.

To model such systems we place the state variables on the edges (Nepusz and Vicsek, 2012). Let $\mathbf{y}_i^-(t)$ and $\mathbf{y}_i^+(t)$ represent vectors consisting of the state variables associated with the incoming and outgoing edges of node i, respectively. Let $\mathbf{M}_i$ denote the $k_{\text{out}}(i)\times k_{\text{in}}(i)$ matrix. The equations governing the edge dynamics can be written as

$$\dot{\mathbf{y}}_i^+(t)=\mathbf{M}_i\mathbf{y}_i^-(t)-\boldsymbol{\tau}_i\otimes\mathbf{y}_i^+(t)+\sigma_i\mathbf{u}_i(t), \tag{32}$$

where $\boldsymbol{\tau}_i$ is a vector of damping terms associated with the outgoing edges, $\otimes$ denotes the entrywise product of two vectors of the same size, and $\sigma_i=1$ if node i is a driver node and 0 otherwise. Note that even though the state variables and the control inputs are defined on the edges, we can still designate a node to be a driver node if its outgoing edges are directly controlled by the control inputs. Equation (32) states that the state variables of the outgoing edges of node i are determined by the state variables of the incoming edges, modulated by a decay term. For a driver node, the state variables of its outgoing edges will also be influenced by the control signals $\mathbf{u}_i$. Since each node i acts as a switchboardlike device mapping the signals of the incoming edges to the outgoing edges using a linear operator $\mathbf{M}_i$, Eq. (32) is often called the *switchboard dynamics*.

There is a mathematical duality between edge dynamics on a network G and nodal dynamics on its line graph $\mathcal{L}(G)$, which represents the adjacencies between edges of G. Each node of $\mathcal{L}(G)$ corresponds to an edge in G, and each edge in $\mathcal{L}(G)$ corresponds to a length-two directed path in G. By applying the minimum input theorem directly to this line graph, we obtain the minimum number of edges we must drive to control the original network. However, this procedure does not minimize the number of driver nodes in the original network. This edge control problem can be mapped to a graph-theoretical problem as follows (Nepusz and Vicsek, 2012). Define node i to be (i) *divergent*, if $k_{\text{out}}(i)>k_{\text{in}}(i)$;

(ii) *convergent*, if $k_{\text{out}}(i) < k_{\text{in}}(i)$; or (iii) *balanced*, if $k_{\text{out}}(i) = k_{\text{in}}(i)$. A connected component in a directed network is called a balanced component if it contains at least one edge and all the nodes are balanced. We can prove that the minimum set of driver nodes required to maintain structural controllability of the switchboard dynamics on a directed network G can be determined by selecting the divergent nodes of G and an arbitrary node from each balanced component.

The controllability properties of this edge dynamics significantly differ from simple nodal dynamics. For example, driver nodes prefer hubs with large out-degree and heterogeneous networks are more controllable, i.e., require fewer driver nodes, than homogeneous networks (Nepusz and Vicsek, 2012). Moreover, positive correlations between the in-degree and out-degree of a node enhances the controllability of edge dynamics, without affecting the controllability of nodal dynamics (Pósfai *et al.*, 2013). Conversely, adding self-loops to individual nodes enhances the controllability of nodal dynamics (Liu, Slotine, and Barabási, 2011a; Pósfai *et al.*, 2013), but leaves the controllability of edge dynamics unchanged.

I. Self-dynamics and its impact on controllability

The nodes of networked systems are often characterized by some self-dynamics, e.g., a term of the form $\dot{x}_i = a_{ii}x_i$, which captures the node's behavior in the absence of interactions with other nodes. If we naively apply structural control theory to systems where each node has a self-dynamic term we obtain a surprising result—a single control input can make an arbitrarily large linear system controllable (Liu, Slotine, and Barabási, 2011a; Cowan *et al.*, 2012). This result represents a special case of the minimum inputs theorem: The self-dynamics contributes a self-loop to each node, hence each node can be matched by itself. Consequently, $G(\mathbf{A})$ has a perfect matching, independent of the network topology, and one input signal is sufficient to control the whole system (Liu, Slotine, and Barabási, 2011a).

To understand the true impact of self-dynamics on network controllability, we must revisit the validity of the assumption that the system parameters are independent of each other. As we show next, relaxing this assumption offers a more realistic characterization of real systems, for which not all system parameters are independent.

Assuming a prototypical linear form of self-dynamics, e.g., first-order $\dot{x} = a_0 x$, second-order $\ddot{x} = a_0 x + a_1 \dot{x}$, etc., we can incorporate the linear self-dynamics with the LTI dynamics of the network in a unified matrix form, as illustrated in Fig. 18. An immediate but counterintuitive result states that in the absence of self-dynamics n_{D} is exactly the same as in the case when each node has a self-loop with identical weight w, i.e.,

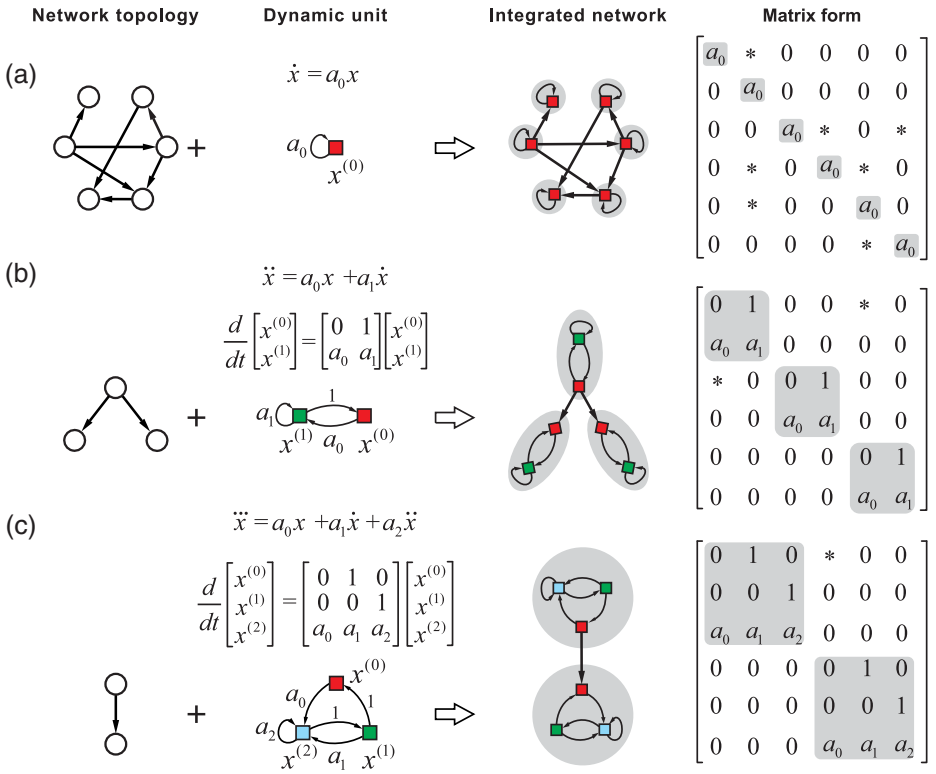

FIG. 18. Integrating the network topology with nodal self-dynamics. (a) First-order self-dynamics $\dot{x} = a_0 x$. (b) Second-order self-dynamics $\ddot{x} = a_0 x + a_1 \dot{x}$. (c) Third-order self-dynamics $\dddot{x} = a_0 x + a_1 \dot{x} + a_2 \ddot{x}$. To develop a graphical representation for the dth-order individual dynamics $x^{(d)} = a_0 x^{(0)} + a_1 x^{(1)} + \cdots + a_{d-1} x^{(d-1)}$, we denote each order by a colored square. The couplings among orders are characterized by links or self-loops. This graphical representation allows the individual dynamics to be integrated with the network topology, giving rise to a unified matrix that reflects the dynamics of the whole system. In particular, each dynamic unit in the unified matrix corresponds to a diagonal block and the nonzero elements (denoted by $*$) outside these blocks stand for the couplings among different dynamic units. Therefore, the original network of N nodes with order d self-dynamics is represented by a $dN \times dN$ matrix. From Zhao *et al.*, 2015.

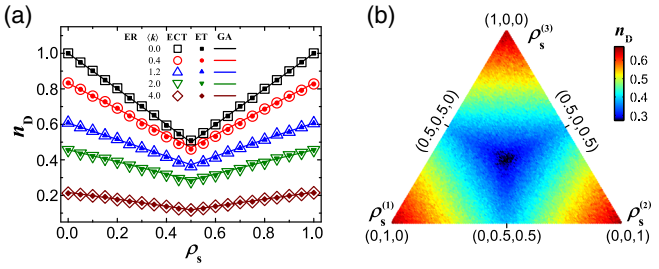

FIG. 19. Impact of first-order self-dynamics on the fraction of driver nodes n_D. The values of the off-diagonal nonzero elements in $\mathbf{A}$ are randomly chosen and hence are independent. (a) n_D in function of ρ_s, the density of nodes that have the same type of nonzero self-loops. We observe a clear symmetry around $\rho_s = 1/2$, indicating that n_D reaches its minimum at $\rho_s = 1/2$, where the densities of nodes with zero and nonzero self-loops are equal. (b) n_D for an Erdős-Rényi random network with three types of self-loops s_1, s_2, and s_3 with densities $\rho_s^{(1)}$, $\rho_s^{(2)}$, and $\rho_s^{(3)}$, respectively. The color bar denotes the value of n_D and the coordinates in the triangle stand for $\rho_s^{(1)}$, $\rho_s^{(2)}$, and $\rho_s^{(3)}$. There is a global symmetry point where the three types of self-loops have the same density $1/3$, and n_D reaches its minimum value. From Zhao *et al.*, 2015.

each node is governed by precisely the same self-dynamics. This is a direct consequence of the identity

$$\begin{aligned}&\mathrm{rank}[\mathbf{B}, \mathbf{AB}, \ldots, \mathbf{A}^{N-1}\mathbf{B}]\\ &\quad = \mathrm{rank}[\mathbf{B}, (\mathbf{A}+w\mathbf{I})\mathbf{B}, \ldots, (\mathbf{A}+w\mathbf{I})^{N-1}\mathbf{B}],\end{aligned} \tag{33}$$

where on the left we have the rank of the controllability matrix in the absence of self-loops, and on the right the same for a network where each node has an identical self-loop. For more general cases the minimum number of driver nodes N_D can be calculated from Eq. (27), i.e., the maximum geometric multiplicity of $\mathbf{A}$'s eigenvalues.

Note a remarkable symmetry in network controllability: If we exchange the fractions of any two types of self-loops with distinct weights, the system's controllability, as measured by n_D, remains the same (Fig. 19). For example, consider a network without self-loops. Equivalently, we can assume that each node contains a self-loop with weight zero. Then we systematically add more nonzero self-loops with identical weights to the network. Equivalently, we are replacing the zero-weight self-loops with nonzero self-loops. n_D will first decrease as the fraction ρ of nonzero self-loops increases, reaching a minimum at $\rho = 1/2$. After that, n_D increases, reaching its maximum at $\rho = 1$, which coincides with n_D observed for $\rho = 0$ [Fig. 19(a)]. We can introduce more types of self-loops with different weights. If we exchange the fractions of any two types of self-loops, n_D remains the same. This exchange-invariant property gives rise to a global symmetry point, where all the different types of self-loops have equal densities and the system displays the highest controllability (i.e., lowest number of driver nodes). This symmetry-induced optimal controllability holds for any network topology and various individual dynamics (Zhao *et al.*, 2015).

J. Control energy

Indentifying the minimum number of driver or actuator nodes sufficient for control is only the first step of the control problem. Once we have that, we need to ask an equally important question: How much effort is required to control a system from a given set of nodes? The meaning of the term "control effort" depends upon the particular application (Kirk, 2004). In the case of a rocket being thrust upward, the control input $u(t)$ is the thrust of the engine, whose magnitude $|u(t)|$ is assumed to be proportional to the rate of fuel consumption. In order to minimize the total expenditure of fuel, the control effort can be defined as $\int_0^T |u(t)|dt$, which is related to the energy consumed by the rocket. In the case of a voltage source driving a circuit containing no energy storage elements, the source voltage is the control input $u(t)$ and the source current is directly proportional to $u(t)$. If the circuit is to be controlled with minimum energy dissipation, we can define the control effort as $\int_0^T u^2(t)dt$, which is proportional to the energy dissipation. If there are several control inputs, the general form of control effort can be defined as $\int_0^T \mathbf{u}^{\mathrm{T}}(t)\mathbf{R}(t)\mathbf{u}(t)dt$, where $\mathbf{R}(t)$ is a real symmetric positive-definite weighting matrix.

Consider the LTI system (11) driven from an arbitrary initial state $\mathbf{x}_i$ toward a desired final state $\mathbf{x}_f$ by the external signal $\mathbf{u}(t)$ in the time interval $t \in [0, T]$. We define the associated control effort in the quadratic form

$$\mathcal{E}(T) \equiv \int_0^T \|\mathbf{u}(t)\|^2 dt, \tag{34}$$

called the "control energy" in the literature (Yan *et al.*, 2012, 2015; Chen *et al.*, 2016). Note that Eq. (34) may not have the physical dimension of energy, i.e., $M L^2 T^{-2}$, in real control problems. But for physical and electronic systems we can always assume there is a hidden constant in the right-hand side of Eq. (34) with proper dimension, which ensures that $\mathcal{E}(T)$ has the dimension of energy. In many systems, like biological or social systems, where Eq. (34) does not correspond to energy, it captures the effort needed to control a system.

For a fixed set of driver nodes the control input $\mathbf{u}(t)$ that can drive the system from $\mathbf{x}_i$ to $\mathbf{x}_f$ can be chosen in many different ways, resulting in different trajectories followed by the system. Each of these trajectories has its own control energy. Of all the possible inputs, the one that yields the minimum control energy is

$$\mathbf{u}(t) = \mathbf{B}^{\mathrm{T}} \exp[\mathbf{A}^{\mathrm{T}}(T-t)]\mathbf{W}^{-1}(T)\mathbf{v}_f, \tag{35}$$

where $\mathbf{W}(t)$ is the Gramian matrix

$$\mathbf{W}(t) \equiv \int_0^t \exp(\mathbf{A}\tau)\mathbf{B}\mathbf{B}^{\mathrm{T}} \exp(\mathbf{A}^{\mathrm{T}}\tau)d\tau, \tag{36}$$

which is nonsingular for any $t > 0$ (Lewis, Vrabie, and Syrmos, 2012). Note that $\mathbf{W}(\infty)$ is known as the controllability Gramian, often denoted with $\mathbf{W}_c$ (Kailath, 1980). The energy associated with the optimal input (35) is $\mathcal{E}(T) = \mathbf{v}_f^{\mathrm{T}}\mathbf{W}^{-1}(T)\mathbf{v}_f$, where $\mathbf{v}_f \equiv \mathbf{x}_f - \exp(\mathbf{A}T)\mathbf{x}_i$ represents the difference between the desired state under control and the final

state during free evolution without control. Without loss of generality, we can set the final state at the origin $\mathbf{x}_f = \mathbf{0}$ and write the control energy as

$$\mathcal{E}(T) = \mathbf{x}_i^T \mathbf{H}^{-1}(T)\mathbf{x}_i, \tag{37}$$

where $\mathbf{H}(T) = \exp(-\mathbf{A}T)\mathbf{W}(T)\exp(-\mathbf{A}^T T)$ is the symmetric Gramian matrix. We can further define the normalized control energy as

$$E(T) \equiv \frac{\mathcal{E}(T)}{\|\mathbf{x}_i\|^2} = \frac{\mathbf{x}_i^T \mathbf{H}^{-1}\mathbf{x}_i}{\mathbf{x}_i^T \mathbf{x}_i}. \tag{38}$$

When $\mathbf{x}_i$ is parallel to the direction of one of $\mathbf{H}$'s eigenvectors, the inverse of the corresponding eigenvalue represents the normalized energy associated with controlling the system along the particular eigendirection.

Using the Rayleigh-Ritz theorem, the normalized control energy obeys the bounds

$$\eta_{\max}^{-1} \equiv E_{\min} \le E(T) \le E_{\max} \equiv \eta_{\min}^{-1}, \tag{39}$$

where $\eta_{\max}$ and $\eta_{\min}$ are the maximum and minimum eigenvalues of $\mathbf{H}$, respectively (Yan *et al.*, 2012).

Assuming linear individual dynamics characterized by the self-loop $a_{ii} = -(a + s_i)$, where $s_i = \sum_{j \ne i} a_{ij}$ is the strength of node i and a is a parameter that can make the symmetric $\mathbf{A}$ (describing an undirected network) either positive or negative definite, we can choose a single node with index c as the driver node. In this case, the lower and upper energy bounds follow

$$E_{\min} \sim \begin{cases} T^{-1} & \text{small } T, \\ 1/[(\mathbf{A}+\mathbf{A}^T)^{-1}]_{cc} & \text{large } T, \mathbf{A} \text{ is PD}, \\ T^{-1} \to 0 & \text{large } T, \mathbf{A} \text{ is semi PD}, \\ \exp(2\lambda_N T) \to 0 & \text{large } T, \mathbf{A} \text{ is not PD}, \end{cases} \tag{40}$$

$$E_{\max} \sim \begin{cases} T^{-\theta}(\theta \gg 1) & \text{small } T, \\ \varepsilon(\mathbf{A}, c) & \text{large } T, \mathbf{A} \text{ is not ND}, \\ T^{-1} \to 0 & \text{large } T, \mathbf{A} \text{ is semi ND}, \\ \exp(2\lambda_1 T) \to 0 & \text{large } T, \mathbf{A} \text{ is ND}. \end{cases} \tag{41}$$

Here $\lambda_1 > \lambda_2 > \cdots > \lambda_N$ are the eigenvalues of $\mathbf{A}$, and $\varepsilon(\mathbf{A}, c)$ is a positive energy that depends on the matrix $\mathbf{A}$ and the choice of the controlled node c. PD or ND means positive definite or negative definite, respectively. The scaling laws (40) and (41) can be generalized to directed networks, in which case the decay exponents λ_1 and λ_N are replaced by $\mathrm{Re}\lambda_1$ and $\mathrm{Re}\lambda_N$, respectively.

Equations (40) and (41) suggest that the scaling of the control energy is rather sensitive to the control time T. For small T, in which case we wish to steer our system very fast to its destination, both $E_{\min}$ and $E_{\max}$ decay with increasing T, implying that setting a somewhat longer control time requires less energy. For large T, however, we reach a point where we cannot reduce the energy by waiting for a longer time. This occurs when the system has its equilibrium point in the origin, then any attempt to steer the system away from the origin must overcome a certain energy barrier.

The control energy is rather sensitive to the direction of the state space in which we want to move the system (see Fig. 20) (Yan *et al.*, 2015). To see this, consider a scale-free network with degree exponent γ. If we drive the system through all its nodes ($N_D = N$), the control energy spectrum, describing the probability that moving in a randomly chosen eigendirection will require energy $\mathcal{E}$, follows the power law $P(\mathcal{E}) \sim \mathcal{E}^{-\gamma}$. Consequently, the maximum energy required for control depends sublinearly on the system size $\mathcal{E}_{\max} \sim N^{1/(\gamma-1)}$, implying that even in the most costly direction the required energy grows slower than the system size. In other words, if we control each node, there are no significant energetic barriers for control. If, however, we aim to control the system through a single node ($N_D = 1$), the control spectrum follows a power law with exponent -1, i.e., $P(\mathcal{E}) \sim \mathcal{E}^{-1}$, which only weakly depends on the network structure. Therefore the maximum energy required for control increases as $\mathcal{E}_{\max} \sim e^N$. This exponential increase means that steering the network in some directions is energetically prohibitive. Finally, if we drive a finite fraction of nodes ($1 < N_D < N$), the control spectrum has multiple peaks and the maximum energy required for control scales as $\mathcal{E}_{\max} \sim e^{N/N_D}$. Hence, as we increase the number of driver nodes, the maximum energy decays exponentially.

These results raise an important question: in the case of $1 < N_D < N$, how to choose the optimal set of N_D driver nodes such that the control energy is minimized? Such a combinatorial optimization problem (also known as the actuator placement problem) has not been extensively studied in the literature. Only recently has it been shown that several objective functions, i.e., energy-related controllability metrics associated with the controllability Gramian $\mathbf{W}_c$ of LTI systems [e.g., $\mathrm{Tr}(\mathbf{W}_c^{-1})$, $\log(\det \mathbf{W}_c)$, and $\mathrm{rank}(\mathbf{W}_c)$], are actually *submodular* (Summers and Lygeros, 2014; Cortesi, Summers, and Lygeros, 2014; Summers, Cortesi, and Lygeros, 2016). A submodular function[3] f has the so-called diminishing returns property that the difference in the function value that a single element x makes when added to an input set $\mathcal{X}$ decreases as the size of the input set increases. The submodularity of objective functions allows for either an efficient global optimization or a simple greedy approximation algorithm with certain performance guarantee to solve the combinatorial optimization problems (Nemhauser, Wolsey, and Fisher, 1978). In particular, the submodularity of those energy-related controllability metrics has been explored to address the actuator placement problem in a model of the European power grid (Summers and Lygeros, 2014; Cortesi, Summers, and Lygeros, 2014; Summers, Cortesi, and Lygeros, 2016).

[3]Denote $\mathcal{P}(\mathcal{S})$ as the power set (i.e., the set of all the subsets) of a set $\mathcal{S}$. Then a submodular function is a set function $f: \mathcal{P}(\mathcal{S}) \to \mathbb{R}$ that satisfies $f(\mathcal{X} \cup \{x\}) - f(\mathcal{X}) \ge f(\mathcal{Y} \cup \{x\}) - f(\mathcal{Y})$, for any $\mathcal{X} \subseteq \mathcal{Y} \subseteq \mathcal{S}$ and $x \in \mathcal{S} \backslash \mathcal{Y}$.

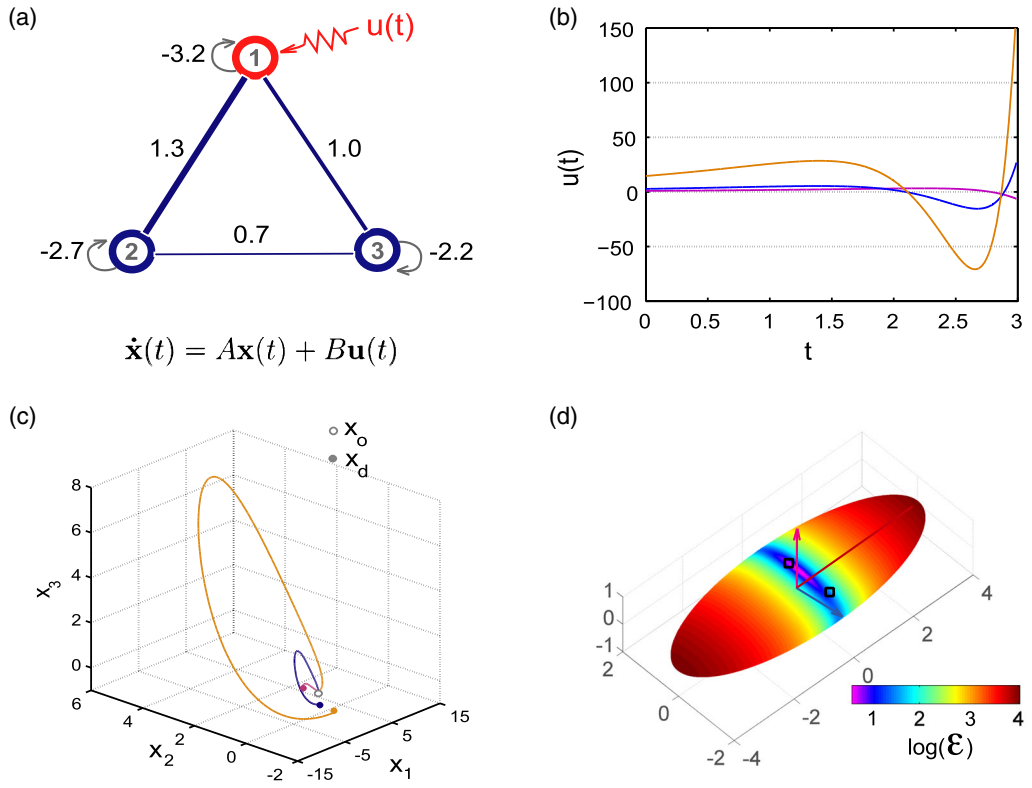

FIG. 20. Energy spectrum. (a) A three-node weighted network can be controlled via a single control input $u(t)$, injected to the driver node shown in red. The input matrix $\mathbf{B}$ is reduced to a vector $(1,0,0)^{\mathrm{T}}$. Each node has a negative self-loop, which makes all eigenvalues of the state matrix $\mathbf{A}$ negative, hence stable. (b) The optimal control signals that minimize the energies required to steer the network from the initial state $\mathbf{x}_0 = \mathbf{x}(0) = (0,0,0)^{\mathrm{T}}$ to three different desired states $\mathbf{x}_{\mathrm{d}} = \mathbf{x}(t)$ at $t = 3$, with the constraint $\|\mathbf{x}_{\mathrm{d}}\| = 1$. (c) The trajectories of the network state $\mathbf{x}(t)$ driven by the control inputs shown in (b). (d) The energy surface for all normalized desired states, i.e., $\|\mathbf{x}_{\mathrm{d}}\| = 1$, which is an ellipsoid spanned by the controllability Gramian's three eigendirections (arrows). The ellipsoid nature of the spectrum illustrates the widely different energies we need to move the network shown in (a) in different directions in the state space. The squares correspond to the three cases depicted in (b) and (c). From Yan *et al.*, 2015.

K. Control trajectories

So far we have focused on the minimization of driver and actuator nodes and the energy cost of controlling LTI systems. The characteristics of the resulting control trajectories are also interesting and worthy of exploration (Sun and Motter, 2013). A state $\mathbf{x}^{(0)}$ of the LTI system is called strictly locally controllable (SLC) if for a ball $B(\mathbf{x}^{(0)}, \varepsilon)$ centered at $\mathbf{x}^{(0)}$ with radius $\varepsilon > 0$ there is a constant $\delta > 0$ such that any final state $\mathbf{x}^{(1)}$ inside the ball $B(\mathbf{x}^{(0)}, \delta)$ can be reached from $\mathbf{x}^{(0)}$ with a control trajectory entirely inside the ball $B(\mathbf{x}^{(0)}, \varepsilon)$ [see Fig. 21(a)]. Figure 21(b) shows that in a two-dimensional LTI system $\dot{x}_1 = x_1 + u_1(t)$, $\dot{x}_2 = x_1$, for any state in the $x_1 > 0$ half plane, the minimal-energy control trajectories to any neighboring final state with a smaller x_2 component will necessarily cross into the $x_1 < 0$ half plane.

It has been shown that for a general LTI system whenever the number of control inputs is smaller than the number of state variables (i.e., $N_{\mathrm{D}} < N$), then almost all the states are not SLC (Sun and Motter, 2013). Therefore, the minimal-energy control trajectory is generally nonlocal and remains finite even when the final state is brought arbitrarily close to the initial state. The length $\int_0^{t_f} \|\dot{\mathbf{x}}(t)\| dt$ of such a trajectory generally increases with the condition number of the Gramian. Furthermore, the optimal control input (35) that minimizes the energy cost $\int_0^{t_f} \|\mathbf{u}(t)\|^2 dt$ will fail in practice if the controllability Gramian (36) is ill conditioned. This can occur even when the controllability matrix is well conditioned. There is a sharp transition, called the controllability transition, as a function of the number of control inputs, below which numerical control always fails and above which it succeeds. These results indicate that even for the simplest LTI dynamics, the disparity between theory and practice poses a fundamental limit on our ability to control large networks (Sun and Motter, 2013).

Indeed, we usually do not use the minimum energy control input (35) to steer the system to desired final states, simply because it is an open-loop (or nonfeedback) controller,[4] which tends to be very sensitive to noise. A more practical and robust strategy is to use a simple linear feedback control to bring the system asymptotically toward a certain state, while

[4]An open-loop control system does not use feedback. The control input to the system is determined using only the current state of the system and a model of the system and is totally independent of the system's output. In contrast, in a closed-loop control system, the output has an effect on the input (through feedback) so that the input will adjust itself based on the output.

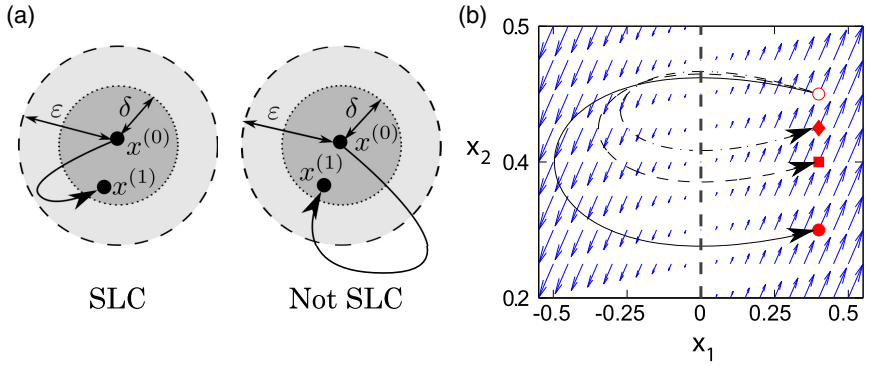

FIG. 21. Strictly local controllability. (a) Illustration of a state that is strictly locally controllable (left) and a state that is not (right). (b) The state space of a simple LTI system with two state variables x_1 and x_2. The curves indicate control trajectories of minimal energy for a given initial state (open symbol) and final states (solid symbols). The arrows in the background represent the vector field in the absence of control input $u_1(t)$. Note that any state that is not on the line $x_1 = 0$ is not SLC, because the minimal-energy control trajectories to any neighboring final state with a smaller x_2 component will necessarily cross into the $x_1 < 0$ half plane. From Sun and Motter, 2013.

minimizing the energy cost. This is a typical objective of optimal control theory, which aims to design control signals that will cause a process to satisfy some physical constraints and maximize (or minimize) a chosen performance criterion (or cost function) (Naidu, 2002; Kirk, 2004).

III. CONTROLLABILITY OF NONLINEAR SYSTEMS

So far we focused on the controllability of linear systems. Yet the dynamics of most real complex systems is nonlinear, prompting us to review the classical results on nonlinear controllability and their applications to networked systems.

Consider a control system of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \tag{42}$$

where the state vector $\mathbf{x}$ is in a smooth connected manifold $\mathcal{M}$ of dimension N, and the control input $\mathbf{u} \in \mathcal{U}$ is a subset of $\mathbb{R}^M$. Note that Eq. (42) has been frequently used to model the behavior of physical, biological, and social systems (Hermann and Krener, 1977). Roughly speaking, Eq. (42) is controllable if one can steer it from any point $\mathbf{x}_0 \in \mathcal{M}$ to any other point $\mathbf{x}_1 \in \mathcal{M}$ by choosing $\mathbf{u}$ from a set of admissible controls $\mathbb{U}$, which is a subset of functions mapping $\mathbb{R}^+$ to $\mathcal{U}$.

The controllability of nonlinear systems has been extensively studied since the early 1970s (Elliot, 1971; Haynes and Hermes, 1970; Lobry, 1970; Brockett, 1972; Sussmann and Jurdjevic, 1972; Hermann and Krener, 1977; Rugh, 1981; Nijmeijer and van der Schaft, 1990; de Figueiredo and Chen, 1993; Isidori, 1995; Sontag, 1998; Conte, Moog, and Perdon, 2007). The goal was to derive results of similar reach and generality as obtained for LTI systems. However, this goal turned out to be too ambitious, suggesting that a general theory on nonlinear controllability may not be feasible. Fortunately, as discussed in this section, the concerted effort on nonlinear control has led to various weaker notions of nonlinear controllability, which are easier to characterize and often offer simple algebraic tests to explore the controllability of nonlinear systems.

A. Accessibility and controllability

As we will see in the coming sections, we can rarely prove or test controllability of an arbitrary nonlinear system. Instead, we prove and test weaker versions of controllability called local accessibility and local strong accessibility. We start by defining these notions.

Accessibility concerns the possibility to reach or access an open set of states in the state space from a given initial state. If the system (42) is locally accessible from an initial state $\mathbf{x}_0$, then we can reach or access the neighborhood of $\mathbf{x}_0$ through trajectories that are within the neighborhood of $\mathbf{x}_0$. Mathematically, the system (42) is called locally accessible from $\mathbf{x}_0$ if for any nonempty neighborhoods $\mathcal{V} \subset \mathcal{M}$ of $\mathbf{x}_0$ and any $t_1 > 0$, the reachable set $\mathcal{R}^{\mathcal{V}}(\mathbf{x}_0, \leq t_1)$ contains a nonempty open set. The system is called locally accessible if this holds for any $\mathbf{x}_0$. Here the reachable set $\mathcal{R}^{\mathcal{V}}(\mathbf{x}_0, \leq t_1)$ includes all states that can be reached from $\mathbf{x}_0$ within a time t_1, following trajectories that are within the neighborhood of $\mathbf{x}_0$. Mathematically, the reachable set from $\mathbf{x}_0$ in time up to t_1 is defined as $\mathcal{R}^{\mathcal{V}}(\mathbf{x}_0, \leq t_1) \equiv \cup_{\tau \leq t_1} \mathcal{R}^{\mathcal{V}}(\mathbf{x}_0, \tau)$. Here $\mathcal{R}^{\mathcal{V}}(\mathbf{x}_0, \tau)$ is the reachable set from $\mathbf{x}_0$ at time $\tau > 0$ following trajectories that remain in $\mathcal{V}$ for $t \leq \tau$.

If we look at states that can be reached exactly at time t_1, then we have a stronger version of local accessibility. The system (42) is said to be locally strongly accessible from $\mathbf{x}_0$ if at any small time $t_1 > 0$ the system can reach or access the neighborhood of $\mathbf{x}_0$ through trajectories that are within the neighborhood of $\mathbf{x}_0$. Mathematically, this means that for any nonempty neighborhoods $\mathcal{V}$ of $\mathbf{x}_0$ and any $t_1 > 0$ sufficiently small, the reachable set $\mathcal{R}^{\mathcal{V}}(\mathbf{x}_0, t_1)$ contains a nonempty open set. If this holds for any $\mathbf{x}_0$, then the system is called locally strongly accessible. Clearly, local strong accessibility from $\mathbf{x}_0$ implies local accessibility from $\mathbf{x}_0$. The converse is generally not true.

Local controllability asks whether the system is controllable in some neighborhood of a given state. Mathematically, the system (42) is called locally controllable from $\mathbf{x}_0$ if for any neighborhood $\mathcal{V}$ of $\mathbf{x}_0$, the reachable set $\mathcal{R}^{\mathcal{V}}(\mathbf{x}_0, \leq t_1)$ is also a neighborhood of $\mathbf{x}_0$ for any t_1 small enough. The system is called locally controllable if this holds for any $\mathbf{x}_0$. Clearly,

local controllability implies local accessibility. It turns out that for a large class of systems local controllability implies local strong accessibility. But the converse is not always true.

If we do not require the trajectories of the system to remain close to the starting point, i.e., we allow excursions, then we have the notion of global controllability. The system (42) is globally controllable from $\mathbf{x}_0$ if the reachable set from $\mathbf{x}_0$ is $\mathcal{M}$ itself, i.e., $\mathcal{R}(\mathbf{x}_0) \equiv \cup_{t_1 \geq 0} \mathcal{R}^{\mathcal{M}}(\mathbf{x}_0, t_1) = \mathcal{M}$. In other words, for any $\mathbf{x}_1 \in \mathcal{M}$, there exists $t_1 > 0$ and $\mathbf{u}: [0, t_1] \to \mathcal{U}$ such that the solution of Eq. (42) starting at $\mathbf{x}_0$ at time 0 with control $\mathbf{u}(t)$ satisfies $\mathbf{x}(t_1) = \mathbf{x}_1$. If this holds for all $\mathbf{x}_0 \in \mathcal{M}$, then the system is called globally controllable.

Complete algebraic characterizations of global controllability of nonlinear systems have proved elusive. Weaker notions of controllability are easier to characterize than controllability. For example, it can be proven that for some nonlinear systems, accessibility can be decided in polynomial time, while controllability is NP hard (Sontag, 1988). For complex networked systems we expect that only weaker notions of controllability can be characterized.

B. Controllability of linearized control system

It is typically difficult to test the controllability of a nonlinear system. Yet, as we discuss next, studying the controllability properties of its linearization around an equilibrium point or along a trajectory can often offer an efficient test of local nonlinear controllability (Coron, 2009).

1. Linearization around an equilibrium point

Consider an equilibrium point $(\mathbf{x}^*, \mathbf{u}^*) \in \mathcal{M} \times \mathcal{U}$ of the nonlinear control system (42), meaning that $\mathbf{f}(\mathbf{x}^*, \mathbf{u}^*) = \mathbf{0}$. Assume that $\mathcal{U}$ contains a neighborhood of $\mathbf{u}^*$. For $\epsilon > 0$, we define a set of control functions $\mathbb{U}_\epsilon \equiv \{\mathbf{u}(\cdot) \in \mathbb{U} | \|\mathbf{u}(t) - \mathbf{u}^*\| < \epsilon, t \geq 0\}$. The linearized control system at $(\mathbf{x}^*, \mathbf{u}^*)$ is a linear control system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ with

$$\mathbf{A} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}^*, \mathbf{u}^*), \qquad \mathbf{B} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\mathbf{x}^*, \mathbf{u}^*). \tag{43}$$

If the linearized control system is controllable (in the sense of an LTI system), then for any $\epsilon > 0$ the original nonlinear system is locally controllable from $\mathbf{x}^*$, where the control functions $\mathbf{u}(\cdot)$ are taken from the set $\mathbb{U}_\epsilon$.

In other words, many real systems operate near some equilibrium points and in the vicinity of such points, controllability can be decided using the tools developed for linear systems, discussed in Sec. II.

2. Linearization around a trajectory

We can also study the linearized control system along a trajectory. Consider a nonlinear control system in the form of Eq. (42). A trajectory represents the path the system follows as a function of time in the state space. It can be mathematically defined as a function $(\bar{\mathbf{x}}, \bar{\mathbf{u}}): [T_0, T_1] \to \mathcal{O}$, where $\mathcal{O}$ is a nonempty open subset of $\mathbb{R}^N \times \mathbb{R}^M$ and $\bar{\mathbf{x}}(t_2) = \bar{\mathbf{x}}(t_1) + \int_{t_1}^{t_2} \mathbf{f}(\bar{\mathbf{x}}(t), \bar{\mathbf{u}}(t)) dt$, for all $(t_1, t_2) \in [T_0, T_1]$. The linearized control system of Eq. (42) along a trajectory $(\bar{\mathbf{x}}, \bar{\mathbf{u}}): [T_0, T_1] \to \mathcal{O}$ is a linear time-varying control system $\dot{\mathbf{x}} = \mathbf{A}(t)\mathbf{x} + \mathbf{B}(t)\mathbf{u}$ with $t \in [T_0, T_1]$, and

$$\mathbf{A}(t) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\bar{\mathbf{x}}(t), \bar{\mathbf{u}}(t)), \qquad \mathbf{B}(t) = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\bar{\mathbf{x}}(t), \bar{\mathbf{u}}(t)). \tag{44}$$

If the linearized control system along the trajectory $(\bar{\mathbf{x}}, \bar{\mathbf{u}}): [T_0, T_1] \to \mathcal{O}$ is controllable in the sense of a linear time-varying system, then the original nonlinear system is locally controllable along the trajectory. Once again, this means that we can use linear control theory to explore the controllability of nonlinear systems.

3. Limitations of linearization

The linearization approaches described may sound powerful, but they have severe limitations. First, they only provide information about controllability in the immediate vicinity of an equilibrium point or a trajectory. Second and most important, it may be the case that the linearized control system is not controllable, but the original nonlinear system is actually controllable.

Consider, for example, a model of a front-wheel drive car with four state variables: the positions (x_1, x_2) of the center of the front axle, the orientation ϕ of the car, and the angle θ of the front wheels relative to the car orientation (Fig. 22). There are two control inputs (u_1, u_2), where u_1, the steering velocity, represents the velocity with which the steering wheel is turning, and u_2 is the driving velocity. Assuming that the front and rear wheels do not slip and that the distance between them is $l = 1$, the car's equations of motion have the form (Nelson, 1967; Sontag, 1998)

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\phi} \\ \dot{\theta} \end{pmatrix} = u_1 \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} + u_2 \begin{pmatrix} \cos(\theta + \phi) \\ \sin(\theta + \phi) \\ \sin\theta \\ 0 \end{pmatrix}. \tag{45}$$

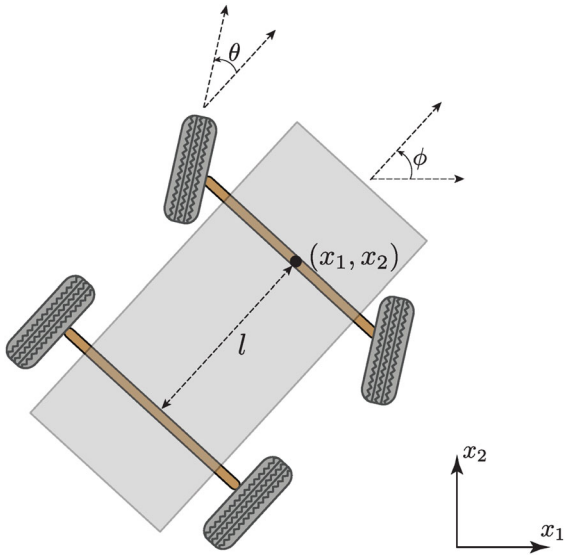

FIG. 22. Controlling a car. A model of a front-wheel drive car with four state variables (x_1, x_2, ϕ, θ) and two control inputs (u_1, u_2). While this system is globally controllable (see Sec. III.E), its linearized dynamics around the origin is not controllable. Adapted from Sontag, 1998.

The linearization of Eq. (45) around the origin is

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\phi} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} u_2 \\ 0 \\ 0 \\ u_1 \end{pmatrix}, \tag{46}$$

which is uncontrollable, because x_2 and ϕ are time invariant and not controlled by any of the system's inputs. Yet, from our driving experience we know that a car is controllable. We will prove that this system is indeed globally controllable in Sec. III.E.

The system (45) belongs to an especially interesting class of nonlinear systems, called control-affine systems, where $\mathbf{f}(\mathbf{x},\mathbf{u})$ is linear in the control signal $\mathbf{u}$

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \sum_{i=1}^{M} \mathbf{g}_i(\mathbf{x}) u_i. \tag{47}$$

Here $\mathbf{f}$ is called the drift vector field, or simply *drift*; and $\mathbf{g}_1,\ldots,\mathbf{g}_M$ are called the control vector fields. The system (47) is called *driftless* if $\mathbf{f}(\mathbf{x}) \equiv \mathbf{0}$, which arises in kinematic models of many mechanical systems, e.g., in Eq. (45). Control-affine systems are natural generalization of linear time-invariant systems. Many nonlinear controllability results were obtained for them. Hereafter we focus on control-affine systems; see Hermann and Krener (1977) and Sontag (1998) for more general nonlinear systems.

C. Basic concepts in differential geometry

Before we discuss the nonlinear tests for accessibility and controllability, we need a few concepts in differential geometry, such as Lie brackets and distributions.

1. Lie brackets

For nonlinear control systems, both controllability and accessibility are intimately tied to Lie brackets. The reason is simple. In the nonlinear framework, the directions in which the state may be moved around an initial state $\mathbf{x}_0$ are those belonging to the Lie algebra generated by vector fields $\mathbf{f}(\mathbf{x}_0,\mathbf{u})$, when $\mathbf{u}$ varies in the set of admissible controls $\mathbb{U}$ (Isidori, 1995; Sontag, 1998). Here the Lie algebra $\mathcal{A}$ generated by a family $\mathcal{F}$ of vector fields is the set of Lie brackets $[\mathbf{f},\mathbf{g}]$ with $\mathbf{f},\mathbf{g} \in \mathcal{F}$, and all vector fields that can be obtained by iteratively computing Lie brackets. In turn, a Lie bracket is the derivative of a vector field with respect to another.

Consider two vector fields $\mathbf{f}$ and $\mathbf{g}$ on an open set $D \subset \mathbb{R}^N$. The Lie bracket operation generates a new vector field $[\mathbf{f},\mathbf{g}]$, defined as

$$[\mathbf{f},\mathbf{g}](\mathbf{x}) \equiv \frac{\partial \mathbf{g}}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}) - \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{g}(\mathbf{x}), \tag{48}$$

where $\partial\mathbf{g}/\partial\mathbf{x}$ and $\partial\mathbf{f}/\partial\mathbf{x}$ are the Jacobian matrices of $\mathbf{g}$ and $\mathbf{f}$, respectively. Higher-order Lie brackets can be recursively defined as

$$\mathrm{ad}_{\mathbf{f}}^0\mathbf{g}(\mathbf{x}) \equiv \mathbf{g}(\mathbf{x}), \tag{49}$$

$$\mathrm{ad}_{\mathbf{f}}^k\mathbf{g}(\mathbf{x}) \equiv [\mathbf{f}, \mathrm{ad}_{\mathbf{f}}^{k-1}\mathbf{g}](\mathbf{x}), \quad \forall\ k \geq 1, \tag{50}$$

where "ad" denotes "adjoint."

To understand the physical meaning of the Lie bracket, consider the following piecewise constant control inputs

$$\mathbf{u}(t) = \begin{cases} (1,0)^{\mathrm{T}}, & t \in [0,\tau), \\ (0,1)^{\mathrm{T}}, & t \in [\tau,2\tau), \\ (-1,0)^{\mathrm{T}}, & t \in [2\tau,3\tau), \\ (0,-1)^{\mathrm{T}}, & t \in [3\tau,4\tau), \end{cases} \tag{51}$$

applied onto a two-inputs control-affine system

$$\dot{\mathbf{x}} = \mathbf{g}_1(\mathbf{x})u_1 + \mathbf{g}_2(\mathbf{x})u_2 \tag{52}$$

with initial state $\mathbf{x}(0) = \mathbf{x}_0$ (Sastry, 1999). The piecewise constant control inputs (51) can be considered as a sequence of "actions" applied, for example, to a car ($\mathbf{g}_1$, $\mathbf{g}_2$, reverse $\mathbf{g}_1$, and reverse $\mathbf{g}_2$). In the limit $\tau \to 0$ the final state reached at $t = 4\tau$ is

$$\mathbf{x}(4\tau) = \mathbf{x}_0 + \tau^2\left(\frac{\partial \mathbf{g}_2}{\partial \mathbf{x}}\mathbf{g}_1(\mathbf{x}_0) - \frac{\partial \mathbf{g}_1}{\partial \mathbf{x}}\mathbf{g}_2(\mathbf{x}_0)\right) + O(\tau^3). \tag{53}$$

We see that up to terms of order τ^2, the state change is exactly along the direction of the Lie bracket $[\mathbf{g}_1,\mathbf{g}_2](\mathbf{x}_0)$ (see Fig. 23).

Consider two examples that demonstrate the meaning of Lie brackets. First, the Brockett system is one of the simplest driftless control-affine systems (Brockett, 1982)

$$\dot{x}_1(t) = u_1, \quad \dot{x}_2(t) = u_2, \quad \dot{x}_3(t) = u_2x_1 - u_1x_2, \tag{54}$$

which can be written in the form of Eq. (52) using $\mathbf{g}_1(\mathbf{x}) = (1,0,-x_2)^{\mathrm{T}}$ and $\mathbf{g}_2(\mathbf{x}) = (0,1,x_1)^{\mathrm{T}}$, or equivalently

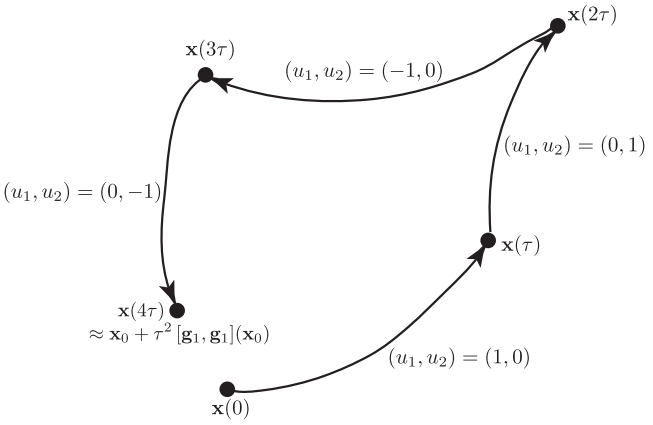

FIG. 23. Lie bracket. The physical meaning of a Lie bracket can be demonstrated by applying the piecewise constant control inputs (51) to a two-inputs driftless control-affine system $\dot{\mathbf{x}} = \mathbf{g}_1(\mathbf{x})u_1 + \mathbf{g}_2(\mathbf{x})u_2$. Up to terms of order τ^2, the difference between the final state $\mathbf{x}(4\tau)$ and the initial state $\mathbf{x}_0$ is given by the Lie bracket $[\mathbf{g}_1,\mathbf{g}_2](\mathbf{x}_0)$.

$$\mathbf{g}_1 = \frac{\partial}{\partial x_1} - x_2 \frac{\partial}{\partial x_3}$$

and

$$\mathbf{g}_2 = \frac{\partial}{\partial x_2} + x_1 \frac{\partial}{\partial x_3}.$$

These two operators $\mathbf{g}_1$ and $\mathbf{g}_2$ have a nontrivial Lie bracket $[\mathbf{g}_1, \mathbf{g}_2](\mathbf{x}) = \mathbf{g}_3(\mathbf{x}) = 2(0,0,1)^{\mathrm{T}}$, or equivalently $\mathbf{g}_3 = 2\partial/\partial x_3$. Consider the system (54) initially at the origin, hence $\mathbf{g}_1(\mathbf{0}) = (1,0,0)^{\mathrm{T}}$, $\mathbf{g}_2(\mathbf{0}) = (0,1,0)^{\mathrm{T}}$. If we again apply the control sequence (51) with time interval $\tau = 1$, we can check that the final state reached at $t = 4$ is $(0,0,2)^{\mathrm{T}}$, which is precisely captured by $[\mathbf{g}_1, \mathbf{g}_2](\mathbf{0})$.

Note that for the Brockett system we have $[\mathbf{g}_1, [\mathbf{g}_1, \mathbf{g}_2]](\mathbf{x}) = [\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]](\mathbf{x}) = \mathbf{0}$. A similar three-dimensional Lie algebra, called the Heisenberg algebra, also arises in quantum mechanics. Hence the Brockett system is also known as the Heisenberg system (Bloch, 2003). Note, however, that the commutation relations obeyed by the Heisenberg algebra do not always apply to general nonlinear systems. To see this consider again the model of a front-wheel drive car (45), representing another two-input control-affine system, where the two control vector fields $\mathbf{g}_1 = (0,0,0,1)^{\mathrm{T}}$ and $\mathbf{g}_2 = (\cos(\theta+\phi), \sin(\theta+\phi), \sin\theta, 0)^{\mathrm{T}}$ can be interpreted as the actions steer and drive, respectively. Some Lie brackets from $\mathbf{g}_1(\mathbf{x})$ and $\mathbf{g}_2(\mathbf{x})$ are

$$\mathbf{g}_3(\mathbf{x}) \equiv [\mathbf{g}_1, \mathbf{g}_2](\mathbf{x}) = \begin{pmatrix} -\sin(\theta+\phi) \\ \cos(\theta+\phi) \\ \cos\theta \\ 0 \end{pmatrix}, \tag{55}$$

$$\mathbf{g}_4(\mathbf{x}) \equiv [[\mathbf{g}_1, \mathbf{g}_2], \mathbf{g}_2](\mathbf{x}) = \begin{pmatrix} -\sin\phi \\ \cos\phi \\ 0 \\ 0 \end{pmatrix}. \tag{56}$$

Equation (55) can be interpreted as $[\text{steer}, \text{drive}] = \text{wriggle}$, arising from the sequence of actions (steer, drive, reverse steer, reverse drive), which is what we do in order to get a car out of a tight parking space. Similarly, Eq. (56) can be interpreted as $[\text{wriggle}, \text{drive}] = \text{slide}$, arising from the sequence of actions (wriggle, drive, reverse wriggle, reverse drive), which is what we do during parallel parking. Equations (55) and (56) indicate that starting from only two control inputs: steer and drive, we can "generate" other actions, e.g., wriggle and slide, which allows us to fully control the car.

These two examples demonstrate that by applying the right sequence of control inputs we can steer the system along a direction that the system does not have direct control over. In general, by choosing more elaborate sequences of control inputs we can steer a control-affine system in directions precisely captured by higher-order Lie brackets, e.g., $[\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]]$, $[[\mathbf{g}_1, \mathbf{g}_2], [\mathbf{g}_2, [\mathbf{g}_1, \mathbf{g}_2]]]$, etc. If the system of interest has a drift term $\mathbf{f}$, we also have to consider Lie brackets involving $\mathbf{f}$. This is the reason why nonlinear controllability is closely related to the Lie brackets.

2. Distributions

To discuss the nonlinear tests of accessibility and controllability, we need the notion of distribution in the sense of differential geometry. A distribution can be roughly considered as the nonlinear version of the controllability matrix of a linear system.

Consider m vector fields $\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_m$ on an open set $\mathcal{D} \subset \mathbb{R}^N$. We denote

$$\Delta(\mathbf{x}) = \text{span}\{\mathbf{g}_1(\mathbf{x}), \mathbf{g}_2(\mathbf{x}), \ldots, \mathbf{g}_m(\mathbf{x})\} \tag{57}$$

as the vector space spanned by the vectors $\mathbf{g}_1(\mathbf{x}), \mathbf{g}_2(\mathbf{x}), \ldots, \mathbf{g}_m(\mathbf{x})$ at any fixed $\mathbf{x} \in \mathcal{D}$. Essentially, we assign a vector space $\Delta(\mathbf{x})$ to each point $\mathbf{x}$ in the set $\mathcal{D}$. The collection of vector spaces $\Delta(\mathbf{x})$, $\mathbf{x} \in \mathcal{D}$, is called a distribution and is referred to by

$$\Delta = \text{span}\{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_m\}. \tag{58}$$

If the vectors $\mathbf{g}_1(\mathbf{x}), \mathbf{g}_2(\mathbf{x}), \ldots, \mathbf{g}_m(\mathbf{x})$ are linearly independent for any $\mathbf{x}$ in $\mathcal{D}$, then the dimension of $\Delta(\mathbf{x})$ is constant and equals m. In this case we call Δ a nonsingular distribution on $\mathcal{D}$. For example, in the Brockett system we have $\mathbf{g}_1(\mathbf{x}) = (1,0,-x_2)^{\mathrm{T}}$, $\mathbf{g}_2(\mathbf{x}) = (0,1,x_1)^{\mathrm{T}}$, and $\mathbf{g}_3(\mathbf{x}) = [\mathbf{g}_1, \mathbf{g}_2](\mathbf{x}) = (0,0,2)^{\mathrm{T}}$. Since $\mathbf{g}_1(\mathbf{x})$, $\mathbf{g}_2(\mathbf{x})$, and $\mathbf{g}_3(\mathbf{x})$ are linearly independent for all $\mathbf{x} \in \mathbb{R}^3$, we conclude that the distribution $\Delta = \text{span}\{\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3\}$ is nonsingular. Similarly, in the front-wheel drive car system of Fig. 22, $\mathbf{g}_1(\mathbf{x})$, $\mathbf{g}_2(\mathbf{x})$, $\mathbf{g}_3(\mathbf{x})$, and $\mathbf{g}_4(\mathbf{x})$ are linearly independent for all $\mathbf{x} \in \mathbb{R}^4$, hence the distribution $\Delta = \text{span}\{\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3, \mathbf{g}_4\}$ is nonsingular. Note that a nonsingular distribution is analogous to a full rank matrix.

D. Nonlinear tests for accessibility

1. Accessibility

Roughly speaking, accessibility concerns whether we can access all directions of the state space from any given state. The accessibility of control-affine systems can be checked using a simple algebraic test based on Lie brackets.

For control-affine systems (47), we denote $\mathcal{C}$ as the linear combinations of recursive Lie brackets of the form

$$[\mathbf{X}_k, [\mathbf{X}_{k-1}, [\ldots, [\mathbf{X}_2, \mathbf{X}_1]\ldots]]], \qquad k = 1, 2, \ldots, \tag{59}$$

where $\mathbf{X}_i$ is a vector field in the set $\{\mathbf{f}, \mathbf{g}_1, \ldots, \mathbf{g}_M\}$. As the linear space $\mathcal{C}$ is a Lie algebra, it is closed under the Lie bracket operation. In other words, $[\mathbf{f}, \mathbf{g}] \in \mathcal{C}$ whenever $\mathbf{f}$ and $\mathbf{g}$ are in $\mathcal{C}$. Hence $\mathcal{C}$ is called the *accessibility algebra*.

The accessibility distribution C is the distribution generated by the accessibility algebra $\mathcal{C}$:

$$C(\mathbf{x}) = \text{span}\{\mathbf{X}(\mathbf{x}) | \mathbf{X} \in \mathcal{C}\}. \tag{60}$$

Consider a control-affine system (47) and a state $\mathbf{x}_0 \in \mathcal{M} \subset \mathbb{R}^N$. If

$$\dim C(\mathbf{x}_0) = N \tag{61}$$

then the system is locally accessible from $\mathbf{x}_0$. Equation (61) is often called the accessibility rank condition (ARC) at $\mathbf{x}_0$. If it holds for any $\mathbf{x}_0$, then the system is called locally accessible.

Interestingly, the sufficient ARC is "almost" necessary for accessibility. Indeed, if the system is accessible then ARC holds for all $\mathbf{x}$ in an open and dense subset of $\mathbb{R}^N$ (Isidori, 1995; Sontag, 1998).

The computation of the accessibility distribution C is nontrivial, because it is not known *a priori* how many (nested) Lie brackets of the vector fields need to be computed until the ARC holds. In practice, a systematic search must be performed by starting with $\{\mathbf{f}, \mathbf{g}_1, \ldots, \mathbf{g}_M\}$ and iteratively generating new, independent vector fields using Lie brackets. This can be achieved by constructing the Philip Hall basis of the Lie algebra, which essentially follows a breadth-first search and the search depth is defined to be the number of nested levels of bracket operations (Serre, 1992; Duleba, 1998).

In general, accessibility does not imply controllability, which is why accessibility is a weaker version of controllability. Consider a simple dynamical system

$$\dot{x}_1 = x_2^2, \qquad \dot{x}_2 = u, \tag{62}$$

which can be written in the control-affine form (47) with $\mathbf{f}(\mathbf{x}) = (x_2^2, 0)^{\mathrm{T}}$ and $\mathbf{g}(\mathbf{x}) = (0, 1)^{\mathrm{T}}$. We can compute some Lie brackets $[\mathbf{f}, \mathbf{g}](\mathbf{x}) = -(2x_2, 0)^{\mathrm{T}}$, $[\mathbf{f}, [\mathbf{f}, \mathbf{g}]](\mathbf{x}) = (2, 0)^{\mathrm{T}}$. Since $[\mathbf{f}, [\mathbf{f}, \mathbf{g}]](\mathbf{x})$ is independent from $\mathbf{g}(\mathbf{x})$, we conclude that $\dim C(\mathbf{x}) = 2$, for any state in $\mathbb{R}^2$, indicating that the system is locally accessible. But the system is not locally controllable: $\dot{x}_1 = x_2^2 > 0$ for all $x_2 \neq 0$, i.e., x_1 always grows as long as the system is not at the x_2 axis. In other words, the drift vector field $\mathbf{f}$ always steers the system to the right unless $x_2 = 0$.

If we compute the accessibility distribution C for a linear system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} = \mathbf{A}\mathbf{x} + \sum_{i=1}^{M} \mathbf{b}_i u_i$ where $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_M]$, we find that $C(\mathbf{x}_0)$ is spanned by $\mathbf{A}\mathbf{x}_0$ together with the constant vector fields $\mathbf{b}_i, \mathbf{A}\mathbf{b}_i, \mathbf{A}^2\mathbf{b}_i, \ldots$, for $i = 1, \ldots, M$. More precisely,

$$C(\mathbf{x}_0) = \mathrm{span}\{\mathbf{A}\mathbf{x}_0\} + \mathrm{Im}(\mathbf{B}, \mathbf{A}\mathbf{B}, \mathbf{A}^2\mathbf{B}, \ldots, \mathbf{A}^{N-1}\mathbf{B}), \tag{63}$$

where Im( ) stands for the image or column space of a matrix. Note that the term $\mathrm{span}\{\mathbf{A}\mathbf{x}_0\}$ does not appear in Kalman's controllability matrix (12). Only at $\mathbf{x}_0 = \mathbf{0}$, Eq. (63) reduces to Kalman's controllability matrix. This shows that accessibility is indeed weaker than controllability, because the former does not imply the latter while the latter induces the former.

2. Strong accessibility

A nonlinear test for strong accessibility tells us whether we can reach states in the neighborhood of the initial state exactly at a given small time. Define $\mathcal{C}_0$ as the strong accessibility algebra, i.e., the smallest algebra which contains $\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_M$ and satisfies $[\mathbf{f}, \mathbf{w}] \in \mathcal{C}_0$, $\forall\ \mathbf{w} \in \mathcal{C}_0$. Note that $\mathcal{C}_0 \subset \mathcal{C}$ and $\mathcal{C}_0$ does not contain the drift vector field $\mathbf{f}$. Define the corresponding strong accessibility distribution

$$C_0(\mathbf{x}) = \mathrm{span}\{\mathbf{X}(\mathbf{x}) | \mathbf{X} \in \mathcal{C}_0\}. \tag{64}$$

If $\dim C_0(\mathbf{x}_0) = N$ then the system is locally strongly accessible from $\mathbf{x}_0$. If this holds for any $\mathbf{x}_0$, then the system is called locally strongly accessible. If we compute the strong accessibility distribution C for a linear system $(\mathbf{A}, \mathbf{B})$, we find that

$$C_0(\mathbf{x}_0) = \mathrm{Im}(\mathbf{B}, \mathbf{A}\mathbf{B}, \mathbf{A}^2\mathbf{B}, \ldots, \mathbf{A}^{N-1}\mathbf{B}). \tag{65}$$

Then $\dim C_0(\mathbf{x}_0) = N$ is equivalent with Kalman's rank condition (13). In other words, strong accessibility and controllability are equivalent notions for linear systems.

E. Nonlinear tests for controllability

For general nonlinear systems, we lack conditions that are both sufficient and necessary for controllability. Yet, as we discuss next, we have some sufficient conditions that are believed to be almost necessary as well.

Consider a special class of control-affine system (47) with $\mathbf{f}(\mathbf{x}) \in \mathrm{span}\{\mathbf{g}_1(\mathbf{x}), \ldots, \mathbf{g}_M(x)\}$ for all $\mathbf{x} \in \mathcal{M} \subset \mathbb{R}^N$. In other words, the drift vector field $\mathbf{f}(\mathbf{x})$, which describes the intrinsic dynamics of the system, can be spanned by the control vector fields $\mathbf{g}_1(\mathbf{x}), \ldots, \mathbf{g}_M(x)$. Then, if $\dim C(\mathbf{x}_0) = N$, the system is locally controllable from $\mathbf{x}_0$. If this holds for all $\mathbf{x} \in \mathcal{M}$, then the system is globally controllable.

Driftless systems $[\mathbf{f}(\mathbf{x}) \equiv \mathbf{0}]$, such as the front-wheel drive car system (45), naturally fall into this class. To see this, we recognize that the determinant of the matrix formed by the vectors $\mathbf{g}_1(\mathbf{x}), \mathbf{g}_2(\mathbf{x}), \mathbf{g}_3(\mathbf{x}) = [\mathbf{g}_1, \mathbf{g}_2](\mathbf{x})$ and $\mathbf{g}_4(\mathbf{x}) = [[\mathbf{g}_1, \mathbf{g}_2], \mathbf{g}_2](\mathbf{x})$, i.e.,

$$\det \begin{pmatrix} 0 & \cos(\theta+\phi) & -\sin(\theta+\phi) & -\sin\phi \\ 0 & \sin(\theta+\phi) & \cos(\theta+\phi) & \cos\phi \\ 0 & \sin\theta & \cos\theta & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \tag{66}$$

is identically equal to 1, regardless of $\mathbf{x}$, implying that $\dim C(\mathbf{x}_0) = N = 4$ for all $\mathbf{x}_0 \in \mathbb{R}^4$. Hence the front-wheel drive car system is globally controllable, in line with our physical intuition and experience.

For control-affine systems that do not fall into the above two classes, Sussmann (1987) provided a general set of sufficient conditions. We call a Lie bracket computed from $\{\mathbf{f}, \mathbf{g}_1, \ldots, \mathbf{g}_M\}$ *bad* if it contains an odd number of $\mathbf{f}$ factors and an even number of each $\mathbf{g}_k$ factors. Otherwise we call it *good*. The degree of a bracket is the total number of vector fields from which it is computed. Denote with $\sum_M$ the permutation group on M symbols. For $\sigma \in \sum_M$ and $\mathbf{b}$ a Lie bracket computed from $\{\mathbf{f}, \mathbf{g}_1, \ldots, \mathbf{g}_M\}$, define $\bar{\sigma}(\mathbf{b})$ as the bracket obtained by fixing $\mathbf{f}$ and changing $\mathbf{g}_k$ by $\mathbf{g}_{\sigma(k)}$, $1 \leq k \leq M$. The control-affine system (47) is locally controllable from $\mathbf{x}_0$ if $\dim C(\mathbf{x}_0) = N$ and every bad bracket $\mathbf{b}$ has the property that $\beta(\mathbf{b})(\mathbf{x}_0) \equiv \sum_{\sigma \in \sum_M} \bar{\sigma}(\mathbf{b})(\mathbf{x}_0)$ is a linear combination of good brackets, evaluated at $\mathbf{x}_0$, of degree lower than $\mathbf{b}$.

F. Controllability of nonlinear networked systems

1. Neuronal network motifs

While most complex systems are described by nonlinear continuous-time dynamics defined over a network, there has been little attention paid so far to the controllability of such systems, due to mathematical challenges. Controllability studies of continuous-time nonlinear dynamics are still limited to very simple networks consisting of a few nodes, such as neuronal network motifs governed by Fitzhugh-Nagumo dynamics (Whalen *et al.*, 2015). These offered an opportunity to study the impact of structural symmetries on nonlinear controllability. The three-node neuronal motifs shown in Fig. 24 can have multiple symmetries. Yet not all symmetries have the same effect on network controllability. For example, with identical nodal and coupling parameters, motif 1 has a full $\mathbf{S}_3$ symmetry, rendering the poorest controllability over the entire range of coupling strengths. Similarly, no controllability is obtained from node 2 in motif 3, which has a reflection $\mathbf{S}_2$ symmetry across the plane through node 2. Surprisingly, the rotational $\mathbf{C}_3$ symmetry in motif 7 does not cause loss of controllability at all. Note that symmetries have an impact on network controllability in linear systems as well. For example, in the case of a directed star with LTI dynamics for which we control the central hub (Fig. 4), a symmetry among the leaf nodes renders the system uncontrollable.

Extending this analysis to larger networks with symmetries remains a challenge, however. Group representation theory might offer tools to gain insight into the impact of symmetries on the controllability of nonlinear networked systems (Whalen *et al.*, 2015). Note, however, that for large real networks such symmetries are less frequent.

2. Boolean networks

The controllability of Boolean networks, a class of discrete-time nonlinear systems that are often used to model gene regulations, has been intensively studied (Akutsu *et al.*, 2007; Cheng and Qi, 2009; Zañudo and Albert, 2015). We can prove that finding a control strategy leading to the desired final state is NP hard for a general Boolean network and this problem can be solved in polynomial time only if the network has a tree structure or contains at most one directed cycle (Akutsu *et al.*, 2007). Interestingly, based on a semitensor product of matrices (Cheng, Qi, and Xue, 2007) and the matrix expression of Boolean logic, the Boolean dynamics can be exactly mapped into the standard discrete-time linear dynamics (Cheng and Qi, 2009). Necessary and sufficient conditions to assure controllability of Boolean networks can then be proved (Cheng and Qi, 2009). Despite the formal simplicity, the price we need to pay is that the size of the discrete-time linear dynamical system is 2^N, where N is the number of nodes in the original Boolean network. Hence, the controllability test will be computationally intractable for large Boolean networks.

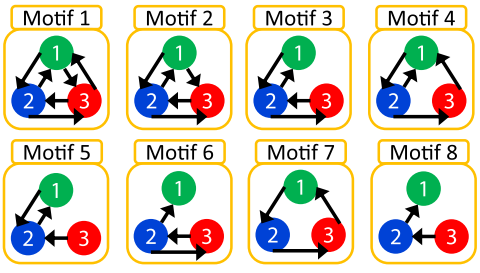

FIG. 24. Symmetries and controllability. The eight different three-node neuronal network motifs studied by Whalen *et al.* (2015). Those motifs display a variety of symmetries. For example, motif 1 has a full $\mathbf{S}_3$ symmetry, and motif 3 has a reflection $\mathbf{S}_2$ symmetry across the plane through node 2. Not all symmetries have the same effect on network controllability.

IV. OBSERVABILITY

Before controlling a system, it is useful to know its position in the state space, allowing us to decide in which direction we should steer it to accomplish the control objective. The position of a system in the state space can be identified only if we can measure the state of all components separately, such as the concentration of each metabolite in a cell, or the current on each transmission line of a power grid. Such detailed measurements are often infeasible and impractical. Instead, in practice we must rely on a subset of well-selected accessible variables (outputs) that can be used to observe the system, i.e., to estimate the state of the system. A system is said to be *observable* if it is possible to recover the state of the whole system from the measured inputs and outputs. This is a fundamental and primary issue in most complex systems.

In general, we can observe a system because its components form a network, hence the state of the nodes depends on the state of their neighbors'. This offers the possibility to estimate all unmeasured variables from the measured ones. If the inputs and model of the system are known, observability can be equivalently defined as the possibility to recover the initial state $\mathbf{x}(0)$ of the system from the output variables.

To be specific, let us assume that we have no knowledge of a system's initial state $\mathbf{x}(0)$, but we can monitor some of its outputs $\mathbf{y}(t)$ in some time interval. The observability problem aims to establish a relationship between the outputs $\mathbf{y}(t)$, the state vector $\mathbf{x}(t)$, and the inputs $\mathbf{u}(t)$ such that the system's initial state $\mathbf{x}(0)$ can be inferred. If no such relation exists, the system's initial state cannot be estimated from the experimental measurements, i.e., the system is not observable. In other words, if the current value of at least one state variable cannot be determined through the outputs sensors, then it remains unknown to the controller. This may disable feedback control, which requires reliable real-time estimates of the system's state.

Note that observability and controllability are mathematically dual concepts. Both concepts were first introduced by Kalman for linear dynamical systems (Kalman, 1963) and were extensively explored in nonlinear dynamical systems by many others (Hermann and Krener, 1977; Diop and Fliess, 1991a, 1991b; Sontag and Wang, 1991; Isidori, 1995; Besanon, 2007).

In this section, we first discuss methods that test the observability of linear and nonlinear control systems. We also discuss the parameter identifiability problem, which is a special case of the observability problem. Finally, we introduce a graphical approach to identify the minimum set of sensor nodes that assure the observability of nonlinear systems

(Siddhartha and van Schuppen, 2001; Aguirre and Letellier, 2005; Letellier and Aguirre, 2005, 2010; Letellier, Aguirre, and Maquet, 2006; Khan and Moura, 2008; Khan and Doostmohammadian, 2011) and its application to metabolic networks (Liu, Slotine, and Barabási, 2013).

A. Observability tests

1. Linear systems

For linear systems there is an exact duality between controllability and observability. To see this, consider an LTI control system

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \tag{67a}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t). \tag{67b}$$

The duality principle states that an LTI system $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ is observable if and only if its dual system $(\mathbf{A}^{\mathrm{T}}, \mathbf{C}^{\mathrm{T}}, \mathbf{B}^{\mathrm{T}})$ is controllable. Mathematically, the duality can be seen and proved from the structure of the controllability Gramian and the observability Gramian. In terms of network language the duality principle has a straightforward interpretation: The linear observability of a network $\mathbf{A}$ can be addressed by studying the controllability of the transposed network $\mathbf{A}^{\mathrm{T}}$, which is obtained by flipping the direction of each link in $\mathbf{A}$ (Fig. 25).

Thanks to the duality principle, many observability tests can be mapped into controllability tests. For example, according to Kalman's rank condition, the system $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ is observable if and only if the observability matrix

$$\mathbf{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{N-1} \end{bmatrix} \tag{68}$$

has full rank, i.e., $\mathrm{rank}\mathbf{O} = N$ (Kalman, 1963; Luenberger, 1979). This rank condition is based on the fact that if the N rows of $\mathbf{O}$ are linearly independent, then each of the N state variables can be determined by linear combinations of the output variables $\mathbf{y}(t)$.

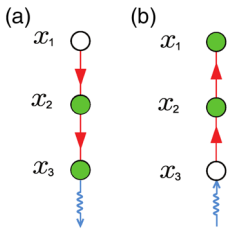

FIG. 25. Duality principle. If a system follows the LTI dynamics (67a), the observability of the network $\mathbf{A}$ shown in (a) can be addressed by studying the controllability of the transposed network $\mathbf{A}^{\mathrm{T}}$ shown in (b), obtained by reversing the direction of each link. This is a general property of all networks.

2. Nonlinear systems

Consider a nonlinear control system with inputs $\mathbf{u}(t) \in \mathbb{R}^K$ and outputs $\mathbf{y}(t) \in \mathbb{R}^M$:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)), \qquad \mathbf{y}(t) = \mathbf{h}(t, \mathbf{x}(t), \mathbf{u}(t)), \tag{69}$$

where $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ are some nonlinear functions.

Mathematically, we can quantify observability from either an algebraic viewpoint (Diop and Fliess, 1991a, 1991b; Conte, Moog, and Perdon, 2007) or a differential geometric viewpoint (Hermann and Krener, 1977). Here we focus on the former. If a system is algebraically observable, then there are algebraic relations between the state variables and the successive derivatives of the system's inputs and outputs (Diop and Fliess, 1991a, 1991b). These algebraic relations guarantee that the system is observable and forbid symmetries. A family of symmetries is equivalent to infinitely many trajectories of the state variables that fit the same specified input-output behavior, in which case the system is not observable. If the number of such trajectories is finite, the system is called locally observable. If there is a unique trajectory, the system is globally observable.

Consider, for example, the dynamical system defined by the equations

$$\dot{x}_1 = x_2 x_4 + u\,, \quad \dot{x}_2 = x_2 x_3\,, \quad \dot{x}_3 = 0\,, \quad \dot{x}_4 = 0\,, \quad y = x_1\,. \tag{70}$$

The system has a family of symmetries σ_λ: $\{x_1, x_2, x_3, x_4\} \to \{x_1, \lambda x_2, x_3, x_4/\lambda\}$ so that the input u and the output y and all their derivatives are independent of λ (Anguelova, 2004). This means that we cannot distinguish whether the system is in state $(x_1, x_2, x_3, x_4)^{\mathrm{T}}$ or its symmetric counterpart $(x_1, \lambda x_2, x_3, x_4/\lambda)^{\mathrm{T}}$, because they are both consistent with the same input-output behavior. Hence we cannot uncover the system's internal state by monitoring x_1 only.

The algebraic observability of a rational system is determined by the dimension of the space spanned by the gradients of the Lie derivatives

$$L_f \equiv \frac{\partial}{\partial t} + \sum_{i=1}^{N} f_i \frac{\partial}{\partial x_i} + \sum_{j\in\mathbb{N}} \sum_{l=1}^{K} u_l^{(j+1)} \frac{\partial}{\partial u_l^{(j)}} \tag{71}$$

of its output functions $\mathbf{h}(t, \mathbf{x}(t), \mathbf{u}(t))$. The observability problem can be further reduced to the so-called rank test: the system (69) is algebraically observable if and only if the $NM \times N$ Jacobian matrix

$$\mathbf{J} = \begin{bmatrix} \frac{\partial L_f^0 h_1}{\partial x_1} & \frac{\partial L_f^0 h_1}{\partial x_2} & \cdots & \frac{\partial L_f^0 h_1}{\partial x_N} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial L_f^0 h_M}{\partial x_1} & \frac{\partial L_f^0 h_M}{\partial x_2} & \cdots & \frac{\partial L_f^0 h_M}{\partial x_N} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial L_f^{N-1} h_1}{\partial x_1} & \frac{\partial L_f^{N-1} h_1}{\partial x_2} & \cdots & \frac{\partial L_f^{N-1} h_1}{\partial x_N} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial L_f^{N-1} h_M}{\partial x_1} & \frac{\partial L_f^{N-1} h_M}{\partial x_2} & \cdots & \frac{\partial L_f^{N-1} h_M}{\partial x_N} \end{bmatrix} \tag{72}$$

has full rank (Diop and Fliess, 1991a, 1991b), i.e.,

$$\text{rank}\mathbf{J} = N. \tag{73}$$

Note that for an LTI system (67a) and (67b), the Jacobian matrix (72) reduces to the observability matrix (68).

For rational dynamical systems, the algebraic observability test can be performed using an algorithm developed by Sedoglavic (2002). The algorithm offers a generic rank computation of the Jacobian matrix (72) using the techniques of symbolic calculation, allowing us to test local algebraic observability for rational systems in polynomial time. This algorithm certifies that a system is locally observable, but its answer for a nonobservable system is probabilistic with a high probability of success. A system that is found nonobservable can be further analyzed to identify a family of symmetries, which can confirm the system is truly nonobservable.

B. Minimum sensors problem

In complex systems, the state variables are rarely independent of each other. The interactions between the system's components induce intricate interdependencies among them. Hence a well-selected subset of state variables can contain sufficient information about the remaining variables to reconstruct the system's complete internal state, making the system observable (Liu, Slotine, and Barabási, 2013).

We assume that we can monitor a selected subset of state variables, i.e., $\mathbf{y}(t) = (\ldots, x_i(t), \ldots)^{\mathrm{T}}$, corresponding to the states of several nodes that we call sensor nodes or just *sensors*. Network observability can then be posed as follows: Identify the minimum set of sensors from whose measurement we can infer all other state variables. For linear systems, this problem can be solved using the duality principle and solving the minimum inputs problem of the transposed network $\mathbf{A}^{\mathrm{T}}$. For general nonlinear systems this trick does not work. While Eq. (73) offers a formal answer to the observability issue and can be applied to small engineered systems, it has notable practical limitations for large and complex systems. First it can confirm only if a specific set of sensors can be used to observe a system or not, without telling us how to identify them. Therefore, a brute-force search for a minimum sensor set requires us to inspect via Eq. (73) about 2^N sensor combinations, a computationally prohibitive task for large systems. Second, the rank test of the Jacobian matrix via symbolic computation is computationally limited to small systems (Sedoglavic, 2002).

To resolve these limitations, we can exploit the dynamic interdependence of the system's components through a graphical representation (Lin, 1974; Reinschke, 1988; Siddhartha and van Schuppen, 2001; Murota, 2009; Khan and Doostmohammadian, 2011). The procedure consists of the following steps (Liu, Slotine, and Barabási, 2013):

(i) Inference diagram: Draw a directed link $x_i \to x_j$ if x_j appears in x_i's differential equation (i.e., if $\partial f_i/\partial x_j$ is not identically zero), implying that one can retrieve some information on x_j by monitoring x_i as a function of time. Since the constructed network captures the information flow to infer the state of individual variables, we call it the inference diagram [Fig. 26(c)].

(ii) SCC decomposition: Decompose the inference diagram into a unique set of maximal SCCs [dashed circles in Fig. 26(c)], i.e., the largest subgraphs chosen such that in each of them there is a directed path from every node to every other node (Cormen, Leiserson, and Rivest, 1990). Consequently, each node in an SCC contains some information about all other nodes within the SCC.

(iii) Sensor node selection: Those SCCs that have no incoming edges are referred to as root SCCs [shaded circles in Fig. 26(c)]. We must choose at least one node from each root SCC to ensure the observability of the whole system. For example, the inference diagram of Fig. 26(c) contains three root SCCs; hence we need at least three sensors to observe the system.

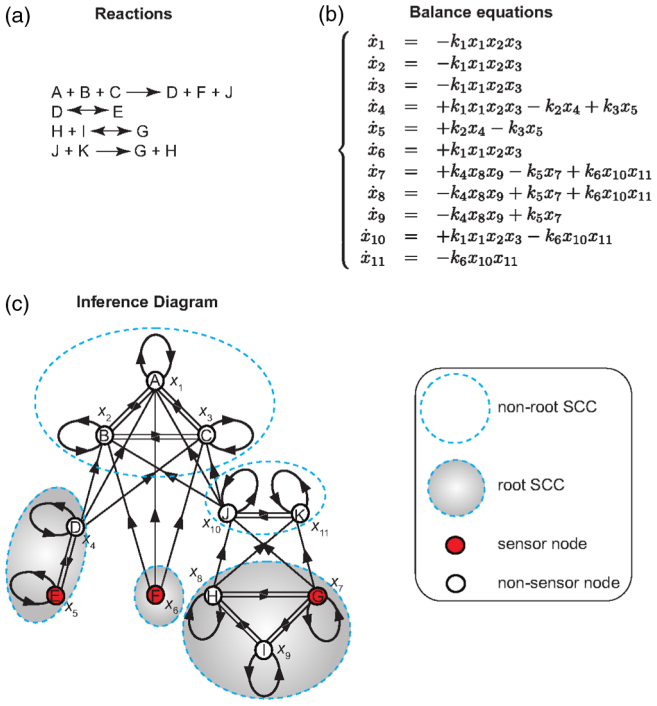

FIG. 26. The graphical approach to determine the minimum sensors of a chemical reaction system. (a) A chemical reaction system with 11 species (A, B, …, J, K) involved in four reactions. Since two reactions are reversible, we have six elementary reactions. (b) The balance equations of the chemical reaction system shown in (a). The concentrations of the 11 species are denoted by $x_1, x_2, \ldots, x_{11}$, respectively. The rate constants of the six elementary reactions are given by $k_1, k_2, \ldots, k_6$, respectively. The balance equations are derived using the mass-action kinetics. (c) The inference diagram is constructed by drawing a directed link $(x_i \to x_j)$ as long as x_j appears in the right-hand side (rhs) of x_i's balance equation shown in (b). SCCs are marked with dashed circles. Root SCCs, which have no incoming links, are shaded in gray. A potential minimum set of sensor nodes, whose measurements allow us to reconstruct the state of all other variables (metabolite concentrations), are shown in red. From Liu, Slotine, and Barabási, 2013.

The graphical approach (GA) described can be used to determine whether a variable provides full observability of small dynamical systems (Letellier and Aguirre, 2005; Aguirre *et al.*, 2008). As these systems have only a few state variables, steps (ii) and (iii) are often not necessary. For large networked systems, the GA is very powerful because it reduces the observability issue, a dynamical problem of a nonlinear system with many unknowns, to a property of the static graph of the inference diagram, which can be accurately mapped for an increasing number of complex systems, from biochemical reactions to ecological systems.

We can prove that monitoring the root SCCs identified by the GA are necessary for observing any nonlinear dynamic system (Liu, Slotine, and Barabási, 2013). In other words, the number of root SCCs yields a strict lower bound for the size of the minimum sensor set. Consequently, any state observer (i.e., a dynamical device that aims to estimate the system's internal state) will fail if it does not monitor these sensors.

If the dynamics is linear, the duality principle maps the minimum sensors problem into the minimum inputs problem and predicts not only the necessary, but also the sufficient sensor set for observability. Numerical simulations on model networks suggest that for linear systems the sufficient sensor set is noticeably larger than the necessary sensor set predicted by the GA (Liu, Slotine, and Barabási, 2013). This is because any symmetries in the state variables leaving the inputs, outputs, and all their derivatives invariant will make the system unobservable (Sedoglavic, 2002). For structured linear systems, the symmetries correspond to a particular topological feature, i.e., dilations, which can be detected from the inference diagram. Yet, for general nonlinear systems, the symmetries cannot be easily detected from the inference diagram only.

For linear systems the minimum sensor set predicted by the GA is generally not sufficient for full observability. Yet, for large nonlinear dynamical systems the symmetries in state variables are extremely rare, especially when the number of state variables is large; hence the sensor set predicted by GA is often not only necessary but also sufficient for observability (Liu, Slotine, and Barabási, 2013).

To better understand network observability, next we apply the developed tools to biochemical and technological networks.

1. Biochemical reaction systems

Consider a biochemical reaction system of N species $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_N\}$ involved in R reactions $\{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_R\}$ with

$$\mathcal{R}_j: \sum_{i=1}^{N} \alpha_{ji}\mathcal{S}_i \rightarrow \sum_{i=1}^{N} \beta_{ji}\mathcal{S}_i, \tag{74}$$

where $\alpha_{ji} \geq 0$ and $\beta_{ji} \geq 0$ are the stoichiometry coefficients. For example, Eq. (74) captures the reaction $2H_2 + O_2 = 2H_2O$ with $\alpha_{11} = 2$, $\alpha_{12} = 1$, and $\beta_{11} = 2$.

Under the continuum hypothesis and the well-mixed assumption the system's dynamics is described by Eq. (69), where $x_i(t)$ is the concentration of species $\mathcal{S}_i$ at time t, the input vector $\mathbf{u}(t)$ represents regulatory signals or external nutrient concentrations, and the vector $\mathbf{y}(t)$ captures the set of experimentally measurable species concentrations or reaction fluxes. The vector $\mathbf{v}(\mathbf{x}) = (v_1(\mathbf{x}), v_2(\mathbf{x}), \ldots, v_R(\mathbf{x}))^{\mathrm{T}}$ is often called the flux vector, which follows the mass-action kinetics (Heinrich and Schuster, 1996; Palsson, 2006)

$$v_j(\mathbf{x}) = k_j \prod_{i=1}^{N} x_i^{\alpha_{ji}} \tag{75}$$

with rate constants $k_j > 0$. The system's dynamics is therefore described by the balance equations

$$\dot{x}_i = f_i(\mathbf{x}) = \sum_{j=1}^{R} \Gamma_{ij} v_j(\mathbf{x}), \tag{76}$$

where $\Gamma_{ij} = \beta_{ji} - \alpha_{ji}$ are the elements of the $N \times R$ stoichiometric matrix $\mathbf{\Gamma}$. The rhs of Eq. (76) represents a sum of all fluxes v_j that produce and consume the species $\mathcal{S}_i$.

Assuming that the outputs $\mathbf{y}(t)$ are just the concentrations of a particular set of sensor species that can be experimentally measured, then the observability problem aims to identify a minimum set of sensor species from whose measured concentrations we can determine all other species' concentrations. In this context, the advantage of the GA is that it does not require the system's kinetic constants (which are largely unknown *in vivo*), relying only on the topology of the inference diagram. For a metabolic network or an arbitrary biochemical reaction system, the topology of the inference diagram is uniquely determined by the full reaction list, which is relatively accurately known for several model organisms (Schellenberger *et al.*, 2010). Applying the GA to biochemical reaction systems offers several interesting results, elucidating the principles behind biochemical network observability (Liu, Slotine, and Barabási, 2013):

(a) Species that are not reactants in any reaction, being instead pure products, will be root SCCs of size 1. Consequently, they are always sensors and must be observed by the external observer [e.g., x_6 in Fig. 26(c)].

(b) For root SCCs of size larger than 1 [e.g., $\{x_4, x_5\}$ and $\{x_7, x_8, x_9\}$ in Fig. 26(c)], any node could be chosen as a sensor. Given that some root SCCs are quite large, and typically we need to monitor only one node for each root SCC, the number of sensor nodes is thus considerably reduced.

(c) A minimum set of sensors consists of all pure products and one node from each root SCC of size larger than 1 [e.g., $\{x_5, x_6, x_7\}$ in Fig. 26(c)].

(d) Since any node in a root SCC can be selected as a sensor node, there are $\Omega_s = \prod_{i=1}^{N_{\text{root-SCC}}} n_i$ equivalent sensor node combinations, representing the product of all root SCCs' sizes. For example, in Fig. 26(c) we have three root SCCs with sizes $n_i = 1$, 2, and 3, hence $\Omega_s = 1 \times 2 \times 3 = 6$. This multiplicity offers significant flexibility in selecting experimentally accessible sensors.

It turns out that the minimum set of sensors obtained by the GA almost always achieves full observability for the whole system, except in some pathological cases (Liu, Slotine, and Barabási, 2013). The sufficiency of the sensors predicted by the GA is rather unexpected because substantial details about the system's dynamics are ignored in the GA. Offering an exact proof that the sufficiency of the predicted sensors for observability is rather difficult, if not impossible. The rigorous proof of sufficiency and the systematic search for exceptional cases making a system unobservable remain open questions.

2. Power grid

In a power grid, the state variables can be chosen to represent the voltage of all nodes, which in practice can be determined by phasor measurement units (PMUs). Since a PMU can measure the real-time voltage and line currents of the corresponding node, a PMU placed on a node i will determine the state variables of both node i and all of its first nearest neighbors. In this case the observability problem can be mapped to a purely graph-theoretical problem. The random placement of PMUs leads to a network observability transition (Yang, Wang, and Motter, 2012), which is a new type of percolation transition that characterizes the emergence of macroscopic observable components in the network as the number of randomly placed PMUs increases (Fig. 27). Using the generating function formalism (Newman, Strogatz, and Watts, 2001), we can analytically calculate the expected size of the largest observable component for networks with any prescribed degree distribution. This has been demonstrated for real power grids (Yang, Wang, and Motter, 2012). Moreover, it has been found that the percolation threshold decreases with the increasing average degree or degree heterogeneity (Yang, Wang, and Motter, 2012).

The random placement of PMUs apparently will not solve the minimum sensors problem. For a power grid, the problem of identifying the minimum set of sensor nodes is reduced to

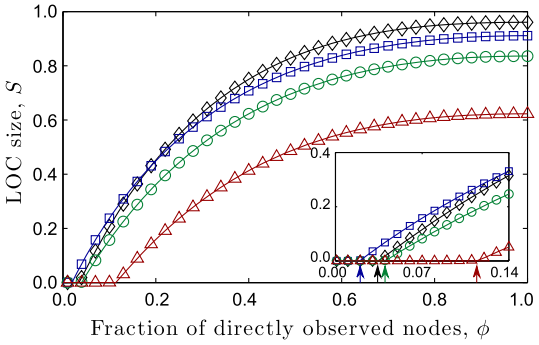

FIG. 27. Observability transitions in power grids. (a) Fraction of the largest observable component as a function of the fraction of directly observed nodes (ϕ) in networks with prescribed degree distributions of the power grids of Eastern North America (black diamonds), Germany (red triangles), Europe (green circles), and Spain (blue squares). The continuous lines are analytical predictions, and the symbols represent the average over ten 10^6-node random networks for ten independent random PMU placements each. The inset shows a magnification around the transitions, with the analytically predicted thresholds ϕ_c indicated by arrows. From Yang, Wang, and Motter, 2012.

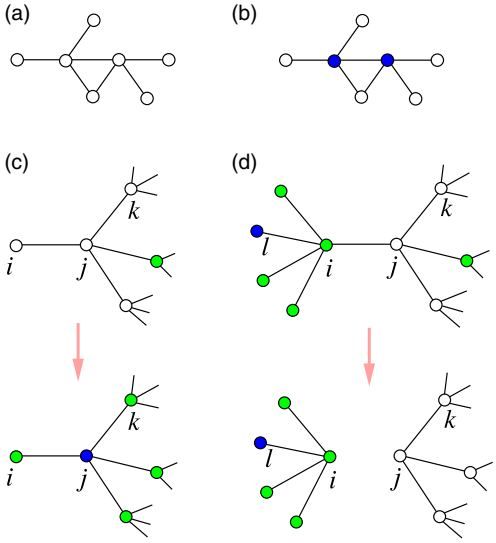

FIG. 28. Dominating set and generalized leaf removal process. (a), (b) Dominating set. A dominating set of a graph $G = (V, E)$ is a subset D of V such that every vertex not in D is adjacent to at least one vertex in D. A minimum dominating set (MDS, shown in blue) is a dominating set of the smallest size. (c), (d) Generalized leaf removal (GLR) process. If a network is sufficiently sparse, then its MDS can be found exactly using GLR, consisting of two basic operations illustrated in (c) and (d). Dark shaded (blue) circles denote nodes occupied with sensor nodes. White circles denote empty (i.e., nonoccupied) and unobservable nodes. Light shaded (green) circles denote empty but observable nodes. (c) For an empty leaf node i, its only adjacent node j must be occupied, i.e., be chosen as a sensor node. Consequently, all adjacent nodes of j are observed. Node j and its adjacent nodes can be removed from the network to simplify the MDS problem. (d) If an empty observed node i has only a single unobserved adjacent node j, then it must be an optimal strategy not to occupy node i. Hence, the link between i and j can be removed from the network to simplify the MDS problem. From Zhao, Habibulla, and Zhou, 2015.

the minimum dominating set (MDS) problem: Identify a minimum node set $D \subseteq V$ for a graph $G = (V, E)$ such that every node not in D is adjacent to at least one node in D [Figs. 28(a) and 28(b)]. Consider a undirected network G. Node i is either empty (with occupation state $c_i = 0$) or occupied by a sensor (with $c_i = 1$). In other words, if $c_i = 1$ then node i can be considered a sensor node. Node i is called observed if it is a sensor node itself or it is not a sensor node but adjacent to one or more sensor nodes. Otherwise node i is unobserved. The MDS problem requires us to occupy a minimum set D of nodes so that all N nodes of G are observed.[5]

The MDS problem for a general graph is NP hard, and the best polynomial algorithms can offer only dominating sets

[5]Interestingly, the MDS problem can also be formalized as a control problem on a undirected network by assuming that every edge in a network is bidirectional and every node in the MDS can control all of its outgoing links separately (Jose and Tatsuya, 2012). This formulation has recently been applied to analyze biological networks (Wuchty, 2014; Nacher and Akutsu, 2016).

with sizes not exceeding $\log N$ times of the minimum size of the dominating sets (Lund and Yannakakis, 1994; Raz and Safra, 1997). Recently, probabilistic methods have been developed to approximate the size of the MDS in scale-free networks (Molnár *et al.*, 2014).

Note that if the underlying network has no core, we can exactly solve the MDS problem in polynomial time using a generalized leaf removal (GLR) process [Figs. 28(c) and 28(d)]. The GLR process can be recursively applied to simplify the network G. If eventually all the nodes are removed, then the set of nodes occupied during this process must be an MDS and choosing them as sensor nodes will make the whole network observable (Zhao, Habibulla, and Zhou, 2015). If, however, the final simplified network is nonempty, then there must be some nodes that are still unobserved after the GLR process. The subnetwork induced by these unobserved nodes is referred to as the core of the original network G. For networks with an extensive core, a belief-propagation algorithm, rooted in spin glass theory, can offer nearly optimal solutions, which also performs well on real-world networks (Zhao, Habibulla, and Zhou, 2015).

C. Target observability

In many applications it is overkill to observe the full system, but it is sufficient to infer the state of a subset of target variables. Such target variables could, for example, correspond to the concentrations of metabolites whose activities are altered by a disease (Barabási, Gulbahce, and Loscalzo, 2011), representing potential biomarkers. In case those target variables cannot be directly measured, we can invoke target observability and aim to identify the optimal sensor(s) that can infer the state of the target variables. These could represent the optimal experimentally accessible biomarkers for a disease. The graphical approach discussed above helps us select such optimal sensors as follows: (a) The state of a target node x_t can be observed from a sensor node x_s only if there is a directed path from x_s to x_t in the inference diagram. For example, in Fig. 26(c), x_4 can be inferred only from x_5 while x_1 can be inferred from any other nodes. (b) There are important differences in the complexity of the inference process, which depends on the size of the subsystem we need to infer for a given sensor choice. The SCC decomposition of the inference diagram indicates that to observe x_t from x_s, we need to reconstruct $\mathcal{N}_s = \sum_{n_i \subset \mathcal{S}_s} n_i$ metabolite concentrations, where $\mathcal{S}_s$ denotes the set of all SCCs that are reachable from x_s, and n_i is the size of the ith SCC. This formula can be extended to multiple targets. (c) To identify the optimal sensor node for any target node, we can minimize $\sum_{n_i \subset \mathcal{S}_s} n_i$, which is the minimum amount of information required for the inference process. For example, if x_t is inside an SCC of size larger than 1 [e.g., x_1 in Fig. 26(c)], then the optimal sensor can be any other node in the same SCC [e.g., x_2 or x_3 in Fig. 26(c)]. If all other nodes in the same SCC is experimentally inaccessible, then the optimal sensor node belongs to the smallest SCC that points to x_i [e.g., x_6 in Fig. 26(c)]. Note that this minimization procedure can be implemented for any inference diagram in polynomial time. Hence the graphical approach can aid the efficient selection of optimal sensors for any targeted node, offering a potentially indispensable tool for biomarker design.

D. Observer design

The observability test and the graphical approach mentioned above do not tell us how to reconstruct the state of the system from measurements. To achieve this we must design an *observer*, a dynamic device that runs a replica of the real system, adjusting its state from the available outputs to uncover the missing variables.

For an LTI system (67a) and (67b), we can design the so-called Luenberger observer (Luenberger, 1964, 1966, 1971)

$$\dot{\mathbf{z}}(t) = \mathbf{A}\mathbf{z}(t) + \mathbf{L}[\mathbf{y}(t) - \mathbf{C}\mathbf{z}(t)] + \mathbf{B}\mathbf{u}(t), \tag{77}$$

where the $N \times K$ matrix $\mathbf{L}$ is to be specified later. Note that with initial condition $\mathbf{z}(0) = \mathbf{x}(0)$, the Luenberger observer will follow $\mathbf{z}(t) = \mathbf{x}(t)$ exactly for all $t > 0$. Because $\mathbf{x}(0)$ is typically unaccessible, we start from $\mathbf{z}(0) \neq \mathbf{x}(0)$ and hope that $\mathbf{z}(t)$ will asymptotically converge to $\mathbf{x}(t)$, i.e., the state of the observer tracks the state of the original system. This can be achieved by choosing a proper $\mathbf{L}$ matrix such that the matrix $[\mathbf{A} - \mathbf{L}\mathbf{C}]$ is asymptotically stable, in which case the error vector $\mathbf{e}(t) = \mathbf{z}(t) - \mathbf{x}(t)$, satisfying $\dot{\mathbf{e}}(t) = [\mathbf{A} - \mathbf{L}\mathbf{C}]\mathbf{e}(t)$, will converge to zero with a rate determined by the largest eigenvalue of $[\mathbf{A} - \mathbf{L}\mathbf{C}]$.

For nonlinear systems the observer design is rather involved and still an open challenge (Friedland, 1996; Besançon, 2007).

1. Parameter identification

Most modeling efforts assume that the system parameters, such as the rate constants of biochemical reactions, are known. Yet, for most complex systems, especially in biological context, the system parameters are usually unknown or are known only approximately. Furthermore, the known parameters are typically estimated *in vitro*, and their *in vivo* relevance is often questionable. This raises a natural question: Can we determine the model parameters through appropriate input and output measurements, such as monitoring the concentrations of properly selected chemical species? This problem is called parameter identification (PI) in control theory (Bellman and Aström, 1970; Pohjanpalo, 1978; Glad and Ljung, 1990; Ljung, 1999; Saccomani, Audoly, and D'Angio, 2003).

We can formalize the parameter identifiability problem as the observability problem of an extended system as follows (Anguelova, 2004). For this we consider the system parameters Θ as special state variables with time-derivative zero ($d\Theta/dt = 0$). We can extend the state vector to include a larger set of state variables, i.e., $(\mathbf{x}(t), \Theta)$, allowing us to formally determine whether and how the system parameters can be identified from the input-output behavior by checking the observability of the extended system. Consequently, PI can be considered as a special observer design problem.

2. Network reconstruction

When the system parameters contain information about the network structure, the corresponding PI problem can be generalized to a network reconstruction (NR) problem. Consider a network whose state variables are governed by a set of ordinary differential equations (ODEs)

$$\dot{x}_i(t) = \sum_{j=1}^{N} a_{ij} f_{ij}(x_i(t), x_j(t)) + u_i(t), \tag{78}$$

where $i = 1, \ldots, N$; the coupling functions $f_{ij}: \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ capture the interactions between nodes: self-interactions when $i = j$ or pairwise interactions when $i \neq j$. The term $u_i(t) \in \mathbb{R}$ represents either known signals or control inputs that can affect node i's state. The interaction matrix $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{N \times N}$ captures the directed interactions between the nodes: $a_{ji} \neq 0$ if node j directly affects node i's dynamics. Given measured temporal data $\{x_i(t), u_i(t)\}_{i=1}^{N}$, $\forall t \in [t_0, t_1]$, NR aims to recover some properties of the $\mathbf{A}$ matrix, e.g., its sign pattern $\mathbf{S} = [s_{ij}] = [\mathrm{sign}(a_{ij})] \in \{-1, 0, 1\}^{n \times n}$, connectivity pattern $\mathbf{C} = [c_{ij}] = [|s_{ij}|] \in \{0, 1\}^{n \times n}$, adjacency pattern $\mathbf{K} = [k_{ij}] = [c_{ij}(1 - \delta_{ij})] \in \{0, 1\}^{n \times n}$ (δ_{ij} is the Kronecker delta), or in-degree sequence $\mathbf{d} = [d_i] = [\sum_j c_{ij}] \in \mathbb{Z}^n$. Note that PI aims to recover the $\mathbf{A}$ matrix itself.

There are three principally different NR approaches, which assume various levels of *a priori* knowledge about the system (Timme and Casadiego, 2014).

a. Driving response

Here we try to measure and evaluate the collective response of a networked system to external perturbations or driving. As the response depends on both the external driving signal (which unit is perturbed, when and how strong is the perturbation, etc.) and the (unknown) structural connectivity of the network, sufficiently many driving-response experiments should reveal the entire network. This approach is relatively simple to implement and the required computational effort scales well with the system size. It has been well established for the reconstruction of gene regulatory networks (Gardner *et al.*, 2003; Tegnér *et al.*, 2003; Yu, 2010; Yu and Parlitz, 2010). Yet this approach requires us to measure and drive the dynamics of all units in the system, which is often infeasible. The collective dynamics suitable for the driving-response experiments also needs to be simple (i.e., to exhibit a stable fixed point or periodic orbits, or to allow the system to be steered into such a state). For systems exhibiting more complex features, e.g., chaos, bifurcations, or multistability, this approach is not applicable. If the system exhibits the same fixed point for different constant inputs (as some biological systems that have "perfect adaptation"), it is impossible to reconstruct the network using driving-response experiments (Prabakaran, Gunawardena, and Sontag, 2014).

b. Copy synchronization

This approach sets up a copy of the original system and updates its interaction matrix continuously until the copy system synchronizes its trajectories with the original system (Yu, Righero, and Kocarev, 2006). We expect the final interaction matrix of the copy system to converge to that of the original system. Unfortunately, sufficient conditions for the convergence of this approach have not been fully understood and the approach is model dependent. Knowing the details of the coupling functions $f_{ij}(x_i, x_j)$ is crucial to set up the copy system. Furthermore, $f_{ij}(x_i, x_j)$ needs to be Lipschitz continuous. These constraints significantly narrow the applicability of this approach.

c. Direct approach

This approach relies on the evaluation of temporal derivatives from time series data (Shandilya and Timme, 2011). Exploiting smoothness assumptions, it finds the unknown interaction matrix by solving an optimization problem (e.g., ℓ_1 or ℓ_2-norm minimization). The rationale is as follows. If the time derivatives of the state variables are evaluated, and if the system coupling functions are also known, then the only remaining unknown parameters are the edge weights or interaction strengths a_{ij}'s. Repeated evaluations of Eq. (78) at different sufficiently closely spaced times $t_m \in \mathbb{R}$ comprise a simple and implicit restriction on the interaction matrix A. This approach serves as a simple starting strategy of NR. Yet it has a fundamental drawback—there is no reason why the true interaction matrix should be optimal in some *a priori* metric. Moreover, it may suffer from the poor evaluation of time derivatives of noisy time series data.

All three approaches suffer from one common issue: The necessary and sufficient conditions under which they succeed are unknown. An important exception is the modular response analysis method (Kholodenko *et al.*, 2002; Sontag, 2008), which is a special driving-response approach and guarantees to recover the interaction matrix using steady-state data collected from sufficiently many perturbation experiments.

Recently, two classes of fundamental limitations of NR were characterized by deriving necessary (and in some cases sufficient) conditions to reconstruct any desired property of the interaction matrix (Angulo *et al.*, 2015). The first class of fundamental limitations is due to our uncertainty about the coupling functions $f_{ij}(x_i, x_j)$, leading to a natural trade-off: the more information we want to reconstruct about the interaction matrix the more certain we need to be about the coupling functions. For example, it is possible to reconstruct the adjacency pattern $\mathbf{K}$ without knowing exactly the coupling functions. But, in order to reconstruct the interaction matrix $\mathbf{A}$ itself, it is necessary to know these functions exactly. In this sense, if we are uncertain about the coupling functions, NR is easier than PI. The second class of fundamental limitations originates solely from uninformative temporal data. Here by uninformative temporal data we mean that the measured experimental signals are not rich enough to span the frequencies of dynamical interest. This leads to a rather counterintuitive result: regardless of how much information we aim to reconstruct (e.g., edge weights, sign pattern, or connectivity pattern), the measured temporal data need to be equally informative. This happens even if we know the coupling functions exactly. Hence, in the sense of informativeness of the measured data, reconstructing any property of the

interaction matrix is as difficult as reconstructing the interaction matrix itself, i.e., NR is as difficult as PI. A practical solution to circumvent this limitation without acquiring more temporal data (i.e., performing more experiments, which are sometimes either infeasible or too expensive) is to leverage prior knowledge of the interaction matrix, e.g., the bounds of the edge weights (Angulo *et al.*, 2015).

V. TOWARD DESIRED FINAL STATES OR TRAJECTORIES

A significant body of work in control theory deals with the design of control inputs that can move the system from a given initial state to a desired final state in the state space (Sontag, 1998). For linear dynamics, Eq. (35) provides the optimal input signal to take an arbitrary linear system into an arbitrary final state using the minimum control energy $\int_0^T \|\mathbf{u}(t)\|^2 dt$. For nonlinear dynamics we lack a ready-to-use solution and finding one can be very difficult. Yet, solving such nonlinear control problems has important applications from robotics to ecosystem management, and from cell reprogramming to drug discovery. For example, in robotics engineers frequently encounter the so-called motion- or path-planning problem, needing to decompose a desired movement into discrete motions that satisfy specific movement constraints and possibly optimize some aspect of the trajectory. The parallel parking problem is a typical example, requiring us to determine the sequence of motions a car must follow in order to parallel park into a parking space.

In many cases, we are interested in steering the system toward a desired trajectory or attractor, instead of an arbitrary final state. A trajectory or an orbit of a dynamical system is a collection of points (states) in the state space. For example, a periodic orbit repeats itself in time with period T, so that $\mathbf{x}(t) = \mathbf{x}(t + nT)$ for any integer $n \geq 1$. Roughly speaking, an attractor is a closed subset $\mathcal{A}$ of a dynamical system's state space such that for "many" choices of initial states the system will evolve toward states in $\mathcal{A}$ (Milnor, 2006). Simple attractors correspond to fundamental geometric objects, such as points, lines, surfaces, spheres, toroids, manifolds, or their simple combinations. Fixed (or equilibrium) point and limit cycle are common simple attractors. Fixed points are defined for mappings $x_{n+1} = f(x_n)$, where x is a fixed point if $x = f(x)$, whereas equilibrium points or equilibria are defined for flows (ODEs) $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$, where $\mathbf{x}$ is an equilibrium point if $\mathbf{f}(\mathbf{x}) = 0$. A limit cycle is a periodic orbit of the dynamical system that is isolated. An attractor is called *strange* if it has a fractal structure that cannot be easily described as fundamental geometric objects or their simple combinations. A strange attractor often emerges in chaotic dynamics.

In this section we briefly review progress made in several directions with the common goal of controlling nonlinear dynamical systems: (a) Control of chaos, which requires us to transform a chaotic motion into a periodic trajectory using open-loop control (Hubler *et al.*, 1988), Poincaré map linearization (Ott, Grebogi, and Yorke, 1990) or time-delayed feedback (Pyragas, 1992). (b) Systematic design of compensatory perturbations of state variables that take advantage of the full basin of attraction of the desired final state (Cornelius, Kath, and Motter, 2013). (c) Construction of the attractor network (Lai, 2014; Wang *et al.*, 2014). (d) Mapping the control problem into a combinatorial optimization problem on the underlying network (Fiedler *et al.*, 2013; Mochizuki *et al.*, 2013).

A. Controlling chaos

A deterministic dynamical system is said to be *chaotic* if its evolution is highly sensitive to its initial conditions. This sensitivity means that arbitrary small measurement errors in the initial conditions grow exponentially with time, destroying the long-term predictability of the system's future state. This phenomenon, known as the *butterfly effect*, is often considered troublesome (Lorenz, 1963). Chaotic behavior commonly emerges in natural and engineered systems, being encountered in chemistry, nonlinear optics, electronics, fluid dynamics, meteorology, and biology (Strogatz, 1994).

It has been realized that well-designed control laws can overcome the butterfly effect, forcing chaotic systems to follow some desired behavior (Hubler *et al.*, 1988; Ott, Grebogi, and Yorke, 1990; Pyragas, 1992; Toroczkai, 1994; Sass and Toroczkai, 1996). Next, we review several key methods devised for the control of chaotic systems from the theoretical perspective (Chen and Dong, 1998; Boccaletti *et al.*, 2000; Fradkov and Evans, 2005).

1. Open-loop control

Since the late 1980s, a series of methods have emerged to manipulate chaotic systems toward a desired "goal dynamics" $\mathbf{g}(t)$ (Hubler *et al.*, 1988). Consider a controlled system

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}) + \mathbf{B}\mathbf{u}(t), \tag{79}$$

where $\mathbf{x} \in \mathbb{R}^N$ is the state vector and $\mathbf{u}(t) \in \mathbb{R}^M$ is the control input. In contrast with the network-based problems discussed earlier, here we assume that all state variables are controlled ($M = N$) and $\det \mathbf{B} \neq 0$. The goal is to design $\mathbf{u}(t)$ so that $\mathbf{x}(t)$ converges to a desired trajectory $\mathbf{g}(t)$, i.e., $|\mathbf{x}(t) - \mathbf{g}(t)| \to 0$ as $t \to \infty$. We can use open-loop control for this purpose, using the control input called the *Hubler action*,

$$\mathbf{u}(t) = \mathbf{B}^{-1}[\dot{\mathbf{g}}(t) - \mathbf{F}(\mathbf{g}(t))], \tag{80}$$

which ensures that $\mathbf{x}(t) = \mathbf{g}(t)$ is a solution of the controlled system. In this case, the error $\mathbf{e}(t) = \mathbf{x}(t) - \mathbf{g}(t)$ satisfies

$$\dot{\mathbf{e}}(t) = \mathbf{F}(\mathbf{e}(t) + \mathbf{g}(t)) - \mathbf{F}(\mathbf{g}(t)), \tag{81}$$

which can be linearized as $\dot{\mathbf{e}}(t) = \mathbf{A}(t)\mathbf{e}(t)$, where $\mathbf{A}(t) = \partial \mathbf{F}(\mathbf{x})/\partial \mathbf{x}|_{\mathbf{x}=\mathbf{g}(t)}$. If the linearized system is uniformly asymptotically stable, i.e., its equilibrium point $\mathbf{e}^* = \mathbf{0}$ is stable for all $t > 0$, then the error $\mathbf{e}(t)$ converges to zero, and $\mathbf{x}(t)$ converges to the desired trajectory $\mathbf{g}(t)$. We call the regions of the state space from which the controlled orbits converge to the goal trajectory $\mathbf{g}(t)$ entrainment regions.

Note that the method (79) and (80) is not tailored to chaotic systems, but potentially works for any nonlinear system. It has

several disadvantages, though, such as the following: (i) the open-loop control (80) requires *a priori* knowledge of the dynamics, which is often not precisely known for complex systems; (ii) the applied controls are not always small, requiring high control energy; and (iii) the convergence of $|\mathbf{x}(t)-\mathbf{g}(t)|\to 0$ for $t\to\infty$ depends on the detailed functional form of $\mathbf{F}(\mathbf{x})$ and the initial condition $\mathbf{x}(0)$, hence this method is not guaranteed to work for arbitrary systems.

2. Linearization of the Poincaré map: OGY method

The OGY method proposed by Ott, Grebogi, and Yorke (1990) exploits the observation that typically an infinite number of unstable periodic orbits (UPOs) are embedded in a chaotic attractor (Fig. 29). Therefore we can obtain a desired periodic motion by making only small perturbations to an accessible system parameter.

The OGY method can be summarized as follows: First, we determine and examine some of the low-period UPOs embedded in the chaotic attractor. Second, we choose a desired UPO. Finally, we design small time-dependent parameter perturbations to stabilize this preexisting UPO.

This method is not only very general and practical, but also suggests that in some systems the presence of chaotic behavior can be an advantage for control. Indeed, if the attractor of a system is not chaotic but has a stable periodic orbit, then small parameter perturbations can only slightly change the existing orbit. Therefore, given that any one of the infinite number of UPOs can be stabilized, we can always choose the UPO that achieves the best system performance. Hence, chaotic behavior offers us a diverse and rich landscape for the desired dynamic behavior of the system.

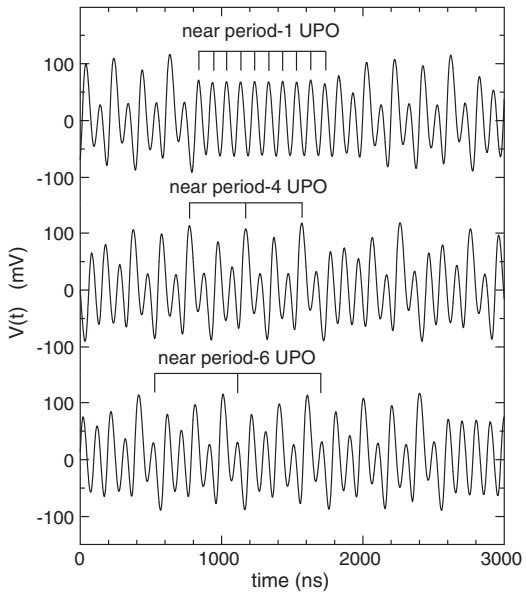

FIG. 29. Chaotic behavior in a nonlinear electronic circuit. The vertical axis measures the voltage drop $V(t)$ across a 50 Ω resistor, being proportional to the current in the circuit. The system ergodically visits the unstable periodic orbits (UPOs) embedded in the chaotic attractor. The plot shows three such UPOs. From Sukow *et al.*, 1997.

To demonstrate this method, let us consider a nonlinear continuous-time dynamical system

$$\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x},u), \tag{82}$$

where $\mathbf{x}\in\mathbb{R}^N$ is the state vector and $u\in\mathbb{R}$ represents a tunable parameter, which can be considered as a control input. Our task is to reach a desired trajectory $\mathbf{x}^*(t)$ that satisfies Eq. (82) with $u=0$. To achieve that we first construct a surface S, called the Poincaré section, which passes through the point $\mathbf{x}_0=\mathbf{x}^*(0)$ transversally to the trajectory $\mathbf{x}^*(t)$ (see Fig. 30). Consider a map $\mathbf{x}\mapsto\mathbf{F}(\mathbf{x},u)$, where $\mathbf{F}(\mathbf{x},u)$ is the point of first return to the Poincaré section of the solution of Eq. (82) that begins at the point $\mathbf{x}$ and was obtained for the constant input u. Since we can integrate Eq. (82) forward in time from $\mathbf{x}$, the map $\mathbf{x}\mapsto\mathbf{F}(\mathbf{x},u)$, called the Poincaré map, must exist. Note that even though we may not be able to write down the map $\mathbf{F}$ explicitly, the knowledge that it exists is still useful (Shinbrot *et al.*, 1993). By considering a sequence of such maps, we get a discrete system

$$\mathbf{x}_{k+1}=\mathbf{F}(\mathbf{x}_k,u_k), \tag{83}$$

where $\mathbf{x}_k=\mathbf{x}(t_k)$, t_k is the time of the kth intersection of the Poincaré section S, and u_k is the value of control $u(t)$ over the interval between t_k and t_{k+1}.

A key step in the OGY method is to linearize the discrete system (83) as

$$\mathbf{z}_{k+1}=\mathbf{A}\mathbf{z}_k+\mathbf{B}u_k, \tag{84}$$

where $\mathbf{z}_k=\mathbf{x}_k-\mathbf{x}_0$, $\mathbf{A}=\partial\mathbf{F}/\partial\mathbf{x}|_{\mathbf{x}_0}$ is the Jacobian matrix, and $\mathbf{B}=\partial\mathbf{F}/\partial u|_{\mathbf{x}_0}$ is a column vector.

To stabilize the linear system (84) and hence steer the original system to a desired periodic orbit that passes through $\mathbf{x}_0$, the OGY method employs a linear state feedback control law

$$u_k=\begin{cases}\mathbf{C}\mathbf{z}_k & \text{if } |\mathbf{z}_k|\le\delta,\\ 0 & \text{otherwise,}\end{cases} \tag{85}$$

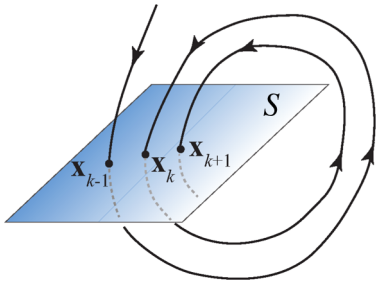

FIG. 30. Poincaré map. In a continuous dynamical system the Poincaré map is the intersection of a periodic orbit in the state space with a certain lower-dimensional subspace, called the Poincaré section S, transversal to the flow of the system. In the Poincaré section S, the Poincaré map $\mathbf{x}\mapsto\mathbf{F}(\mathbf{x},u)$ projects point $\mathbf{x}$ onto point $\mathbf{F}(\mathbf{x},u)$, i.e., $\mathbf{x}_k=\mathbf{F}(\mathbf{x}_{k-1},u_{k-1})$, $\mathbf{x}_{k+1}=\mathbf{F}(\mathbf{x}_k,u_k)$, etc.

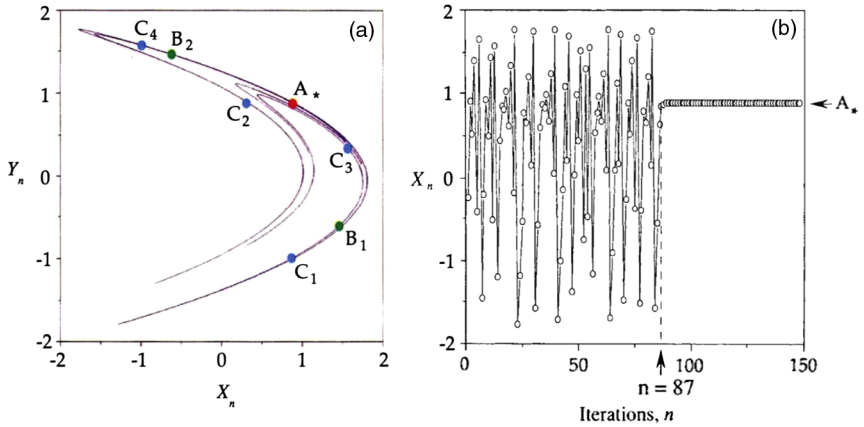

FIG. 31. Controlling chaos. The use of the Ott-Grebogi-Yorke (OGY) method to control the chaotic behavior in the Hénon map $X_{n+1} = p + 0.3Y_n - X_n^2, Y_{n+1} = X_n$, where the parameter p is set to $p_0 = 1.4$. (a) The Hénon attractor contains period-1 point A^*, which is revisited in each map iteration, period-2 points B_1 and B_2, which are revisited every other map iteration, i.e., $B_1 \to B_2 \to B_1 \to B_2 \to \cdots$, and period-4 points C_1, C_2, C_3, and C_4, which are cycled through every four map iterations. (b) The result of stabilizing the periodic orbit A^* of the Hénon attractor by tuning p by less than 1% around p_0. The arrow indicates the time step at which the small perturbation is initiated. For the first 86 iterations, the trajectory moves chaotically on the attractor, never falling within the desired small region about A^*. On the 87th iteration, following the application of the control perturbation, the state falls within the desired region, and is held near A^*. From Shinbrot *et al.*, 1993.

where $\delta > 0$ is a sufficiently small parameter. Note that the control is applied only in some neighborhood of the desired trajectory, which ensures the smallness of the control action. To guarantee the efficiency of the method, the matrix $\mathbf{C}$ must be chosen so that in the linear closed-loop system $\mathbf{z}_{k+1} = (\mathbf{A} + \mathbf{BC})\mathbf{z}_k$, the norm $|(\mathbf{A} + \mathbf{BC})\mathbf{z}| \le \rho|\mathbf{z}|$ decreases, where $\rho < 1$.

Extensive numerical simulations have corroborated the practical utility of the OGY method (see Fig. 31). Furthermore, the OGY method was proven to be effective in experimental systems as well, allowing the stabilization of UPOs in a chaotically oscillating magnetoelastic ribbon, a driven diode circuit, a multimode laser with an intracavity crystal, a thermal convection loop, and the Belousov-Zhabotinsky reaction (Boccaletti *et al.*, 2000). Slow convergence was often reported, which is a price we must pay to achieve global stabilization of a nonlinear system with small control action (Fradkov and Evans, 2005).

The advantage of the OGY method is that it does not require prior knowledge of the system's dynamics. Instead, we just rely on the system's behavior to learn the necessary small perturbation to nudge it toward a desired trajectory. This is similar to the balancing of a stick on our palm, which can be achieved without knowing Newton's second law of motion and the stick's detailed equation of motion. Indeed, in the OGY method, both $\mathbf{A}$ and $\mathbf{B}$ in Eq. (84) can be extracted purely from observations of the trajectory on the chaotic attractor (Shinbrot *et al.*, 1993).

3. Time-delayed feedback: Pyragas method

The Pyragas method employs continuous feedback to synchronize the current state of a system with a time-delayed version of itself, offering an alternative approach to stabilizing a desired UPO embedded in a chaotic attractor (Pyragas, 1992). Consider the nonlinear system (82). If it has a desired UPO $\Gamma = \{\mathbf{x}^*(t)\}$ with period T for $\mathbf{u} = \mathbf{0}$, then we can use the feedback control

$$\mathbf{u}(t) = K[\mathbf{x}(t) - \mathbf{x}(t - \tau)], \tag{86}$$

where K is the feedback gain and τ is the delay time, to stabilize the desired UPO. If $\tau = T$ and the solution $\mathbf{x}(t)$ of the closed-loop system (82) and (86) begins on the UPO, then it remains on the UPO for all $t \ge 0$. Surprisingly, $\mathbf{x}(t)$ can converge to the UPO even if it initially is not on the UPO, i.e., $\mathbf{x}(0) \notin \Gamma$.

Considering that not all the state variables are experimentally accessible, we can rewrite Eq. (86) as

$$u(t) = K[y(t) - y(t - T)] \tag{87}$$

for a desired UPO of period T. Here $y(t) \in \mathbb{R}$ is an experimentally accessible output signal. The advantage of the time-delayed feedback control law (87) is that it does not require rapid switching or sampling, nor does it require a reference signal corresponding to the desired UPO. Unfortunately, the domain of system parameters over which control can be achieved via Eq. (87) is limited. Furthermore, the method fails for highly unstable orbits. Note, however, that an extended variant of the Pyragas method, using a control law whose form is closely related to the amplitude of light reflected from a Fabry-Pérot interferometer, can stabilize highly unstable orbits (Socolar, Sukow, and Gauthier, 1994).

Despite the simple form (86) and (87) of the control signal, the analytical study of the closed-loop system is challenging. Indeed, while there are extensive numerical and experimental results pertaining to the properties and application of the Pyragas method, the sufficient conditions that guarantee its applicability remain unknown (Fradkov and Evans, 2005).

Note that similar to the Pyragas method, a geometric method of stabilizing UPOs (Toroczkai, 1994; Sass and Toroczkai, 1996) also uses time delays. This method does not require explicit knowledge of the dynamics (which is similar to the OGY method), but only experimentally accessible state information within a short period of the system's immediate past. More specifically, it requires a rough location of the UPO and a single parameter easily computed from four data points. This geometric method does not have the problems of the Pyragas method in stabilizing UPOs. The drawback of this geometric method is that it has been formulated only for 2D maps and 3D flows.

B. Compensatory perturbations of state variables

The control tools described previously were mainly designed for low-dimensional dynamical systems with a simple structure. Most complex systems are high dimensional, however, consisting of a network of components connected by nonlinear interactions. We need, therefore, tools to bring a networked system to a desired target state. A recently proposed method can work even when the target state is not directly accessible due to certain constraints (Cornelius, Kath, and Motter, 2013). The basic insight of the approach is that each desirable state has a "basin of attraction," representing a region of initial conditions whose trajectories converge to it. For a system that is in an undesired state, we need to identify perturbations to the state variables that can bring the system to the basin of attraction of the desired target state. Once there, the system will evolve spontaneously to the target state. Assume that a physically admissible perturbation fulfills some constraints that can be represented by vector expressions of the form

$$\mathbf{g}(\mathbf{x}_0, \mathbf{x}_0') \leq \mathbf{0} \quad \text{and} \quad \mathbf{h}(\mathbf{x}_0, \mathbf{x}_0') = \mathbf{0}, \tag{88}$$

where the equality and inequality apply to each component individually. To iteratively identify compensatory perturbations we use the following procedure (see Fig. 32): Given the current initial state of the network $\mathbf{x}_0$, we integrate the system's dynamics over a time window $t_0 \leq t \leq t_0 + T$ to identify the time when the orbit is closest to the target $t_c \equiv \arg\min |\mathbf{x}^* - \mathbf{x}(t)|$. We then integrate the variational equation up to t_c to obtain the corresponding variational matrix $\mathbf{M}(t_c)$, which maps a small change $\delta\mathbf{x}_0$ in the initial state of the network to a change $\delta\mathbf{x}(t_c)$ in the resulting perturbed orbit at t_c according to $\delta\mathbf{x}(t_c) = \mathbf{M}(t_c) \cdot \delta\mathbf{x}_0$. This mapping is used to select an incremental perturbation $\delta\mathbf{x}_0$ that minimizes the distance between the perturbed orbit and the target at time t_c, subject to the constraints (88) and additional constraints on $\delta\mathbf{x}_0$ to ensure the validity of the variational approximation.

The selection of $\delta\mathbf{x}_0$ is performed via a nonlinear optimization that can be efficiently solved using sequential quadratic programming. The initial condition is then updated $\mathbf{x}_0 \to \mathbf{x}_0 + \delta\mathbf{x}_0$, and we test whether the new initial state lies

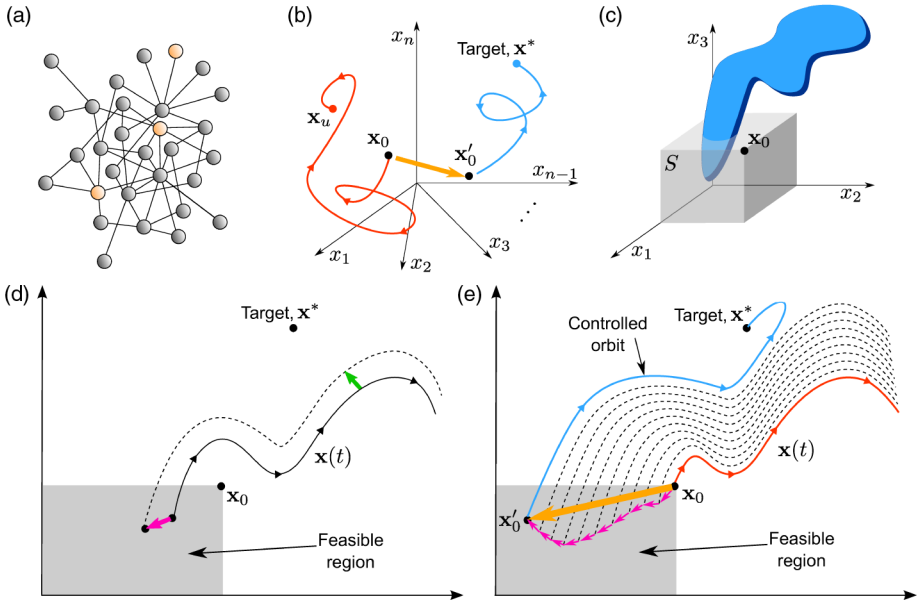

FIG. 32. Steering a network using compensatory perturbations of state variables. (a) The control set (shown in yellow) is a set of nodes that are accessible to compensatory perturbations. (b) In the absence of control, the network is in an initial state $\mathbf{x}_0$ and evolves to an undesirable equilibrium $\mathbf{x}_u$ (red curve). By perturbing the initial state (orange arrow), the network reaches a new state $\mathbf{x}_0'$, which evolves to the desired target state (blue curve). (c) Typically, the compensatory perturbations must obey some constraints. In this example, we can perturb only three out of N dimensions, corresponding to the three-node control set (shown in yellow), and the state variable of each control node can only be reduced. These constraints form a cube (gray volume) within the three-dimensional subspace of the control nodes. The network can be steered to the target state if and only if the corresponding slice of the target's basin of attraction (blue volume) intersects this cube. (d) Along each orbit there is a point that is closest to the target state. We seek to identify a perturbation (magenta arrow) to the initial condition that brings the closest point closer to the target (green arrow). (e) This process is repeated (dashed curves), until we identify a perturbed state $\mathbf{x}_0'$ that is in the attraction basin of the target state, hence the system automatically evolves to the target state. This results in a compensatory perturbation $\mathbf{x}_0 \to \mathbf{x}_0'$ (orange arrow). From Cornelius, Kath, and Motter, 2013.

in the target's basin of attraction by integrating the system dynamics over a long time τ. If the system's orbit reaches a small ball of radius κ around $\mathbf{x}^*$ within this time, we declare success and recognize $\mathbf{x}_0' - \mathbf{x}_0$ as a compensatory perturbation (for the updated $\mathbf{x}_0'$). If not, we calculate the time of closest approach of the new orbit and repeat the procedure.

Similar to the open-loop control of chaos discussed in Sec. V.A.1, the approach based on compensatory perturbation potentially works for any nonlinear system. It has been successfully applied to the mitigation of cascading failures in a power grid and the identification of drug targets in a cancer signaling network (Cornelius, Kath, and Motter, 2013). Yet the approach requires *a priori* knowledge of the detailed model describing the dynamics of the system we want to control, a piece of knowledge we often lack in complex systems. With an imperfect model, a compensatory perturbation may steer the system into a different basin of attraction than the desired one. Studying the dependence of the success rate of this approach on the parameter uncertainty and system noise remains an analytically challenging issue. Moreover, it is unclear how to choose the optimal control set consisting of one or more nodes accessible to compensatory perturbations so that some control objectives, such as the number of control nodes or the amount of control energy, are minimized.

C. Small perturbations to system parameters

The control method described in Sec. V.B perturbs the state variables of a networked system. In analogy with the OGY method (Ott, Grebogi, and Yorke, 1990), we can also control complex networked systems via perturbations to the system parameters (Lai, 2014), an approach complementary to the approaches based on perturbations of the state variables. The key step of this approach is to choose a set of experimentally adjustable parameters and determine whether small perturbations to these parameters can steer the system toward the desired attractor (Lai, 1996, 2014). Depending on the physical constraints the control parameters obey, the directed control path from the undesired attractor to the desired attractor can be either via a direct connection or via intermediate attractors along the control path. If there are no feasible control paths reaching the desired attractor, then we cannot steer the system to that attractor, hence control is not possible.

Considering each attractor as a node, and the control paths as directed edges between them, we can construct an "attractor network," whose properties determine the controllability of the original dynamic network (Lai, 2014). For a given nonlinear system, the attractor network can be constructed as follows. First, we identify all possible attractors and choose a set of system parameters that can be experimentally perturbed. Second, we set the system into a specific attractor a, and determine the set of attractors into which the system can evolve from the original attractor a with a reasonable combination of the adjustable parameters. This effectively draws a link from attractor a to all other attractors reachable by feasible parameter perturbations. Finally, we repeat this procedure for all attractors, obtaining the attractor network (Lai, 2014).

To illustrate the construction of such an attractor network, consider the epigenetic state network (ESN) that describes the phenotypic transitions on the epigenetic landscape of a cell (Fig. 33). In the epigenetic landscape, two neighboring fixed-point attractors, corresponding to stable cell phenotypes, are connected by a minimal-energy path through an unstable transition point (first-order saddle point) (Wang *et al.*, 2011, 2014). The number of fixed points (nodes) and saddle points (edges) grows exponentially with the number of genes (dimensionality). We can rely on a conditional root-finding algorithm (Wang *et al.*, 2014) to construct this ESN. The obtained ESN captures the global architecture of stable cell phenotypes, helping us translate the metaphorical Waddington epigenetic landscape concept (Waddington and Kacser, 1957; Slack, 2002) into a mathematical framework of cell phenotypic transitions.

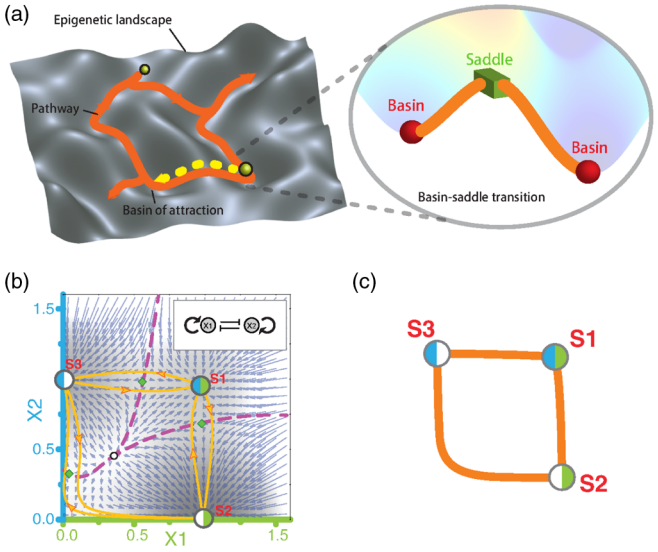

FIG. 33. Epigenetic state network (ESN). (a) On the epigenetic landscape, a minimal-energy path connects two neighboring attractors through an unstable transition point (first-order saddle point). The landscape can be represented by a network, where nodes are attractors or basins of attraction and edges are minimal-energy paths connecting the neighboring attractors. (b) The vector field of a mutually inhibitive two-gene circuit (inset). Nodes S1, S2, and S3 are fixed-point attractors. The pie diagram of each attractor represents the expression pattern of the two genes. The first-order saddle points (green diamond) are surrounded by forward and backward optimal paths (dark blue) connecting two neighboring attractors. (c) The ESN constructed from (a) by connecting neighboring attractors. From Wang *et al.*, 2014.

D. Dynamics and control at feedback vertex sets

For regulatory networks described as a digraph of dependencies, it was recently shown that open-loop control applied to a feedback vertex set (FVS) will force the remaining network to stably follow the desired trajectories (Fiedler *et al.*, 2013; Mochizuki *et al.*, 2013). An FVS is a subset of nodes in the absence of which the digraph becomes acyclic, i.e., contains no directed cycles (Fig. 34). Unlike the approaches discussed in Secs. V.B and V.C, this approach has rigorous analytical underpinnings.

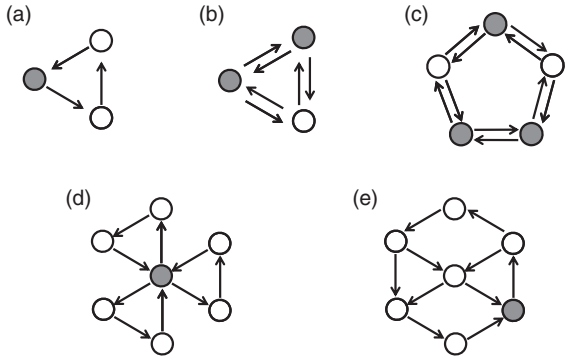

FIG. 34. Feedback vertex set (FVS). Examples of FVSs in directed graphs, whose removal renders the graphs acyclic. The gray vertices represent a choice of a minimal FVS in each panel (a)–(e). Controlling the dynamics of the nodes in an FVS allows us to switch the dynamics of the whole system from one attractor to some other attractor. From Mochizuki *et al.*, 2013.

Consider a general nonautonomous nonlinear networked system

$$\dot{x}_i = F_i(t, x_i, x_{\mathcal{I}_i}), \tag{89}$$

where $i = 1, \ldots, N$, and $\mathcal{I}_i$ denotes the set of upstream neighbors of node i, i.e., $j \in \mathcal{I}_i$ if there is a directed edge $(j \to i)$ in the network. The corresponding network is often called the *system digraph*, which is the transpose of the inference diagram introduced in Sec. IV.B.

An open-loop control applied to the nodes of an FVS will completely control the dynamics of those nodes and hence effectively remove all the incoming links to them. Consequently, those nodes will not be influenced by other nodes. They will, however, continue to influence other nodes and drive the whole system to a desired attractor. Consider, for example, the gene regulatory network of circadian rhythms in mice, consisting of 21 nodes [Fig. 35(a)]. In general there can be multiple minimal FVSs for a given digraph. One such minimal FVS of size seven, i.e., $\mathcal{F} = \{\text{PER1}, \text{PER2}, \text{CRY1}, \text{CRY2}, \text{RORc}, \text{CLK}, \text{BMAL1}\}$, is highlighted by red circles in Fig. 35(a). The associated dynamical system can be described by a set of ODEs involving 21 variables and hundreds of parameters. Under a particular choice of parameters, the system has several invariant sets: (i) two stable periodic oscillations (P1 and P2), (ii) one unstable periodic oscillation (UP), and (iii) one unstable stationary point (USS) [Figs. 35(b) and 35(c)]. Let us aim to steer the system from P1 to P2. To achieve this, we first need to calculate the time tracks of the seven FVS nodes on the desired invariant set P2, denoted as x_i^{P2}, $i \in \mathcal{F}$, which can be done by numerically integrating the ODEs. Then we prescribe the time tracks of the seven nodes in $\mathcal{F}$ to follow their desired values x_i^{P2}. This way, we effectively

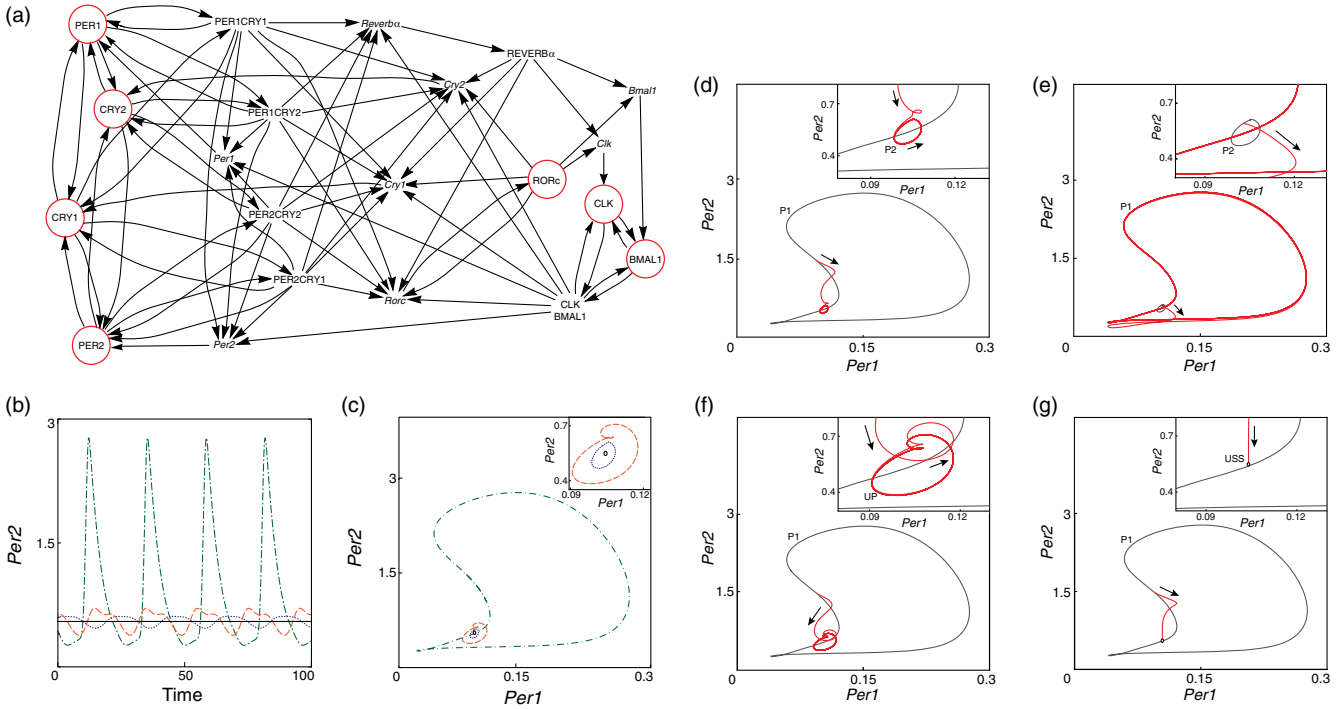

FIG. 35. Controlling a system through its feedback vertex set (FVS). (a) A regulatory network with 21 variables describes the mammalian circadian rhythms in mice (Mirsky *et al.*, 2009). A minimal FVS of seven elements, denoted as $\mathcal{I}$, is highlighted by red circles. (b) Trajectories of two stable periodic orbits, period1 (P1, dotted and broken curves) and period2 (P2, dotted curve), one unstable periodic orbit (UP, broken curve) and one unstable stationary state (USS, solid line), represented by time tracks of the variable Per2. (c) Trajectories of the same solutions in the phase plane of the two variables Per1 and Per2, which are not in the FVS. (d)–(g) Numerical trajectories of successful open-loop controls of circadian rhythms via the full feedback vertex set $\mathcal{I}$. Zooms into P2, UP, and USS are shown as top-right insets. The initial and resulting trajectories of the control experiment are shown in light gray and red (dark gray) curves, respectively. (d) From P1 to P2. (e) From P2 to P1. (f) From P1 to UP. (g) From P1 to USS. From Mochizuki *et al.*, 2013.

remove any influence from the other 14 nodes to the 7 nodes in $\mathcal{F}$. The dynamics of the remaining 14 nodes $x_i, i \notin \mathcal{F}$, are determined by the remaining 14 ODEs of the system, where the initial state of these remaining nodes is chosen to coincide with an arbitrary point on the P1 trajectory. As shown in Fig. 35(d), the trajectories of the remaining 14 nodes deviate from the original stable periodic orbit P1 and quickly converge to the competing orbit P2. The whole system eventually displays periodic oscillation on the P2 orbit. In this example, the identified FVS is a minimal one, i.e., any subset of $\mathcal{F}$ is not an FVS. Yet, a minimal FVS is not guaranteed to be the minimum one that contains the least number of nodes. Naturally, it will be more desirable to identify and control the nodes in the minimum FVS. Unfortunately, finding the minimum FVS of a general digraph is an NP-hard problem (Karp, 1972).

This FVS-based open-loop control can be applied to a wide range of nonlinear dynamical systems. It requires only a few conditions (e.g., continuous, dissipative, and decaying) on the nonlinear functions F_i that are very mild and satisfied by many real systems (Fiedler *et al.*, 2013). For systems associated with a digraph $G(V, E)$, we can rigorously prove that clamping the dynamics of a subset of nodes $S \subseteq V$ will control the rest of the network toward the desired attractor for all choices of nonlinearities F_i that satisfy the above-mentioned conditions if and only if S is an FVS in G (Fiedler *et al.*, 2013). Yet, specific systems do exist (with certain nonlinearity F_i) where clamping a reduced FVS (i.e., removing one or more nodes from an FVS) is sufficient to control the system to a desired attractor. In other words, for a specific system, clamping an FVS might not be necessary. It would be a natural starting point, though.

Note that to apply the two approaches discussed in the previous Secs. V.B and V.C, namely, compensatory perturbations of state variables (Sec. V.B), and small perturbations of system parameters (Sec. V.C), we need a detailed knowledge of the system dynamics, including all system parameters. In many cases, we lack such a piece of knowledge. In contrast, to apply the FVS-based open-loop control (Sec. V.D), we just need the trajectories of FVS nodes on the desired attractors. We do not have to know full dynamics, nor the exact parameter values. We just need to assure a few mild conditions on the nonlinear functions F_i are satisfied.

VI. CONTROLLING COLLECTIVE BEHAVIOR

Dynamical agents interacting through complex networks can display a wide range of collective behavior, from synchronization to flocking among many interacting agents. In particular, the study of network-mediated synchronization has a long history, with applications from biology to neuroscience, engineering, computer science, economy, and social sciences (Arenas *et al.*, 2008). Flocking has also gained significant attention in the past two decades, capturing phenomena from the coordinated motion of birds or fish to self-organized networks of mobile agents. Applications range from massive distributed sensing using mobile sensor networks to the self-assembly of connected mobile networks, and military missions such as reconnaissance, surveillance, and combat using cooperative unmanned aerial vehicles (Olfati-Saber, Fax, and Murray, 2007). These problems pose, however, a number of fundamental questions pertaining to the control of self-organized networks.

If we aim to achieve a desired collective behavior, it is often infeasible to directly control all nodes of a large network. This difficulty is partially alleviated by the notion of pinning control (Wang and Chen, 2002a, 2002b), which relies heavily on feedback processes. In pinning control a feedback control input is applied to a small subset of nodes called pinned nodes, which propagates to the rest of the network through the edges. The design and implementation of feedback control must take into account both the individual dynamics of the components and the network topology. Conceptually, pinning control is similar to the minimum controllability problem of a linear system discussed in Sec. II. The key difference is that, instead of fully controlling a system, pinning control aims to control only the system's collective behavior, such as synchronization or flocking. Pinning control has been extensively applied to the synchronization of coupled oscillators and flocking of interacting agents (Wang and Chen, 2002a, 2002b; Li, Wang, and Chen, 2004; Sorrentino *et al.*, 2007; Chen and Duan, 2008; Porfiri and di Bernardo, 2008; Zou and Chen, 2008; Yu, Chen, and Lü, 2009; Yu *et al.*, 2013; Bernardo and DeLellis, 2014).

In this section we review some fundamental results on controlling the collective behavior of complex networked systems. We pay particular attention to the pinning control of synchronization and flocking. Synchronization of coupled oscillators is typically studied on fixed network topology. We build on the master stability formalism to explore pinning synchronization, focusing on local and global stability conditions and adaptive strategies. Flocking of multiagent systems are typically associated with switching or time-varying network topology, because the agents, such as robots, vehicles, or animals, are often mobile. To illustrate this we discuss the Vicsek model of flocking behavior, emphasizing its control theoretical interpretation. Finally, we review key protocols that can induce flocking in multiagent systems.

A. Synchronization of coupled oscillators

Consider a static network of N identical nodes (oscillators) with nearest-neighbor coupling:

$$\dot{\mathbf{x}}_i = \mathbf{f}(\mathbf{x}_i) + \sigma \sum_{j=1}^{N} a_{ij} w_{ij} [\mathbf{h}(\mathbf{x}_j) - \mathbf{h}(\mathbf{x}_i)]$$
$$= \mathbf{f}(\mathbf{x}_i) - \sigma \sum_{j=1}^{N} g_{ij} \mathbf{h}(\mathbf{x}_j), \tag{90}$$

where $\mathbf{x}_i \in \mathbb{R}^d$ is the d-dimensional state vector of the ith node, $\mathbf{f}(\mathbf{x}_i)$: $\mathbb{R}^d \to \mathbb{R}^d$ determines the individual dynamics of each node, σ is the coupling strength, also called the *coupling gain*, $\mathbf{A} = (a_{ij})$ is the $N \times N$ adjacency matrix of the network, and $w_{ij} \geq 0$ is the weight of link (i, j). The output function $\mathbf{h}(\mathbf{x})$: $\mathbb{R}^d \to \mathbb{R}^d$ is used to couple the oscillators and is identical for all oscillators. For example, if we use $\mathbf{h}(\mathbf{x}) = (x, 0, 0)^{\mathrm{T}}$ for a three-dimensional oscillator, such as the Lorenz or Rössler oscillator, it means that the oscillators are coupled

only through their x components. In general, $\mathbf{h}(\mathbf{x})$ can be any linear or nonlinear mapping of the state vector $\mathbf{x}$. $\mathbf{G} = (g_{ij})$ is the $N \times N$ coupling matrix of the network ($g_{ij} = -a_{ij}w_{ij}$ for $i \neq j$ and $g_{ii} = -\sum_{j=1, j\neq i}^{N} g_{ij}$). If $w_{ij} = 1$ for all links, $\mathbf{G}$ is the Laplacian matrix $\mathbf{L}$ of the network. Note that $\mathbf{G}$ is not necessarily symmetric.

The system (90) is synchronized when the trajectories of all nodes converge to a common trajectory, i.e.,

$$\lim_{t\to\infty} \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\| = 0 \tag{91}$$

for all $i, j = 1, \ldots, N$. Such synchronization behavior describes a continuous system that has a uniform movement, used to model synchronized neurons, lasers, and electronic circuits (Pecora and Carroll, 1998).

Because of the diffusive coupling, the completely synchronized state $\mathbf{x}_1(t) = \mathbf{x}_2(t) = \cdots = \mathbf{x}_N(t) = \mathbf{s}(t)$ is a natural solution of Eq. (90). This also defines a linear invariant manifold, called the synchronization manifold, where all the oscillators evolve synchronously as $\dot{\mathbf{s}} = \mathbf{f}(\mathbf{s})$. Note that $\mathbf{s}(t)$ may be an equilibrium point, a periodic orbit, or even a chaotic solution.

Despite the fact that the completely synchronized state is a natural solution of Eq. (90), it may not emerge spontaneously. For example, if the coupling gain σ is close to zero, the oscillators tend to behave independently. If the coupling gain σ is too strong, the oscillators may not synchronize either. Our goal is to identify the conditions under which the system (90) can synchronize. A broad spectrum of methods allows us to address this question (Wu and Chua, 1994; Pecora and Carroll, 1998; Barahona and Pecora, 2002; Belykh, Belykh, and Hasler, 2004b; Chen, 2007; Russo and Di Bernardo, 2009). The best-known method, discussed next, is based on the calculation of the eigenvalues of the coupling matrix.

1. Master stability formalism and beyond

Consider the stability of the synchronization manifold in the presence of a small perturbation $\mathbf{x}_i(t) = \mathbf{s}(t) + \delta\mathbf{x}_i(t)$. By expanding $\mathbf{f}(\mathbf{x}_i)$ and $\mathbf{h}(\mathbf{x}_i)$ to the first order of $\delta\mathbf{x}_i$, we obtain a linear variational equation for $\delta\mathbf{x}_i(t)$,

$$\delta\dot{\mathbf{x}}_i = \mathcal{J}(\mathbf{s})\delta\mathbf{x}_i - \sigma\sum_{j=1}^{N} g_{ij}\mathcal{E}(\mathbf{s})\delta\mathbf{x}_j, \tag{92}$$

with Jacobian matrices $\mathcal{J}(\mathbf{s}) = \partial\mathbf{f}(\mathbf{x})/\partial\mathbf{x}|_{\mathbf{x}=\mathbf{s}}$ and $\mathcal{E}(\mathbf{s}) = \partial\mathbf{h}(\mathbf{x})/\partial\mathbf{x}|_{\mathbf{x}=\mathbf{s}}$. Let $\delta\mathbf{X} \equiv [\delta\mathbf{x}_1, \ldots, \delta\mathbf{x}_N]^{\mathrm{T}}$. Then formally we have

$$\delta\dot{\mathbf{X}} = [\mathbf{I} \otimes \mathcal{J}(\mathbf{s}) - \sigma\mathbf{G} \otimes \mathcal{E}(\mathbf{s})]\delta\mathbf{X}, \tag{93}$$

where $\mathbf{I}$ is the $N \times N$ identity matrix and $\otimes$ is the Kronecker product (a.k.a. matrix direct product).

The key idea of the master stability formalism is that we need to consider only variations that are transverse to the synchronization manifold, as variations along $\mathbf{s}(t)$ leave the system in the synchronized state (Pecora and Carroll, 1998; Barahona and Pecora, 2002). If these transverse variations damp out, then the synchronization manifold is stable. To separate out the transverse variations, we can project $\delta\mathbf{X}$ into the eigenspace spanned by the eigenvectors $\mathbf{e}_i$ of the coupling matrix $\mathbf{G}$, i.e., $\delta\mathbf{X} = (\mathbf{P} \otimes \mathbf{I}_{\mathrm{d}})\Xi$ with $\mathbf{P}^{-1}\mathbf{G}\mathbf{P} = \hat{\mathbf{G}} = \mathrm{Diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)$. Then we have

$$\dot{\Xi} = [\mathbf{I} \otimes \mathcal{J}(\mathbf{s}) - \sigma\hat{\mathbf{G}} \otimes \mathcal{E}(\mathbf{s})]\Xi, \tag{94}$$

which results in a block diagonalized variational equation with N blocks, corresponding to N decoupled eigenmodes. Each block has the form

$$\dot{\boldsymbol{\xi}}_i = [\mathcal{J}(\mathbf{s}) - \sigma\lambda_i\mathcal{E}(\mathbf{s})]\boldsymbol{\xi}_i, \tag{95}$$

where $\boldsymbol{\xi}_i$ is the eigenmode associated with the eigenvalue λ_i of $\mathbf{G}$. Note that in deriving Eq. (94) we implicitly assumed that the coupling matrix $\mathbf{G}$ is diagonalizable, which is always true for symmetric $\mathbf{G}$. Thus each eigenmode of the perturbation is decoupled from the others and will damp out independently and simultaneously. If $\mathbf{G}$ is not diagonalizable, we can transform $\mathbf{G}$ into the Jordan canonical form. In this case, some eigenmodes of the perturbation may suffer from a long transient (Nishikawa and Motter, 2006).

We can order the eigenvalues of $\mathbf{G}$ such that $0 = \lambda_1 \leq \mathrm{Re}\lambda_2 \leq \cdots \leq \mathrm{Re}\lambda_N$. Because the row sum of $\mathbf{G}$ is zero, the minimal eigenvalue λ_1 is always zero with the corresponding eigenvector $\mathbf{e}_1 = (1, 1, \ldots, 1)^{\mathrm{T}}$. Hence the first eigenmode $\dot{\boldsymbol{\xi}}_1 = \mathcal{J}(\mathbf{s})\boldsymbol{\xi}_1$ corresponds to the perturbation parallel to the synchronization manifold. According to the Gerschgorin circle theorem, all other eigenvalues must have non-negative real parts. The corresponding $(N-1)$ eigenmodes are transverse to the synchronization manifold and must decay to have a stable synchronization manifold.

The form of each block in Eq. (95) is the same up to the scalar multiplier $\sigma\lambda_i$. This leads to the variational equation, called the master stability equation,

$$\dot{\boldsymbol{\xi}} = [\mathcal{J} - (\alpha + i\beta)\mathcal{E}]\boldsymbol{\xi}. \tag{96}$$

For small $\boldsymbol{\xi}$ we have $\|\boldsymbol{\xi}(t)\| \sim \exp[\Lambda(\alpha, \beta)t]$, which decays exponentially if the maximum Lyapunov characteristic exponent $\Lambda(\alpha, \beta) < 0$. Consequently, $\Lambda(\alpha, \beta)$ is called the master stability function (MSF). Given a coupling strength σ, the sign of the MSF in the point $\sigma\lambda_i$ in the complex plane reveals the stability of that eigenmode. If all eigenmodes are stable [i.e., $\Lambda(\sigma\lambda_i) < 0$ for all i's], then the synchronization manifold is stable at that coupling strength. Note that since the master stability formalism assesses only the linear stability of the synchronized state, it yields only the necessary but not the sufficient condition for synchronization.

For undirected and unweighted networks, the coupling matrix $\mathbf{G}$ is symmetric and all its eigenvalues are real, simplifying the stability analysis. In this case, depending on $\mathcal{J}$ and $\mathcal{E}$, the MSF $\Lambda(\alpha)$ can be classified as follows:

(i) Bounded: $\Lambda(\alpha) < 0$ for $\alpha_1 < \alpha < \alpha_2$. This usually happens when $\mathbf{h}(\mathbf{x}) \neq \mathbf{x}$. The linear stability of the synchronized manifold requires that $\alpha_1 < \sigma\lambda_2 \leq \cdots \leq \sigma\lambda_N < \alpha_2$. This condition can be fulfilled only for σ when the eigenratio R satisfies

$$R \equiv \frac{\lambda_N}{\lambda_2} < \frac{\alpha_2}{\alpha_1}. \tag{97}$$

The beauty of this inequality comes from the fact that its rhs depends only on the dynamics while its left-hand side (lhs) depends only on the network structure. If $R > \alpha_2/\alpha_1$, for any σ the synchronization manifold is unstable, indicating that it is impossible to synchronize the network. If $R < \alpha_2/\alpha_1$, the synchronization manifold is stable for $\sigma_{\min} = \alpha_1/\lambda_2 < \sigma < \sigma_{\max} = \alpha_2/\lambda_N$. The synchronizability of the network can be quantified by the relative interval $\sigma_{\max}/\sigma_{\min} = \alpha_2/(\alpha_1 R)$. A network is more synchronizable for higher $\sigma_{\max}/\sigma_{\min}$ (or smaller R).

(ii) Unbounded: $\Lambda(\alpha) < 0$ for $\alpha > \alpha_1$. The stability criteria of the synchronized manifold is $\alpha_1 < \sigma\lambda_2 \leq \cdots \leq \sigma\lambda_N$, which is true if

$$\sigma > \sigma_{\min} = \alpha_1/\lambda_2. \tag{98}$$

The larger λ_2 is the smaller the synchronization threshold $\sigma_{\min}$ is, hence the more synchronizable is the network.

Inequalities (97) and (98) demonstrate that the MSF framework provides an objective criteria (R or λ_2) to assess the synchronizability of complex networks based on the spectrum of the coupling matrix $\mathbf{G}$ only, without referring to specific oscillators and output functions. The MSF framework allows us to address the impact of the network topology and edge weights on synchronizability (Arenas *et al.*, 2008). Consequently, there have been numerous numerical attempts to relate the spectral properties of network models to a single structural characteristic of the underlying network, such as mean degree, degree heterogeneity, path lengths, clustering coefficient, degree-degree correlations, etc. (Arenas *et al.*, 2008). The outcome of these analyses is occasionally confusing, because in a networked environment it is usually impossible to isolate a single structural characteristic while keeping the others fixed. Overall, several network characteristics can influence synchronizability, but none of them is an exclusive factor in the observed dependencies.

The fundamental limitation of MSF is that it assesses only the linear or local stability of the synchronized state, which is a necessary but not a sufficient condition for synchronization (Arenas *et al.*, 2008). To obtain a sufficient condition, one can use global stability analysis, such as Lyapunov's direct method (Wu and Chua, 1994, 1995a, 1995b, 1995c; Belykh, Belykh, and Hasler, 2004a, 2004b, 2006; Belykh *et al.*, 2005; Chen, 2006, 2007, 2008; Li *et al.*, 2009) or contraction theory (Lohmiller and Slotine, 1998; Wang and Slotine, 2005; Li, Small, and Fu, 2007; Pham and Slotine, 2007; Russo and Di Bernardo, 2009; Tabareau, Slotine, and Pham, 2010; Aminzare and Sontag, 2014).

2. Pinning synchronizability

If a network of coupled oscillators cannot synchronize spontaneously, we can design controllers that, applied to a subset of pinned nodes $\mathcal{C}$, help synchronize the network. Hence the pinned nodes behave like *leaders* (Wang and Chen, 2002a; Li, Wang, and Chen, 2004; Wang and Slotine, 2005, 2006), forcing the remaining *follower* nodes to synchronize. This procedure, known as pinning synchronization, is fundamentally different from spontaneous synchronization of coupled oscillators, where we do not specify the synchronized trajectory $\mathbf{s}(t)$, hence the system "self-organizes" into the synchronized trajectory under appropriate conditions. In pinning synchronization, we choose the desired trajectory $\mathbf{s}(t)$, aiming to achieve some desired control objective, and this trajectory must be explicitly taken into account in the feedback controller design. Note that in the literature pinning synchronizability is often called pinning controllability. Here we use the term synchronizability to avoid confusion with the classical notion of controllability discussed in Secs. II and III.

A controlled network is described by

$$\dot{\mathbf{x}}_i = \mathbf{f}(\mathbf{x}_i) - \sigma \sum_{j=1}^{N} g_{ij}\mathbf{h}(\mathbf{x}_j) + \delta_i \mathbf{u}_i(t), \tag{99}$$

where $\delta_i = 1$ for pinned nodes and 0 otherwise, and

$$\mathbf{u}_i(t) = \sigma[\mathbf{p}_i(\mathbf{s}(t)) - \mathbf{p}_i(\mathbf{x}_i(t))] \tag{100}$$

is the d-dimensional linear feedback controller (Wang and Chen, 2002a; Li, Wang, and Chen, 2004), $\mathbf{p}_i(\mathbf{x}(t))$ is the pinning function that controls the input of node i, and $\mathbf{s}(t)$ is the desired synchronization trajectory satisfying $\dot{\mathbf{s}}(t) = \mathbf{f}(\mathbf{s}(t))$. Note that in the fully synchronized state $\mathbf{x}_1(t) = \mathbf{x}_2(t) = \cdots = \mathbf{x}_N(t) = \mathbf{s}(t)$, we have $\mathbf{u}_i(t) = \mathbf{0}$ for all nodes. The form of the linear feedback controller (100) implies that the completely synchronized state is a natural solution of the controlled network (99).

Similar to spontaneous synchronization, we must derive the necessary and sufficient conditions for pinning synchronization. These conditions are more important from the control perspective, because they are the prerequisite for the design of any practical controller. If we focus on the local (or global) stability of the synchronized manifold of the controlled network (99), we obtain the necessary (or sufficient) condition for pinning synchronization, describing the local (or global) pinning synchronizability.

a. Local pinning synchronizability

Given the presence of inhomogeneous dynamics at the controlled and uncontrolled nodes, the MSF approach cannot be directly applied to the controlled network (99). Instead, we first introduce a virtual node whose dynamics follows $\dot{\mathbf{s}}(t) = \mathbf{f}(\mathbf{s}(t))$, representing the desired synchronization solution (Sorrentino *et al.*, 2007; Zou and Chen, 2008). The extended system now has $N+1$ nodes: $\mathbf{y}_i(t) = \mathbf{x}_i(t)$ for $i = 1, \ldots, N$, and $\mathbf{y}_{N+1}(t) = \mathbf{s}(t)$. The virtual node is connected to each pinned node.

We choose the pinning function

$$\mathbf{p}_i(\mathbf{x}) = \kappa_i \mathbf{h}(\mathbf{x}) \tag{101}$$

with control gains. By defining the pinning function via Eq. (101) we can then rewrite Eq. (99) in the form of Eq. (90), with an effective coupling matrix satisfying the

zero row-sum condition, allowing us to apply the MSF approach. Indeed, plugging Eq. (101) into Eq. (100), we have $\mathbf{u}_i(t) = \sigma\kappa_i[\mathbf{h}(\mathbf{s}(t)) - \mathbf{h}(\mathbf{x}_i(t))]$ and Eq. (99) becomes

$$\dot{\mathbf{y}}_i = \mathbf{f}(\mathbf{y}_i) - \sigma \sum_{j=1}^{N+1} m_{ij}\mathbf{h}(\mathbf{y}_j), \tag{102}$$

where

$$\mathbf{M} = \begin{bmatrix} g_{11}+\delta_1\kappa_1 & g_{12} & \cdots & g_{1N} & -\delta_1\kappa_1 \\ g_{21} & g_{22}+\delta_2\kappa_2 & \cdots & g_{3N} & -\delta_2\kappa_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ g_{N1} & g_{N2} & \cdots & g_{NN}+\delta_N\kappa_N & -\delta_N\kappa_N \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \tag{103}$$

is the effective coupling matrix of the $(N+1)$-dimensional extended system. Apparently, $\mathbf{M} = (m_{ij})_{(N+1)\times(N+1)}$ is a zero row-sum matrix; hence we can sort its eigenvalues as $0 = \lambda_1 \le \mathrm{Re}\lambda_2 \le \cdots \le \mathrm{Re}\lambda_{N+1}$. We can now apply the MSF approach to numerically explore the local stability of the synchronization manifold of the controlled network (102).

The role of the control gain (κ_i), coupling gain (σ), and the number and locations of the pinned nodes on local pinning synchronizability has been systematically studied (Sorrentino *et al.*, 2007). Consider, for example, a Barabási-Albert (BA) scale-free network of N identical Rössler oscillators coupled in the x and z directions. By assuming $\kappa_1 = \cdots = \kappa_N = \kappa$, it was found that for a wide range of coupling gain σ, the eigenratio $R^{N+1} \equiv \mathrm{Re}\lambda_{N+1}/\mathrm{Re}\lambda_2$ of the new coupling matrix $\mathbf{M}$ is minimized and hence the local pinning synchronizability is maximized around a specific σ-dependent value of the control gain κ. In other words, too large or too small control gain can reduce the network pinning synchronizability [Figs. 36(a) and 36(b)]. By contrast, the number of pinned nodes, regardless if they are chosen randomly or selectively within the network, has a monotonic impact on pinning synchronizability: Controlling more nodes always enhances the network pinning synchronizability, in line with our intuition [Figs. 36(c) and 36(d)]. Furthermore, selective pinning, when the nodes are chosen in the order of decreasing degree, yields better synchronizability than random pinning.

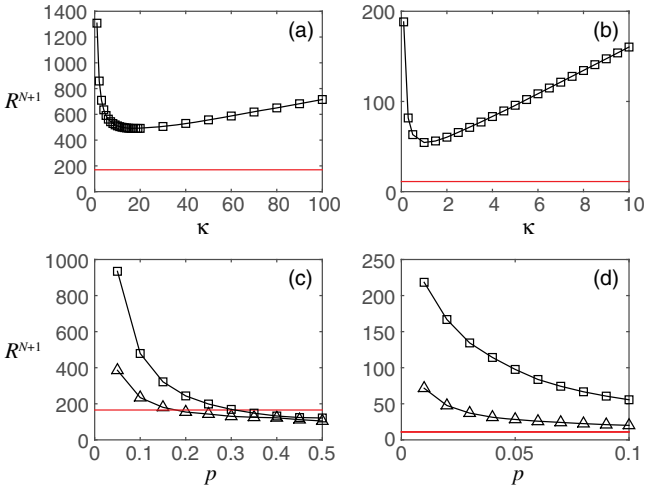

FIG. 36. Local pinning synchronizability of scale-free networks. The local pinning synchronizability is quantified by the eigenratio $R^{N+1} = \mathrm{Re}\lambda_{N+1}/\mathrm{Re}\lambda_2$ of the extended system (102). The calculation was performed for $N = 10^3$ identical Rössler oscillators coupled in the x and z directions, with coupling gain σ and a p fraction of pinned nodes, placed on a Barabási-Albert (BA) scale-free network with mean degree $\langle k\rangle = 4$. We choose $p = 0.1$ fraction of nodes to pin and study the impact of control gain κ on local pinning synchronizability with coupling gain (a) $\sigma = 0.3$ and (b) 2.8, respectively. We find that in both cases the eigenratio $R^{N+1} \equiv \mathrm{Re}\lambda_{N+1}/\mathrm{Re}\lambda_2$ of the new coupling matrix $\mathbf{M}$ is minimized and hence the local pinning synchronizability is maximized around a specific σ-dependent value of the control gain κ. We study the impact of the fraction of pinned nodes on local pinning synchronizability: (c) $\sigma = 0.3$, $\kappa = 10$; (d) $\sigma = 2.8$, $\kappa = 1.5$. The horizontal continuous lines (red) represent the eigenratio R^N of the corresponding uncontrolled system (99). We find that the number of pinned nodes, regardless if they are chosen randomly or selectively within the network, has a monotonic impact on the pinning synchronizability. Controlling more nodes always enhances the network pinning synchronizability. In all plots squares represent the case of random pinning, i.e., a p fraction of nodes is randomly chosen to be pinned. In (c) and (d), triangles represent the case of selective pinning, where nodes have been sorted in the order of decreasing degree and the top p fraction of the nodes are chosen to be pinned. From Sorrentino *et al.*, 2007.

b. Global pinning synchronizability

By describing the time evolution of the controlled network (102) in terms of the error dynamics, we can map the global pinning synchronizability of Eq. (102) to the global asymptotic stability of the synchronized manifold, which can be studied via Lyapunov stability theory.

If the desired asymptotic trajectory is an equilibrium point [$\dot{\mathbf{s}} = \mathbf{f}(\mathbf{s}) = \mathbf{0}$], we can derive sufficient conditions for globally stabilizing the pinning controlled network (Li, Wang, and Chen, 2004). For a more general desired trajectory, it has been shown that a single feedback controller can pin a complex network to a homogenous solution, without assuming symmetry, irreducibility, or linearity of the couplings (Chen, Liu, and Lu, 2007).

If the oscillator dynamics $\mathbf{f}(\mathbf{x})$ fulfills

$$\mathbf{f}(\mathbf{z}_1) - \mathbf{f}(\mathbf{z}_2) = \mathcal{F}_{\mathbf{z}_1,\mathbf{z}_2}(\mathbf{z}_1 - \mathbf{z}_2), \quad \forall\ \mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^d, \tag{104}$$

where $\mathcal{F}_{\mathbf{z}_1,\mathbf{z}_2} \in \mathbb{R}^{d\times d}$ is bounded, i.e., there exists a positive constant α such that for any $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^d$, $\|\mathcal{F}_{\mathbf{z}_1,\mathbf{z}_2}\| \le \alpha$, then we can derive tractable sufficient conditions for global pinning synchronizability in terms of the network topology, the oscillator dynamics, and the linear state feedback (Porfiri and di Bernardo, 2008). Note that the condition (104) applies to a large variety of chaotic oscillators (Jiang, Tang, and Chen,

2003). The results indicate that for a connected network, even for a limited number of pinned nodes, global pinning synchronizability can be achieved by properly selecting the coupling strength and the feedback gain (Chen, Liu, and Lu, 2007).

If $\mathbf{h}(\mathbf{x}) = \mathbf{\Gamma}\mathbf{x}$ and the oscillator dynamics $\mathbf{f}(\mathbf{x})$ satisfies

$$(\mathbf{x}-\mathbf{y})^{\mathrm{T}}[\mathbf{f}(\mathbf{x},t)-\mathbf{f}(\mathbf{y},t)] \le (\mathbf{x}-\mathbf{y})^{\mathrm{T}}\mathbf{K}\mathbf{\Gamma}(\mathbf{x}-\mathbf{y}) \quad (105)$$

for a constant matrix $\mathbf{K}$, sufficient conditions for global pinning synchronizability can also be derived (Yu, Chen, and Lü, 2009; Song and Cao, 2010; Yu *et al.*, 2013). Note that the condition (105) is so mild that many systems from the Lorenz system to the Chen system, the Lü system, recurrent neural networks, or Chua's circuit may satisfy this condition (Yu *et al.*, 2013). Counterintuitively, it was found that for undirected networks the small-degree nodes, instead of hubs, should be pinned first when the coupling strength σ is small (Yu, Chen, and Lü, 2009). For directed networks, nodes with very small in-degree or large out-degree should be pinned first (Yu *et al.*, 2013). This result can be understood by realizing that low in-degree nodes receive less information from other nodes and hence are less "influenced" by others. In the extreme case, nodes with zero in-degree will not be influenced by any other nodes, hence they must be pinned first. On the other hand, large out-degree nodes can influence many other nodes, hence it makes sense to pin them first.

3. Adaptive pinning control

Implementing the linear feedback pinning controller (100) requires detailed knowledge of the global network topology. This is because we have to check whether there are possible coupling and control gains that ensure pinning synchronizability. Yet, in practice we do not always have access to the global network topology. Given this limitation, recently adaptive control has been proposed for pinning synchronization, in which case a controller adapts to a controlled system with parameters that vary in time, or are initially uncertain, without requiring a detailed knowledge of the global network topology (Wang and Slotine, 2006; Wang *et al.*, 2008, 2010; Zhou, Lu, and Lü, 2008; DeLellis, di Bernardo, and Turci, 2010; DeLellis, di Bernardo, and Russo, 2011). As discussed next, many different strategies have been designed to tailor the control gains, coupling gains, or to rewire the network topology to ensure pinning synchronizability.

(i) Adaptation of control gains: To adapt the control gain κ_i in Eq. (101), representing the ratio between the pinning function and output function, we choose the control input $\mathbf{u}_i(t) = -\delta_i\kappa_i(t)[\mathbf{x}_i(t)-\mathbf{s}]$, and the control gains as (Wang *et al.*, 2008; Zhou, Lu, and Lü, 2008)

$$\dot{\kappa}_i(t) = q_i|\mathbf{e}_i(t)|. \quad (106)$$

In other words, the control gain κ_i varies in time and adapts to the error vector $\mathbf{e}_i(t) \equiv \mathbf{s}(t)-\mathbf{x}_i(t)$ that describes the deviation of the oscillator i from the reference signal $\mathbf{s}(t)$. If the individual dynamics $\mathbf{f}(\mathbf{x})$ satisfies the Lipschitz condition, then the global stability of this adaptive strategy can be assured.

(ii) Adaptation of coupling gains: The coupling gain σ_{ij}, defining the mutual coupling strength between node pair (i,j), can also be adapted using (DeLellis, di Bernardo, and Turci, 2010)

$$\dot{\sigma}_{ij}(t) = \eta_{ij}|\mathbf{e}_i(t)-\mathbf{e}_j(t)|^2. \quad (107)$$

This strategy is very effective in controlling networks of quadratic dynamical systems, where the dynamics $\mathbf{f}(\mathbf{x},t)$ of each oscillator satisfies $(\mathbf{x}-\mathbf{y})^{\mathrm{T}}[\mathbf{f}(\mathbf{x},t)-\mathbf{f}(\mathbf{y},t)]-(\mathbf{x}-\mathbf{y})^{\mathrm{T}}\Delta(\mathbf{x}-\mathbf{y}) \le -\omega(\mathbf{x}-\mathbf{y})^{\mathrm{T}}(\mathbf{x}-\mathbf{y})$. Here Δ is a $d\times d$ diagonal matrix and ω is a real positive scalar.

Note that the adaptive strategies (106) and (107) are based on the local error vectors of nodes or between neighboring nodes, hence they avoid the need for a prior tuning of the control or coupling gains. This is attractive in many circumstances. However, these adaptive strategies still require a prior selection of the pinned nodes based on some knowledge of the network topology. This limitation can be avoided by choosing pinned nodes in an adaptive fashion, as discussed next.

(iii) Adaptive selection of pinning nodes: Adaptive pinning can be achieved by assuming the pinning node indicator δ_i to be neither fixed nor binary. A common approach is to introduce

$$\delta_i(t) = b_i^2(t), \quad (108)$$

where $b_i(t)$ satisfies the dynamics

$$\ddot{b}_i + \zeta\dot{b}_i + \frac{dU(b_i)}{db_i} = g(|\mathbf{e}_i|). \quad (109)$$

In other words, $b_i(t)$ follows the dynamics of a unitary mass in a potential $U(b_i)$ subject to an external force g that is a function of the pinning error $\mathbf{e}_i$ and a linear damping term described by $\zeta\dot{b}_i$. This is termed as the edge-snapping mechanism. For convenience, $U(\cdot)$ can be chosen as a double-well potential $U(z) = kz^2(z-1)^2$, where the parameter k defines the height of the barrier between the two wells. Then Eq. (109) has only two stable equilibria, 0 and 1, describing whether node i is pinned or not, respectively. Sufficient conditions for the edge-snapping mechanism (109) to drive the network to a steady-state pinning configuration have been derived (DeLellis, di Bernardo, and Russo, 2011). The key advantage of the adaptive selection of pinning nodes is that we do not have to choose the nodes we need to pin before we design the controller. Instead, we can select them as we go in an adaptive fashion.

(iv) Adaptation of the network topology: We can ensure synchronization by adapting the network topology.

Specially, we can set each off-diagonal element of the Laplacian matrix of the network as

$$\mathcal{L}_{ij}(t) = -\sigma_{ij}(t)\alpha_{ij}^2(t), \tag{110}$$

where $\sigma_{ij}(t)$ is the mutual coupling strength between node pair (i,j), which is adapted as in Eq. (107). The weight $\alpha_{ij}(t)$ is associated with every undirected edge of the target pinning edge and is adapted as

$$\dot{\alpha}_{ij} + \nu\dot{\alpha}_{ij} + \frac{dU(\alpha_{ij})}{d\alpha_{ij}} = c(|\mathbf{e}_{ij}|),$$
$$i,j = 1,\ldots,N, \quad i \neq j, \tag{111}$$

where $\mathbf{e}_{ij}(t) = \mathbf{e}_j(t) - \mathbf{e}_i(t)$, and $U(\cdot)$ can be again chosen as a double-well potential so that Eq. (111) has only two stable equilibria, 0 and 1. In this case, the target network topology evolves in a decentralized way. The local mismatch of the trajectories can be considered as an external forcing on the edge dynamics (111), inducing the activation of the corresponding link, i.e., $\alpha_{ij} = 1$.

These adaptive strategies cope better when pinning controllability using a nonadaptive or static approach is initially not feasible. They are also successful in ensuring network synchronization in the presence of perturbations or deterioration, such as link failures (Jin, Yang, and Che, 2012).

Taken together, we have multiple strategies to force a networked system to synchronize. The discussed tools have a wide range of applications for systems in which a synchronized state is desired. In some cases synchronization can be harmful, as in the case of synchronized clients or routers that cause congestion in data traffic on the Internet (Li and Chen, 2003), or in schizophrenia. In this case the synchronized state can be destroyed by the addition of a single link with inhibitory coupling (Slotine, Wang, and Rifai, 2004).

B. Flocking of multiagent dynamical systems

The flocking of birds, shoaling of fish, swarming of insects, and herding of land animals are spectacular manifestations of a coordinated collective behavior of multiagent systems. These phenomena have fascinated scientists from diverse disciplines, from ecologists to physicists, social, and computer scientists (Olfati-Saber, 2006; Vicsek and Zafeiris, 2012). Many models have been proposed to reproduce the behavior of such self-organized systems. The first widely known flocking simulation was primarily motivated by the visual appearance of a few dozen coherently flying objects, e.g., imaginary birds and spaceships (Reynolds, 1987). Yet the quantitative interpretation of the emerging behavior of huge flocks in the presence of perturbations was possible only following the development of a statistical physics-based interpretation of flocking obtained through the Vicsek model (Vicsek *et al.*, 1995). As discussed next, the Vicsek model and its variants can be interpreted as a decentralized feedback control system with time-varying network structure, offering a better understanding of the origin of collective behavior (Jadbabaie, Lin, and Morse, 2003; Moreau, 2005; Ren and Beard, 2005; Olfati-Saber, 2006).

1. Vicsek model and the alignment problem

The Vicsek model explains the origin of alignment, a key feature of flocking behavior (Vicsek *et al.*, 1995). It is a discrete-time stochastic model in which autonomous agents move in a plane with a constant speed v_0, initially following randomly chosen directions. The position $\mathbf{x}_i$ of agent i changes as

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1), \tag{112}$$

where the velocity of each agent has the same absolute value v_0. The direction of agent i is updated using a local rule that depends on the average of its own direction and the directions of its "neighbors," i.e., all agents within a distance r from agent i (Fig. 37), and some random perturbations. In other words,

$$\theta_i(t+1) = \langle\theta_i(t)\rangle_r + \Delta_i(t). \tag{113}$$

Here $\langle\theta_i(t)\rangle_r \equiv \arctan\left[\langle\sin\theta(t)\rangle_r / \langle\cos\theta(t)\rangle_r\right]$ denotes the average direction of the agents (including agent i) within a circle of radius r. The perturbations are contained in $\Delta_i(t)$, which is a random number taken from a uniform distribution in the interval $[-\eta/2, \eta/2]$. Therefore the final direction of agent i is obtained after rotating the average direction of the neighbors with a random angle. These random perturbations can be rooted in any stochastic or deterministic factors that affect the motion of the flocking agents.

The Vicsek model has three parameters: (i) the agent density ρ (number of agents in the area L^2), (ii) the speed v_0, and (iii) the magnitude of perturbations (i.e., the noise level) η. The model's order parameter is the normalized average velocity

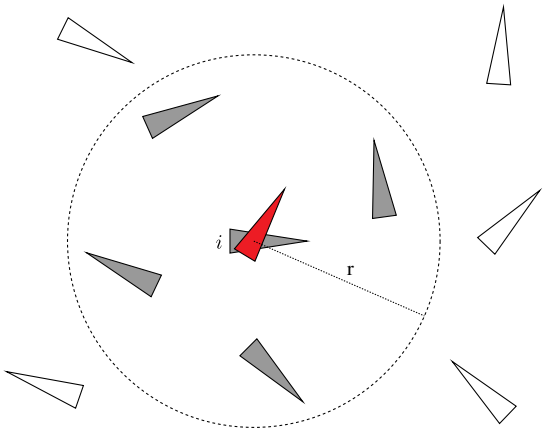

FIG. 37. Vicsek model. The direction of agent i at time $t+1$ (red) is the average of its own direction and the directions of all other agents at a distance less than r to agent i at time t (gray), up to some random perturbations. Agents outside this circle (white), do not contribute to the direction of agent i at time $t+1$.

$$\phi \equiv \frac{1}{Nv_0}\left|\sum_{i=1}^{N}\mathbf{v}_i\right|. \tag{114}$$

For small speed v_0, if we decrease the magnitude of perturbations η, the Vicsek model displays a continuous phase transition from a disordered phase [zero average velocity ϕ, implying that all agents move independently of each other, Fig. 38(b)] to an ordered phase when almost all agents move in the same direction, through a spontaneous symmetry breaking of the rotational symmetry [Fig. 38(d)]. This much-studied kinetic phase transition takes place despite the fact that each agent's set of nearest neighbors changes with time as the system evolves and the absence of centralized coordination.

Numerical results indicate that the phase transition is second order and the normalized average velocity ϕ scales as

$$\phi \sim [\eta_c(\rho) - \eta]^{\beta}, \tag{115}$$

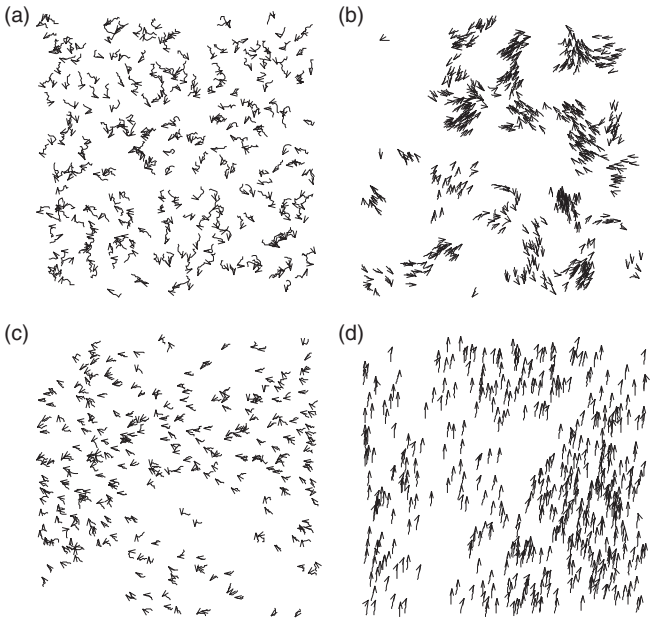

FIG. 38. Emergence of order in the Vicsek model. The panels show the agent velocity for varying values of the density and the noise level. The actual velocity of an agent is indicated by a small arrow while their trajectories for the last 20 time steps are shown as short continuous curves. The number of agents is $N = 300$, and the absolute velocity is $v_0 = 0.03$. (a) At $t = 0$, the positions and the direction of velocities are randomly distributed: $L = 7$, $\eta = 2.0$. Here we use the interaction radius r as the distance unit. (b) For small densities ($L = 25$) and low noise level ($\eta = 0.1$), the agents form groups that move together in random directions. (c) At higher densities ($L = 7$) and noise level ($\eta = 2.0$) the agents move randomly with some correlation. (d) When the density is large ($L = 5$) and the noise level is low ($\eta = 0.1$), the motion becomes ordered on a macroscopic scale and all agents tend to move in the same spontaneously selected direction. From Vicsek *et al.*, 1995.

where the critical exponent $\beta \approx 0.45$ and $\eta_c(\rho)$ is the critical noise for $L \to \infty$ (see Fig. 39) (Vicsek *et al.*, 1995). Many studies have explored the nature of this phase transition (whether it is first or second order), finding that two factors play an important role: (i) the precise way that the noise is introduced into the system, and (ii) the speed v_0 with which the agents move (Grégoire and Chate, 2004; Aldana *et al.*, 2007; Pimentel *et al.*, 2008; Aldana, Larralde, and Vazquez, 2009; Baglietto and Albano, 2009).

The Vicsek model raises a fundamental control problem: Under what conditions can the multiagent system display a particular collective behavior? Behind each flock of collectively moving agents, like biological organisms or robots, there is a dynamically changing or temporal network, where two agents are connected if they interact, e.g., if their distance is under a certain threshold. Since the agents are moving, the network of momentarily interacting units evolves in time in a complicated fashion.

To offer a control theoretical explanation for the emergence of the ordered phase in the Vicsek model, we consider the following updating rule (Jadbabaie, Lin, and Morse, 2003):

$$\theta_i(t+1) = \frac{1}{1+k_i(t)}\left(\theta_i(t) + \sum_{j\in\mathcal{N}_i(t)}\theta_j(t)\right). \tag{116}$$

Although the scalar average in Eq. (116) is fundamentally different from the vectorial average in Eq. (113), this updating rule still captures the essence of the Vicsek model in the absence of perturbation. More importantly, Eq. (116) can be considered as a decentralized feedback control system

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) + \mathbf{u}(t) \tag{117}$$

with the control input

$$\mathbf{u}(t) = -(\mathbf{D}_{\sigma(t)} + \mathbf{I})^{-1}\mathbf{L}_{\sigma(t)}\boldsymbol{\theta}(t). \tag{118}$$

Here $\mathbf{L}_p = \mathbf{D}_p - \mathbf{A}_p$ is the Laplacian matrix of graph G_p with $p \in \mathcal{P}$. $\mathbf{A}_p$ is the adjacency matrix of graph G_p and $\mathbf{D}_p$ is a diagonal matrix whose ith diagonal element is the degree of node i in the graph G_p. $\sigma(t)$: $0, 1, \ldots, \to \mathcal{P}$ is a switching signal whose value at time t is the index of the interaction graph at time t, i.e., $G(t)$.

If the interaction radius r is small, some agents are always isolated, implying that $G(t)$ is never connected. If r is large, then $G(t)$ is always a complete graph. The situation of interest is between the two extremes. The goal is to show that for any initial set of agent directions $\boldsymbol{\theta}(0)$ and for a large class of switching signals the directions of all agents will converge to the same steady state θ_{ss}, reaching alignment asymptotically. Mathematically, this means that the state vector $\boldsymbol{\theta}(t)$ converges to a vector of the form $\theta_{ss}\mathbf{1}$ with θ_{ss} the steady-state direction, i.e.,

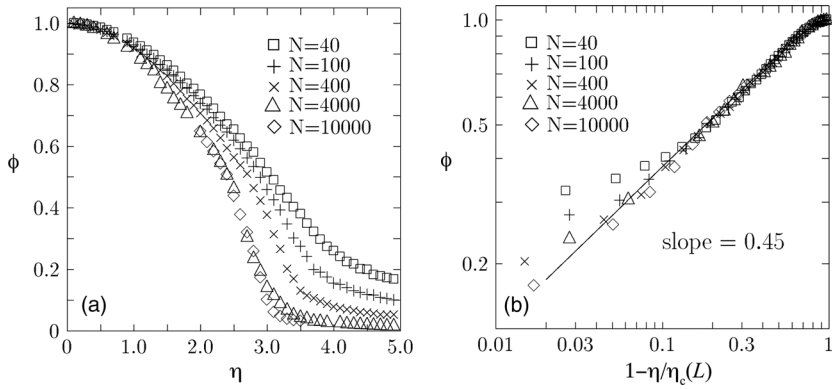

FIG. 39. Kinetic phase transition in the Vicsek model. (a) The normalized average velocity (ϕ) vs the magnitude of perturbations (noise level η) in cells of various sizes (L) with a fixed density $\rho = N/L^2 = 0.4$. As η decreases, ϕ increases, implying the emergence of order in the Vicsek model. (b) Dependence of ϕ on $[\eta_c(L) - \eta]/\eta_c$ in log-log scale. The slope of the lines is associated with the critical exponent β for which we get $\beta = 0.45 \pm 0.07$. The scaling behavior of ϕ observed in such a kinetic phase transition is analogous to what we often observe in continuous phase transitions in equilibrium systems. From Vicsek *et al.*, 1995.

$$\lim_{t\to\infty} \boldsymbol{\theta}(t) = \theta_{ss}\mathbf{1}, \tag{119}$$

where $\mathbf{1} \equiv (1, \ldots, 1)^{\mathrm{T}}_{N\times 1}$, representing the case when all agents move in the same direction.

If $G(t)$ is connected for all $t \geq 0$, then we can prove that alignment will be asymptotically reached (Jadbabaie, Lin, and Morse, 2003). But this condition is very stringent. It can be relaxed by considering that the agents are linked together across a time interval, i.e., the collection or union of graphs encountered along the interval is connected. It has been proven that if the N agents are linked together for each time interval, then the alignment will be asymptotically reached (Jadbabaie, Lin, and Morse, 2003). This result has been further extended by proving that if the collection of graphs is ultimately connected, i.e., there exists an initial time t_0 such that over the infinite interval $[t_0, \infty)$ the union graph $\mathcal{G} = \cup_{t=t_0}^{\infty} G_t$ is connected, then the alignment is asymptotically reached (Moreau, 2005).

Although the control theoretical analysis (Jadbabaie, Lin, and Morse, 2003; Moreau, 2005; Ren and Beard, 2005) is deterministic, ignoring the presence of noise, it offers rigorous theoretical explanations, based on the connectedness of the underlying graph, for some fundamental aspects of the Vicsek model. For example, by applying the nearest-neighbor rule, all agents tend to align the same direction despite the absence of centralized coordination and despite the fact that each agent's set of nearest neighbors changes in time. These control theoretical results suggest that to understand the effect of additive noise, we should focus on how noise inputs affect the connectivity of the associated neighbor graphs. For example, the numerical finding that for a fixed noise beyond a critical agent density all agents eventually become aligned can be adequately explained by percolation theory of random graphs (Jadbabaie, Lin, and Morse, 2003).

2. Alignment via pinning

While the virtue of the Vicsek model is its ability to spontaneously reach an ordered phase, we can also ask if such a phase can be induced externally. Therefore, we consider an effective pinning control strategy in which a single pinned node (agent) facilitates the alignment of the whole group. This is achieved by adding to the Vicsek model an additional agent, labeled 0, which acts as the group's leader. Agent 0 moves at the same constant speed v_0 as its N followers but with a fixed direction θ_0, representing the desired direction for the whole system. Each follower's neighbor set includes the leader whenever it is within the follower's circle of radius r. Hence we have

$$\theta_i(t+1) = \frac{1}{1+k_i(t)+b_i(t)}\left(\theta_i(t) + \sum_{j\in\mathcal{N}_i(t)} \theta_j(t) + b_i(t)\theta_0\right), \tag{120}$$

where $b_i(t) = 1$ whenever the leader is a neighbor of agent i and 0 otherwise. It has been proved that if the $(N + 1)$ agents are linked together for each time interval, then alignment will be asymptotically reached (Jadbabaie, Lin, and Morse, 2003). In other words, if the union of graphs of the $(N + 1)$ agents encountered along each time interval is connected, then eventually all the follower agents will align with the leader.

3. Distributed flocking protocols

Alignment, addressed by the Vicsek model, is only one feature of flocking behavior. Indeed, there are three heuristic rules for flocking (Reynolds, 1987): (i) cohesion: an attempt to stay close to nearby flock mates; (ii) separation: avoid collisions with nearby flock mates; and (iii) alignment: an attempt to match velocity with nearby flock mates.

We therefore need a general theoretical framework to design and analyze distributed flocking algorithms or protocols that embody these three rules. The formal approach described next extracts the interaction rules that can ensure the emergence of flocking behavior (Olfati-Saber, 2006).

Consider a gradient-based flocking protocol equipped with a velocity consensus mechanism, where each agent is steered by the control input

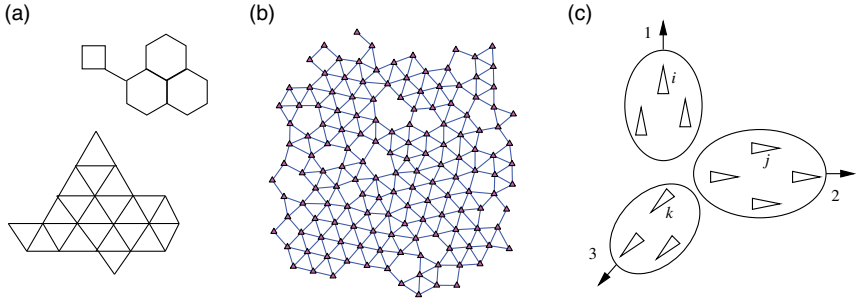

FIG. 40. Geometry of flocking and fragmentation. (a) Lattice-type flocking configuration in $D = 2$. In this ideal case, each agent is at the same distance from all of its neighbors on the proximity graph. (b) A quasilattice for $D = 2$ with $N = 150$ nodes. (c) Fragmentation phenomenon, where agents merge to form a few groups and different groups are moving in different directions. This configuration will never lead to flocking behavior. From Olfati-Saber, 2006.

$$\mathbf{u}_i = \mathbf{f}_i^{\mathrm{g}} + \mathbf{f}_i^{\mathrm{d}}. \tag{121}$$

The first term

$$\mathbf{f}_i^{\mathrm{g}} \equiv -\nabla_{\mathbf{q}_i} V_i(\mathbf{q}) \tag{122}$$

is gradient based and regulates the distance between agent i and its neighbors, avoiding the collision and cohesion of the agents. This term is derived from a smooth collective potential function $V_i(\mathbf{q})$, which has a unique minimum when each agent is at the same distance from all of its neighbors on the proximity graph $G(\mathbf{q})$, representing the ideal case for flocking [Fig. 40(a)]. The second term

$$\mathbf{f}_i^{\mathrm{d}} = \sum_{j \in \mathcal{N}_i(t)} a_{ij}(t)(\mathbf{p}_j - \mathbf{p}_i) \tag{123}$$

regulates the velocity of agent i to match the average velocity of its neighbors, being responsible for the velocity alignment. Here the weighted spatial adjacency matrix $\mathbf{A}(t) = [a_{ij}(t)]$ is calculated from the proximity network $G(\mathbf{q})$. The flocking protocol (121) embodies all three rules of Reynolds. However, for a generic initial state and a large number of agents (e.g., $N > 100$), the protocol (121) leads to fragmentation, rather than flocking (Olfati-Saber, 2006), meaning that the agents spontaneously form several groups, where different groups move in different directions [Fig. 40(c)]. To resolve this fragmentation issue, we introduce a navigational feedback term to the control input of each agent

$$\mathbf{u}_i = \mathbf{f}_i^{\mathrm{g}} + \mathbf{f}_i^{\mathrm{d}} + \mathbf{f}_i^{\gamma}, \tag{124}$$

where

$$\mathbf{f}_i^{\gamma} = -c_1(\mathbf{q}_i - \mathbf{q}_\gamma) - c_2(\mathbf{p}_i - \mathbf{p}_\gamma) \tag{125}$$

drives agent i to follow a group objective. The group objective can be considered as a virtual leader with the following equation of motion:

$$\dot{\mathbf{q}}_\gamma = \mathbf{p}_\gamma, \qquad \dot{\mathbf{p}}_\gamma = \mathbf{f}_\gamma(\mathbf{q}_\gamma, \mathbf{p}_\gamma), \tag{126}$$

where $\mathbf{q}_\gamma, \mathbf{p}_\gamma, \mathbf{f}_\gamma(\mathbf{q}_\gamma, \mathbf{p}_\gamma) \in \mathbb{R}^D$ are the position, velocity, and acceleration (control input) of the virtual leader, respectively. By taking into account the navigational feedback, the protocol (124) enables a group of agents to track a virtual leader that moves at a constant velocity and hence leads to flocking behavior (see Fig. 41) (Olfati-Saber, 2006).

Note that protocol (124) requires all agents to be informed, i.e., to know the group objective, or, equivalently, the current state $(\mathbf{q}_\gamma, \mathbf{p}_\gamma)$ of the virtual leader. It turns out that this is not necessary for flocking. Motivated by the idea of pinning control, it has been shown that, even when only a fraction of agents are informed (or pinned), the flocking protocol (124) still enables all the informed agents to move with the desired constant velocity. An uninformed agent will also move with the desired velocity if it can be influenced by the informed agents from time to time (Su, Wang, and Lin, 2009). Numerical simulations suggest that the larger the informed group is, the bigger fraction of agents will move with the desired velocity (Su, Wang, and Lin, 2009).

If the virtual leader travels with a varying velocity $\mathbf{p}_\gamma(t)$, the flocking protocol (124) enables all agents to eventually achieve a common velocity. Yet this common velocity is not guaranteed to match $\mathbf{p}_\gamma(t)$. To resolve this issue, we incorporate the acceleration of the virtual leader into the navigational feedback (125) as follows:

$$\mathbf{f}_i^{\gamma} = \mathbf{f}_\gamma(\mathbf{q}_\gamma, \mathbf{p}_\gamma) - c_1(\mathbf{q}_i - \mathbf{q}_\gamma) - c_2(\mathbf{p}_i - \mathbf{p}_\gamma). \tag{127}$$

The resulting protocol enables the asymptotic tracking of the virtual leader with a varying velocity, ensuring that the position and velocity of the center of mass of all agents will converge exponentially to those of the virtual leader (Su, Wang, and Lin, 2009).

In summary, the combination of control theoretical and network science approaches can help us understand the emergence of order in multiagent systems. These tools are indispensable if we wish to understand how to induce order externally, aiming to control the collective behavior of the system.

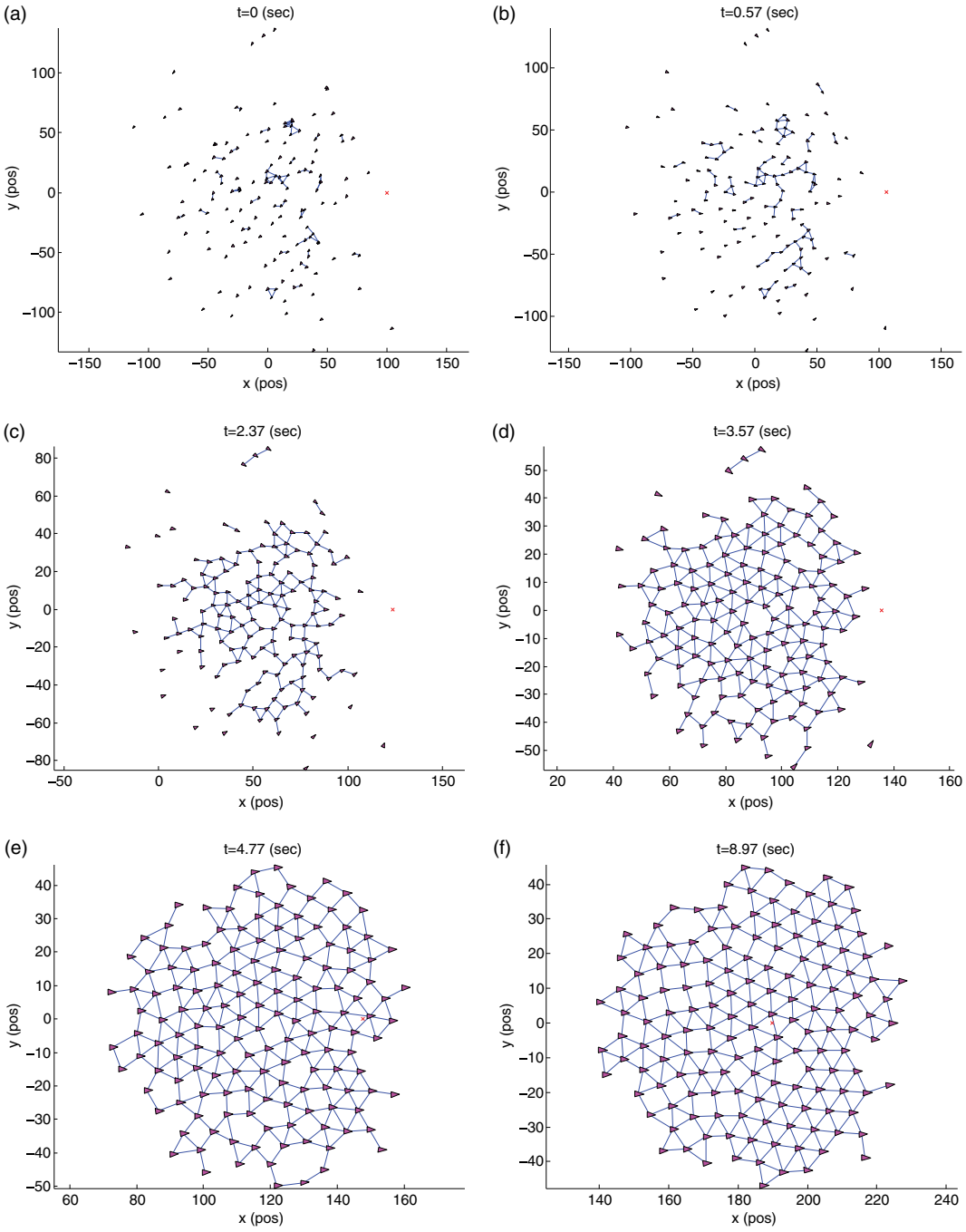

FIG. 41. Flocking behavior in multiagent systems. After the application of the flocking algorithm (124) for a few seconds, the flocking of $N = 100$ agents in $2D$ is observed. From Olfati-Saber, 2006.

VII. OUTLOOK

Given the rapid advances in the control of complex networked systems, we have chosen to focus on a group of results that will likely stay with us for many years to come. The process of organizing the material has also exposed obvious gaps in our knowledge. Therefore, next we highlight several research topics that must be addressed to realize the potential of the control of complex systems. Some of these may be addressed shortly; others, however, may continue to challenge the community for many years to come.

A. Stability of complex systems

Stability is a fundamental issue in the analysis and the design of a control system, because an unstable system is extremely difficult and costly to control, and such a system can also be potentially dangerous (Slotine and Li, 1991; Chen, 2001). Loosely speaking, a system is stable if its trajectories do not change too much under small perturbations.

The stability of a nonlinear dynamical system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$ can be analyzed by the Lyapunov stability theory (LST), without explicitly integrating the differential equation. LST

includes two methods: (i) The indirect (or linearization) method, concerned with small perturbation around a system's equilibrium points $\mathbf{x}^*$ and the stability conclusion, is inferred from a linear approximation of the nonlinear systems around this equilibrium point. This justifies the use of linear control for the design and analysis of weakly nonlinear systems. (ii) The direct method is based on the so-called Lyapunov function—an "energylike" scalar function whose time variation can be viewed as "energy dissipation." It is not restricted to small perturbations and in principle can be applied to any dynamical system. Yet we lack a general theory to find a suitable Lyapunov function for an arbitrary system. We have to rely on our experience and intuition to formulate Lyapunov functions (Slotine and Li, 1991).

For a wide range of complex systems certain diagonal-type Lyapunov functions are useful for stability analysis (Kaszkurewicz *et al.*, 2000). More importantly, in many cases the necessary and sufficient conditions for the stability of nonlinear systems are also the necessary and sufficient conditions for the diagonal stability of a certain matrix associated with the nonlinear system. This matrix naturally captures the underlying network structure of the nonlinear dynamical system.

Matrix diagonal stability is a well-known notion in stability analysis since its introduction by Volterra around 1930 in the context of ecological systems (Volterra, 1931). Yet its usefulness is limited by the difficulty of characterizing the class of large diagonally stable matrices. Although there are efficient optimization-based algorithms to numerically check if a given matrix is diagonally stable (Boyd *et al.*, 1994), there are no effective theoretical tools to characterize general large diagonally stable matrices. Recently, however, necessary and sufficient diagonal stability conditions for matrices associated with special interconnection or network structures were studied (Arcak and Sontag, 2006, 2008; Arcak, 2011), improving our understanding of the stability of gene regulatory and ecological networks. More research is required to understand stability, an important prerequisite for control.

The stability concepts we discussed previously consider perturbations of initial conditions for a fixed dynamical system. There is another important notion of stability, i.e., structural stability, which concerns whether the qualitative behavior of the system trajectories will be affected by small perturbations of the system model itself (Andronov and Pontryagin, 1937; Kuznetsov, 2004). The notion of structural stability has not been well explored in complex networked systems.

B. Controlling adaptive networks

Adaptability, representing a system's ability to respond to changes in the external conditions, is a key characteristic of complex systems. Indeed, the structure of many real networks coevolves with the dynamics that takes place on them, naturally adapting to shifting environments (Gross and Sayama, 2009).

Adaptive networks, also known as state-dependent dynamic networks in control theory (Mesbahi, 2005; Mesbahi and Egerstedt, 2010), are collections of units that interact through a network, whose topology evolves as the state of the units changes with time. Adaptive networks are a special class of temporal networks, whose edges are not continuously active (Karsai *et al.*, 2011; Holme and Saramäki, 2012; Pan and Li, 2014; Pósfai and Hövel, 2014). If the temporal order of the network snapshots at different time points depend on the states of the nodes, then the temporal network is adaptive. A special case of adaptive networks is switched systems, which consist of a family of subsystems and a switching law that orchestrates the switching among them (Xie, Zheng, and Wang, 2002; Xie and Wang, 2003).

Adaptive networks are ubiquitous, especially in biology. For example, mycelial fungi and acellular slime molds grow as self-organized networks that explore new territory for food sources, while maintaining an effective internal transport system to resist continuous attacks or random damage (Fessel *et al.*, 2012). Honed by evolution, these biological networks are examples of adaptive transportation networks, balancing real-world compromises between search strategy and transport efficiency (Tero *et al.*, 2010). The genome is also an intriguing example of an adaptive network, where the chromosomal geometry directly relates to the genomic activity, which in turn strongly correlates with geometry (Rajapakse, Groudine, and Mesbahi, 2011). Similarly, neuronal connections (synapses) in our brains can strengthen or weaken and form, in response to changes in brain activity, a phenomenon called synaptic plasticity (Perin, Berger, and Markram, 2011; Bayati and Valizadeh, 2012).

A comprehensive analytical framework is needed to address the control of adaptive networks. This framework must recognize the network structure itself as a dynamical system, together with the nodal or edge dynamics on the network, capturing the feedback mechanisms linking the structure and dynamics. Studying the controllability of such systems would be a natural starting point because seemingly mild limitations on either the network structure or the dynamical rules may place severe constraints on the controllability of the whole system. Identifying these constraints is crucial if we want to refrain from improving systems that already operate close to their fundamental limits.

C. Controlling networks of networks

Many natural and engineered systems are composed of a set of coupled layers or a network of subsystems, characterized by different time scales and structural patterns. New notions, from multiplex networks (Boccaletti *et al.*, 2014; Kivelä *et al.*, 2014) to networks of networks (D'Agostino and Scala, 2014; Gao, Li, and Havlin, 2014), have recently been proposed to explore the properties of these systems, focusing mainly on their structural integrity and robustness. Consider a multiplex network, i.e., a set of coupled layered networks, where different layers have different characteristics. We can model such a system as a layered network, whose interconnections between layers capture the interactions between a node in one layer and its counterpart in another layer. Similarly, in a network of networks each node itself is a network or a multi-input and multi-output (MIMO) subsystem. Different nodes (subsystems) could have totally different dimensions and dynamics. This is rather different from the control framework discussed in much of this paper, where we typically

assumed that all the nodes share the same type of dynamics or even just scalar dynamics (with state variables $x_i \in \mathbb{R}$ for all nodes).

Developing a framework to control networks of networks is a necessary step if we want to understand the control principles of complex systems. Early attempts have focused on the issues of controllability or observability with linear dynamics (Chapman, Nabi-Abdolyousefi, and Mesbahi, 2014; Yuan *et al.*, 2014; Menichetti, Dall'Asta, and Bianconi, 2016; Wang *et al.*, 2016; Zhou, 2015; Zhang, Garas, and Schweitzer, 2016). For example, some controllability conditions on the overall network topology, the node dynamics, the external control inputs, and the inner interactions have been derived for a networked MIMO system (Wang *et al.*, 2016). Interestingly, the controllability of the networked MIMO system is an integrated result of multiple factors, which cannot be decoupled into the controllability of the individual subsystem or the properties solely determined by the network topology. Despite these efforts, we lack a general framework to systematically explore the control of networks of networks. Yet the problem's importance will likely trigger more research in both network science and control theory.

D. Noise

Complex systems, especially biological systems, are noisy. They are affected by two kinds of noise: the intrinsic randomness of individual events and the extrinsic influence of changing environments (Lestas, Vinnicombe, and Paulsson, 2010; Hilfinger and Paulsson, 2011). Consider, for example, regulatory processes in a cell. The intrinsic noise is rooted in the low copy number of biomolecules or diffusive cellular dynamics. In particular, if N is the number of molecules in the system, fluctuations in N lead to statistical noise with intensity in the order of $N^{-1/2}$. For large N, we can assume that a continuous deterministic dynamics effectively describes the changes of the average concentrations. However, for small N the statistical noise cannot be ignored. For example, gene regulation may be affected by large fluctuations due to the low copy number of transcription factors. The extrinsic noise of a biological system is mainly due to the changing environments experienced by the system. The environmental change may have microscopic origin (such as cellular age or cell cycle stage and organelle distributions) or can be related to the macroscopic physical or chemical environment (like illumination conditions, temperature, pressure, and pH level). To infer or reconstruct the states of a biological system, we also need to deal with the measurement error, which is independent of the biological system and can also be considered as extrinsic noise.

Both internal and external noises are known to affect the control of complex systems. At this time we lack a full understanding of the role of noise or stochastic fluctuations on the control of complex systems.

E. Controlling quantum networks

Quantum control theory aims to offer practical methods to control quantum systems. Despite recent progress, quantum control theory is still in its infancy (Dong and Petersen, 2010) for several reasons. First, in classical control it is assumed that the measurement does not affect the measured system. By contrast, in quantum control it is difficult, if not impossible, to acquire information about quantum states without destroying them. Second, some classes of quantum control tasks, such as controlling quantum entanglement and protecting quantum coherence, are unique for quantum systems. In other words, there are no corresponding tasks in classical control theory.

The notion of quantum networks was recently proposed by the quantum information community (Acín, Cirac, and Lewenstein, 2007; Perseguers, Cirac, and Wehr, 2008; Cuquet and Calsamiglia, 2009, 2012; Lapeyre, Wehr, and Lewenstein, 2009; Perseguers, 2010; Perseguers *et al.*, 2010, 2013; Czekaj, Chhajlany, and Horodecki, 2012; Ritter *et al.*, 2012), offering fresh perspectives in the field of complex networks. In a quantum network, each node possesses exactly one qubit for each of its neighbors. Since nodes can act on these qubits, they are often called "stations." The edge between two nodes represents the entanglement between two qubits. The degree of entanglement between two nodes can be considered as the connection probability (p) in the context of classical random graphs.

In a classical random graph if we let p scale with the graph size as $p \sim N^z$, increasingly complex subgraphs appear as z exceeds a series of thresholds. For example, for $z \leq -2$ almost all graphs contain only isolated nodes and edges. When z passes through $-3/2$ (or $-4/3$), trees of order 3 (or 4) suddenly appear. As z approaches -1, trees and cycles of all orders appear (Albert and Barabási, 2002). Surprisingly, in quantum networks any subgraph can be generated by local operations and classical communication, provided that the entanglement between pairs of nodes scales with the graph size as $p \sim N^{-2}$ (Perseguers *et al.*, 2010). In other words, thanks to the superposition principle and the ability to coherently manipulate the qubits at the stations, even for the lowest nontrivial connection probability that is just sufficient to get simple connections in a classical graph, we obtain quantum subgraphs of any complexity.

This result illustrates that quantum networks have unique properties that are impossible in their classical counterparts. Hence, the control of quantum complex networks will require new methodologies.

F. Conclusion

Revealing the control principles of complex networked systems remains a challenging problem that, given its depth and applications, will probably engage multiple research communities for the next decade. In this review we aimed to summarize in a coherent fashion the current body of knowledge on this fascinating topic. This forced us to explore key notions in control theory, such as controllability and observability, but also to explore how to steer a complex networked system to a desired final state or trajectory or a desired collective behavior. There are many outstanding open questions to be addressed, advances on which will require interdisciplinary collaborations. We hope that this review will catalyze new interdisciplinary approaches, moving our

understanding of control forward and enhancing our ability to control complex systems.

ACKNOWLEDGMENTS

We thank Jianhua Xing, Haijun Zhou, Wenxu Wang, Tamás Vicsek, Edward Ott, David Luenberger, Justin Ruths, Atsushi Mochizuki, Daniel J. Gauthier, Andrew Whalen, Adilson Motter, and Francesco Sorrentino for allowing us to reproduce their figures. We are grateful to Marco Tulio Angulo, Guanrong Chen, Gang Yan, Aming Li, Zoltán Toroczkai, Frank Schweitzer, Sérgio Pequito, Francesco Sorrentino, Atsushi Mochizuki, Bernold Fiedler, Jean-Jacques Slotine, Hiroki Sayama, Travis E. Gibson, Chuliang Song, and Yandong Xiao for reading our manuscript and providing helpful suggestions. This research was supported by the John Templeton Foundation: Mathematical and Physical Sciences (Grant No. PFI-777), Army Research Laboratories (ARL) Network Science (NS) Collaborative Technology Alliance (CTA) Grant No. ARL NS-CTA W911NF-09-2-0053, and the European Commission Grants No. FP7317532 (MULTIPLEX) and No. 641191 (CIMPLEX).

REFERENCES

Acín, A., J. I. Cirac, and M. Lewenstein, 2007, Nat. Phys. **3**, 256.
Aguirre, L. A., S. B. Bastos, M. A. Alves, and C. Letellier, 2008, Chaos **18**, 013123.
Aguirre, L. A., and C. Letellier, 2005, J. Phys. A **38**, 6311.
Akutsu, T., M. Hayashida, W.-K. Ching, and M. K. Ng, 2007, J. Theor. Biol. **244**, 670.
Albert, R., and A.-L. Barabási, 2002, Rev. Mod. Phys. **74**, 47.
Aldana, M., V. Dossetti, C. Huepe, V. M. Kenkre, and H. Larralde, 2007, Phys. Rev. Lett. **98**, 095702.
Aldana, M., H. Larralde, and B. Vazquez, 2009, Int. J. Mod. Phys. B **23**, 3661.
Aminzare, Z., and E. D. Sontag, 2014, IEEE Trans. Network Sci. Eng. **1**, 91.
Andronov, A. A., and L. S. Pontryagin, 1937, Dokl. Akad. Nauk SSSR **14**, 247.
Anguelova, M., 2004, "Nonlinear Observability and Identifiability: General Theory and a Case Study of a Kinetic Model for S. cerevisiae," Ph.D. thesis (Chalmers University of Technology and Göteborg University).
Angulo, M. T., J. A. Moreno, G. Lippner, A.-L. Barabási, and Y.-Y. Liu, 2015, arXiv:1508.03559v2.
Antsaklis, P. J., and A. N. Michel, 1997, *Linear Systems* (McGraw-Hill, New York).
Arcak, M., 2011, IEEE Trans. Autom. Control **56**, 2766.
Arcak, M., and E. Sontag, 2008, Math. Biosci. Eng. **5**, 1.
Arcak, M., and E. D. Sontag, 2006, Automatica **42**, 1531.
Arenas, A., A. Daz-Guilera, J. Kurths, Y. Moreno, and C. Zhou, 2008, Phys. Rep. **469**, 93.
Baglietto, G., and E. V. Albano, 2009, Comput. Phys. Commun. **180**, 527.
Barabási, A.-L., 2016, *Network Science* (Cambridge University Press, Cambridge, UK).
Barabási, A.-L., and R. Albert, 1999, Science **286**, 509.
Barabási, A.-L., N. Gulbahce, and J. Loscalzo, 2011, Nat. Rev. Genet. **12**, 56.
Barahona, M., and L. M. Pecora, 2002, Phys. Rev. Lett. **89**, 054101.
Bayati, M., and A. Valizadeh, 2012, Phys. Rev. E **86**, 011925.
Bellman, R., and K. J. Aström, 1970, Math. Biosci. **7**, 329.
Belykh, I., V. Belykh, and M. Hasler, 2006, Chaos **16**, 015102.
Belykh, I., M. Hasler, M. Lauret, and H. Nijmeijer, 2005, Int. J. Bifurcation Chaos Appl. Sci. Eng. **15**, 3423.
Belykh, I. V., V. N. Belykh, and M. Hasler, 2004a, Physica D (Amsterdam) **195**, 188.
Belykh, V. N., I. V. Belykh, and M. Hasler, 2004b, Physica D (Amsterdam) **195**, 159.
Berényi, A., M. Belluscio, D. Mao, and G. Buzsáki, 2012, Science **337**, 735.
Bernardo, M. d., and P. DeLellis, 2014, Scholarpedia **9**, 29958.
Besançon, G., 2007, Ed., *Nonlinear Observers and Applications*, Lecture Notes in Control and Information Sciences (Springer-Verlag, Berlin/Heidelberg).
Besanon, G., 2007, *Nonlinear Observers and Applications* (Springer, New York).
Bloch, A. M., 2003, *Nonholonomic Mechanics and Control* (Springer-Verlag, Berlin).
Boccaletti, S., G. Bianconi, R. Criado, C. del Genio, J. Gmez-Gardees, M. Romance, I. Sendia-Nadal, Z. Wang, and M. Zanin, 2014, Phys. Rep. **544**, 1.
Boccaletti, S., C. Grebogi, Y.-C. Lai, H. Mancini, and D. Maza, 2000, Phys. Rep. **329**, 103.
Bollobás, B., 2001, *Random Graphs* (Cambridge University Press, Cambridge).
Boyd, S., L. E. Ghaoui, E. Feron, and V. Balakrishnan, 1994, *Linear Matrix Inequalities in System and Control Theory* (SIAM, Philadelphia, PA).
Brockett, R., 1972, SIAM J. Control **10**, 265.
Brockett, R., 1982, in *New Directions in Applied Mathematics*, edited by P. Hilton and G. Young (Springer, New York), pp. 11–27.
Chapman, A., and M. Mesbahi, 2013, in *2013 American Control Conference* (IEEE, New York), pp. 6126–6131, doi:10.1109/ACC.2013.6580798.
Chapman, A., M. Nabi-Abdolyousefi, and M. Mesbahi, 2014, IEEE Trans. Autom. Control **59**, 2668.
Chen, G., 2001, "Stability of nonlinear systems," in *Wiley Encyclopedia of Electrical and Electronics Engineering* (John Wiley & Sons, Inc., New York).
Chen, G., 2014, Int. J. Control Autom. Syst. **12**, 221.
Chen, G., and X. Dong, 1998, *From Chaos to Order: Methodologies, Perspectives and Applications* (World Scientific, Singapore).
Chen, G., and Z. Duan, 2008, Chaos **18**, 037102.
Chen, M., 2006, IEEE Trans. Circuits Syst. II **53**, 1185.
Chen, M., 2007, Phys. Rev. E **76**, 016104.
Chen, M., 2008, IEEE Trans. Circuits Syst. I **55**, 1335.
Chen, T., X. Liu, and W. Lu, 2007, IEEE Trans. Circuits Syst. I **54**, 1317.
Chen, Y.-Z., L.-Z. Wang, W.-X. Wang, and Y.-C. Lai, 2016, R. Soc. Open Sci. **3**, 160064.
Cheng, D., and H. Qi, 2009, Automatica **45**, 1659.
Cheng, D., H. Qi, and A. Xue, 2007, J. Syst. Sci. Complex. **20**, 304.
Cohen, R., K. Erez, D. ben Avraham, and S. Havlin, 2000, Phys. Rev. Lett. **85**, 4626.
Commault, C., J.-M. Dion, and J. van der Woude, 2002, Kybernetika **38**, 503 [https://eudml.org/doc/33600].
Conte, G., C. H. Moog, and A. M. Perdon, 2007, *Algebraic Methods for Nonlinear Control Systems* (Springer, New York).
Cormen, T. H., C. E. Leiserson, and R. L. Rivest, 1990, *Introduction to Algorithms* (MIT Press, Cambridge, MA).
Cornelius, S. P., W. L. Kath, and A. E. Motter, 2013, Nat. Commun. **4**, 1942.

Coron, J.-M., 2009, *Control and Nonlinearity* (American Mathematical Society, Providence, RI).

Cortesi, F. L., T. H. Summers, and J. Lygeros, 2014, in *53rd IEEE Conference on Decision and Control* (IEEE, New York), doi:10.1109/CDC.2014.7039832.

Cowan, N. J., E. J. Chastain, D. A. Vilhena, J. S. Freudenberg, and C. T. Bergstrom, 2012, PLoS One **7**, e38398.

Cuquet, M., and J. Calsamiglia, 2009, Phys. Rev. Lett. **103**, 240503.

Cuquet, M., and J. Calsamiglia, 2012, Phys. Rev. A **86**, 042304.

Czekaj, L., R. W. Chhajlany, and P. Horodecki, 2012, Phys. Rev. A **85**, 032328.

D'Agostino, G., and A. Scala, 2014, *Networks of Networks: The Last Frontier of Complexity* (Springer International Publishing, Switzerland).

Davison, E. J., 1977, Automatica **13**, 109.

de Figueiredo, R. J., and G. Chen, 1993, *Nonlinear Feedback Control Systems: An Operator Theory Approach* (Academic Press, New York).

DeLellis, P., M. di Bernardo, and G. Russo, 2011, IEEE Trans. Circuits Syst. I **58**, 576.

DeLellis, P., M. di Bernardo, and L. Turci, 2010, in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems* (IEEE, New York), pp. 685–688, doi:10.1109/ISCAS.2010.5537492.

Dion, J.-M., C. Commault, and J. van der Woude, 2003, Automatica **39**, 1125.

Diop, S., and M. Fliess, 1991a, in *Proceedings of the 30th IEEE Conference on Decision and Control* (IEEE, New York), pp. 714–719, doi:10.1109/CDC.1991.261405.

Diop, S., and M. Fliess, 1991b, in Proceedings of the First European Control Conference (ECC'91), pp. 152–157.

Dong, D., and I. Petersen, 2010, IET Control Theory Applications **4**, 2651.

Dorogovtsev, S. N., A. V. Goltsev, and J. F. F. Mendes, 2008, Rev. Mod. Phys. **80**, 1275.

Duleba, I., 1998, *Algorithms of Motion Planning for Nonholonomic Robots* (Technical University of Wroclaw, Wroclaw, Poland).

Elliot, D., 1971, J. Diff. Equ. **10**, 364.

Erdős, P., and A. Rényi, 1960, Publ. Math. Inst. Hung. Acad. Sci., Ser. A **5**, 17 [https://www.emis.de/classics/Erdos/cit/10316301.htm].

Fessel, A., C. Oettmeier, E. Bernitt, N. Gauthier, and H.-G. Dobereiner, 2012, Phys. Rev. Lett. **109**, 078103.

Fiedler, B., A. Mochizuki, G. Kurosawa, and D. Saito, 2013, J. Dyn. Differ. Equ. **25**, 563.

Fradkov, A. L., and R. J. Evans, 2005, Annu. Rev. Control **29**, 33.

Friedland, B., 1996, *Advanced Control System Design* (Prentice Hall, Englewood Cliffs, NJ).

Gao, J., D. Li, and S. Havlin, 2014, Natl. Sci. Rev. **1**, 346.

Gao, J., Y.-Y. Liu, R. M. D'Souza, and A.-L. Barabási, 2014, Nat. Commun. **5**, 5415.

Gardner, T. S., D. Di Bernardo, D. Lorenz, and J. J. Collins, 2003, Science **301**, 102.

Glad, S., and L. Ljung, 1990, in *Proceedings of the 29th IEEE Conference on Decision and Control* (IEEE, New York), pp. 3236–3240, doi:10.1109/CDC.1990.203389.

Glover, K., and L. M. Silverman, 1976, IEEE Trans. Autom. Control **21**, 534.

Goh, K.-I., B. Kahng, and D. Kim, 2001, Phys. Rev. Lett. **87**, 278701.

Grégoire, G., and H. Chate, 2004, Phys. Rev. Lett. **92**, 025702.

Gross, T., and H. Sayama, 2009, *Adaptive Networks: Theory, Models and Applications (Understanding Complex Systems)* (NECSI, Cambridge, MA).

Hautus, M., 1969, Proc. K. Ned. Akad. Wet., Ser. A: Math. Sci. **72**, 443.

Haynes, G., and H. Hermes, 1970, SIAM J. Control **8**, 450.

Heinrich, R., and S. Schuster, 1996, *The Regulation of Cellular Systems* (Springer, New York).

Hermann, R., and A. J. Krener, 1977, IEEE Trans. Autom. Control **22**, 728.

Hilfinger, A., and J. Paulsson, 2011, Proc. Natl. Acad. Sci. U.S.A. **108**, 12167.

Holme, P., and J. Saramäki, 2012, Phys. Rep. **519**, 97.

Hopcroft, J. E., and R. M. Karp, 1973, SIAM J. Comput. **2**, 225.

Hosoe, S., 1980, IEEE Trans. Autom. Control **25**, 1192.

Hosoe, S., and K. Matsumoto, 1979, IEEE Trans. Autom. Control **24**, 963.

Hou, L., S. Lao, J. Bu, and L. Bai, 2013, in the *Third International Conference on Intelligent System Design and Engineering Applications (ISDEA)* (IEEE, New York), pp. 709–711, doi:10.1109/ISDEA.2012.168.

Hou, L.-L., S.-Y. Lao, G. Liu, and L. Bai, 2012, Chin. Phys. Lett. **29**, 108901.

Hubler, A., R. Georgii, M. Kuchler, W. Stelzl, and E. Luscher, 1988, Helv. Phys. Acta **61**, 897 [http://www.e-periodica.ch/digbib/view?pid=hpa-001:1988:61#899].

Iglesias, P. A., and B. P. Ingalls, 2009, *Control Theory and Systems Biology* (The MIT Press, Cambridge, MA).

Isidori, A., 1995, *Nonlinear Control Systems* (Springer-Verlag, Berlin).

Jadbabaie, A., J. Lin, and A. S. Morse, 2003, IEEE Trans. Autom. Control **48**, 988.

Jarczyk, J. C., F. Svaricek, and B. Alt, 2011, in *50th IEEE Conference on Decision and Control and European Control Conference* (IEEE, New York), pp. 1213–1218, doi:10.1109/CDC.2011.6160392.

Jia, T., Y.-Y. Liu, E. Csóka, M. Pósfai, J.-J. Slotine, and A.-L. Barabási, 2013, Nat. Commun. **4**, 2002.

Jiang, G.-P., W. K.-S. Tang, and G. Chen, 2003, Chaos Solitons Fractals **15**, 925.

Jin, X.-Z., G.-H. Yang, and W.-W. Che, 2012, IEEE Trans. Neural Netw. Learn. Syst. **23**, 1345.

Johnston, R. D., G. W. Barton, and M. L. Brisk, 1984, Int. J. Control **40**, 257.

Jose, C. N., and A. Tatsuya, 2012, New J. Phys. **14**, 073005.

Kailath, T., 1980, *Linear Systems* (Prentice-Hall, Inc., Englewood Cliffs, NJ).

Kalman, R. E., 1963, J. Soc. Indus. Appl. Math. Ser. A **1**, 152.

Karp, R. M., 1972, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, edited by R. E. Miller and J. W. Thatcher (Plenum, New York), Chap. 9, pp. 85–103.

Karsai, M., M. Kivelä, R. K. Pan, K. Kaski, J. Kertész, A.-L. Barabási, and J. Saramäki, 2011, Phys. Rev. E **83**, 025102.

Kaszkurewicz, E., A. Bhaya, A. Bhaya, and E. Kaszkurewicz, 2000, *Matrix Diagonal Stability in Systems and Computation* (Springer Science+Business Media, New York).

Khan, U. A., and M. Doostmohammadian, 2011, in the *2011 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)* (IEEE, New York), p. 137, doi:10.1109/CAMSAP.2011.6135906.

Khan, U. A., and J. M. F. Moura, 2008, IEEE Trans. Signal Process. **56**, 4919.

Kholodenko, B. N., A. Kiyatkin, F. J. Bruggeman, E. Sontag, H. V. Westerhoff, and J. B. Hoek, 2002, Proc. Natl. Acad. Sci. U.S.A. **99**, 12841.

Kirk, D. E., 2004, *Optimal Control Theory: An Introduction* (Dover Publications, New York).

Kivelä, M., A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, 2014, J. Complex Netw. **2**, 203.
Kuznetsov, Y. A., 2004, *Elements of Applied Bifurcation Theory* (Springer, New York).
Lai, Y.-C., 1996, Phys. Lett. A **221**, 375.
Lai, Y.-C., 2014, Natl. Sci. Rev. **1**, 339.
Lapeyre, G., J. Wehr, and M. Lewenstein, 2009, Phys. Rev. A **79**, 042324.
Lee, E. B., and L. Markus, 1968, *Foundations of optimal control theory*, SIAM series in applied mathematics (Wiley, New York).
Lestas, I., G. Vinnicombe, and J. Paulsson, 2010, Nature (London) **467**, 174.
Letellier, C., and L. Aguirre, 2005, Phys. Rev. E **72**, 056202.
Letellier, C., L. Aguirre, and J. Maquet, 2006, Commun. Nonlinear Sci. Numer. Simul. **11**, 555.
Letellier, C., and L. a. Aguirre, 2010, Phys. Rev. E **82**, 016204.
Lewis, F. L., D. L. Vrabie, and V. L. Syrmos, 2012, *Optimal Control* (John Wiley & Sons, Inc., New York), 3rd ed.
Lezon, T. R., J. R. Banavar, M. Cieplak, A. Maritan, and N. V. Fedoroff, 2006, Proc. Natl. Acad. Sci. U.S.A. **103**, 19033.
Li, K., M. Small, and X. Fu, 2007, Phys. Rev. E **76**, 056213.
Li, P., M. Chen, Y. Wu, and J. Kurths, 2009, Phys. Rev. E **79**, 067102.
Li, X., and G. Chen, 2003, IEEE Trans. Circuits Syst. I **50**, 1381.
Li, X., X. Wang, and G. Chen, 2004, IEEE Trans. Circuit Syst. **51**, 2074.
Lin, C.-T., 1974, IEEE Trans. Autom. Control **19**, 201.
Linnemann, A., 1986, IEEE Trans. Autom. Control **31**, 638.
Liu, B., T. Chu, L. Wang, and G. Xie, 2008, IEEE Trans. Autom. Control **53**, 1009.
Liu, Y.-Y., E. Csóka, H. Zhou, and M. Pósfai, 2012, Phys. Rev. Lett. **109**, 205703.
Liu, Y.-Y., J.-J. Slotine, and A.-L. Barabási, 2011a, Nature (London) **473**, 167.
Liu, Y.-Y., J.-J. Slotine, and A.-L. Barabási, 2011b, Nature (London) **478**, E4.
Liu, Y.-Y., J.-J. Slotine, and A.-L. Barabási, 2012, PLoS One **7**, e44459.
Liu, Y.-Y., J.-J. Slotine, and A.-L. Barabási, 2013, Proc. Natl. Acad. Sci. U.S.A. **110**, 2460.
Ljung, L., 1999, *System Identification: Theory for the User* (Prentice Hall, Upper Saddle River, NJ), 2nd ed.
Lobry, C., 1970, SIAM J. Control **8**, 573.
Lohmiller, W., and J.-J. E. Slotine, 1998, Automatica **34**, 683.
Lo Iudice, F., F. Garofalo, and F. Sorrentino, 2015, Nat. Commun. **6**, 8349.
Lorenz, E. N., 1963, J. Atmos. Sci. **20**, 130.
Lovász, L., and M. D. Plummer, 2009, *Matching Theory* (American Mathematical Society, Providence, RI).
Luenberger, D., 1964, IEEE Trans. Mil. Electron. **8**, 74.
Luenberger, D., 1966, IEEE Trans. Autom. Control **11**, 190.
Luenberger, D., 1971, IEEE Trans. Autom. Control **16**, 596.
Luenberger, D. G., 1979, *Introduction to Dynamic Systems: Theory, Models, & Applications* (John Wiley & Sons, New York).
Lund, C., and M. Yannakakis, 1994, J. ACM **41**, 960.
Maxwell, J. C., 1867, Proc. R. Soc. London **16**, 270.
Mayeda, H., 1981, IEEE Trans. Autom. Control **26**, 795.
Mayeda, H., and T. Yamada, 1979, SIAM J. Control Optim. **17**, 123.
Menichetti, G., L. Dall'Asta, and G. Bianconi, 2014, Phys. Rev. Lett. **113**, 078701.
Menichetti, G., L. Dall'Asta, and G. Bianconi, 2016, Sci. Rep. **6**, 20706.
Mesbahi, M., 2005, IEEE Trans. Autom. Control **50**, 387.
Mesbahi, M., and M. Egerstedt, 2010, *Graph Theoretic Methods in Multiagent Networks* (Princeton University Press, Princeton, NJ).
Mézard, M., and G. Parisi, 2001, Eur. Phys. J. B **20**, 217.
Milnor, J. W., 2006, Scholarpedia **1**, 1815.
Milo, R., S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, 2002, Science **298**, 824.
Mirsky, H. P., A. C. Liu, D. K. Welsh, S. A. Kay, and F. J. Doyle, 2009, Proc. Natl. Acad. Sci. U.S.A. **106**, 11107.
Mochizuki, A., B. Fiedler, G. Kurosawa, and D. Saito, 2013, J. Theor. Biol. **335**, 130.
Molnár, F., N. Derzsy, E. Czabarka, L. Székely, B. K. Szymanski, and G. Korniss, 2014, Sci. Rep. **4**, 6308.
Moreau, L., 2005, IEEE Trans. Autom. Control **50**, 169.
Müller, F.-J., and A. Schuppert, 2011, Nature (London) **478**, E4.
Murota, K., 2009, *Matrices and Matroids for Systems Analysis, Algorithms and Combinatorics* (Springer, New York).
Nacher, J. C., and T. Akutsu, 2016, Methods **102**, 57.
Naidu, D. S., 2002, *Optimal Control Systems*, Electrical Engineering Series (CRC Press, Boca Ratan), 1st ed.
Nelson, E., 1967, *Tensor Analysis* (Princeton University Press, Princeton, NJ).
Nemhauser, G., L. Wolsey, and M. Fisher, 1978, Math. Program. **14**, 265.
Nepusz, T., and T. Vicsek, 2012, Nat. Phys. **8**, 568.
Newman, M. E. J., 2006, Proc. Natl. Acad. Sci. U.S.A. **103**, 8577.
Newman, M. E. J., S. H. Strogatz, and D. J. Watts, 2001, Phys. Rev. E **64**, 026118.
Nijmeijer, H., and A. van der Schaft, 1990, *Nonlinear Dynamical Control Systems* (Springer-Verlag, New York).
Nishikawa, T., and A. E. Motter, 2006, Phys. Rev. E **73**, 065106.
Nishikawa, T., A. E. Motter, Y.-C. Lai, and F. C. Hoppensteadt, 2003, Phys. Rev. Lett. **91**, 014101.
Olfati-Saber, R., 2006, IEEE Trans. Autom. Control **51**, 401.
Olfati-Saber, R., J. Fax, and R. Murray, 2007, Proc. IEEE **95**, 215.
Olshevsky, A., 2014, IEEE Transactions on Control of Network Systems **1**, 249.
Ott, E., C. Grebogi, and J. A. Yorke, 1990, Phys. Rev. Lett. **64**, 1196.
Palsson, B. O., 2006, *Systems Biology: Properties of Reconstructed Networks* (Cambridge University Press, Cambridge).
Pan, Y., and X. Li, 2014, PLoS One **9**, e94998.
Pastor-Satorras, R., and A. Vespignani, 2001, Phys. Rev. Lett. **86**, 3200.
Pastor-Satorras, R., and A. Vespignani, 2004, *Evolution and Structure of the Internet: A Statistical Physics Approach* (Cambridge University Press, Cambridge).
Pecora, L. M., and T. L. Carroll, 1998, Phys. Rev. Lett. **80**, 2109.
Peixoto, M., 1962, Topology **1**, 101.
Pequito, S., S. Kar, and A. Aguiar, 2013, in *American Control Conference*, pp. 6108–6113, doi:10.1109/ACC.2013.6580796.
Pequito, S., S. Kar, and A. P. Aguiar, 2015, IEEE Trans. Autom. Control **61**, 303.
Perin, R., T. K. Berger, and H. Markram, 2011, Proc. Natl. Acad. Sci. U.S.A. **108**, 5419.
Perseguers, S., 2010, Phys. Rev. A **81**, 012310.
Perseguers, S., J. I. Cirac, and J. Wehr, 2008, Phys. Rev. A **77**, 022308.
Perseguers, S., G. J. Lapeyre, D. Cavalcanti, M. Lewenstein, and A. Acín, 2013, Rep. Prog. Phys. **76**, 096001.
Perseguers, S., M. Lewenstein, A. Acín, and J. I. Cirac, 2010, Nat. Phys. **6**, 539.
Pham, Q.-C., and J.-J. Slotine, 2007, Neural Netw. **20**, 62.
Pimentel, J. A., M. Aldana, C. Huepe, and H. Larralde, 2008, Phys. Rev. E **77**, 061138.

Pohjanpalo, H., 1978, Math. Biosci. **41**, 21.
Poljak, S., 1990, IEEE Trans. Autom. Control **35**, 367.
Porfiri, M., and M. di Bernardo, 2008, Automatica **44**, 3100.
Pósfai, M., and P. Hövel, 2014, New J. Phys. **16**, 123055.
Pósfai, M., Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, 2013, Sci. Rep. **3**, 1067.
Prabakaran, S., J. Gunawardena, and E. Sontag, 2014, Biophys. J. **106**, 2720.
Pyragas, K., 1992, Phys. Lett. A **170**, 421.
Rahmani, A., M. Ji, M. Mesbahi, and M. Egerstedt, 2009, SIAM J. Control Optim. **48**, 162.
Rajapakse, I., M. Groudine, and M. Mesbahi, 2011, Proc. Natl. Acad. Sci. U.S.A. **108**, 17257.
Raz, R., and S. Safra, 1997, in *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing, STOC '97* (ACM, New York), pp. 475–484.
Régin, J.-C., 1994, in *Proceedings of the Twelfth National Conference on Artificial Intelligence (Vol. 1), AAAI '94* (American Association for Artificial Intelligence, Menlo Park, CA), pp. 362–367.
Reinschke, K. J., 1988, *Multivariable Control: A Graph-Theoretic Approach*, Lecture Notes in Control and Information Sciences (Springer, New York).
Reinschke, K. J., F. Svaricek, and H. D. Wend, 1992, in *Proceedings of the 31st IEEE Conference on Decision and Control* (IEEE, New York), Vol. 1, pp. 203–208, doi:10.1109/CDC.1992.371757.
Ren, W., and R. Beard, 2005, IEEE Trans. Autom. Control **50**, 655.
Reynolds, C. W., 1987, SIGGRAPH Computer Graphics **21**, 25.
Ritter, S., C. Nölleke, C. Hahn, A. Reiserer, A. Neuzner, M. Uphoff, M. Mücke, E. Figueroa, J. Bochmann, and G. Rempe, 2012, Nature (London) **484**, 195.
Rugh, W. J., 1981, *Nonlinear system theory: the Volterra/Wiener approach* (Johns Hopkins University Press, Baltimore).
Rugh, W. J., 1993, *Linear system theory* (Prentice Hall, Englewood Cliffs, NJ).
Russo, G., and M. Di Bernardo, 2009, IEEE Trans. Circuits Syst. II **56**, 177.
Ruths, J., and D. Ruths, 2014, Science **343**, 1373.
Saccomani, M. P., S. Audoly, and L. D'Angio, 2003, Automatica **39**, 619.
Sahni, N., *et al.*, 2015, Cell **161**, 647.
Sass, B., and Z. Toroczkai, 1996, J. Phys. A **29**, 3545.
Sastry, S., 1999, *Nonlinear Systems: Analysis, Stability, and Control* (Springer, New York).
Schellenberger, J., J. O. Park, T. M. Conrad, and B. Palsson, 2010, BMC Bioinf. **11**, 213.
Sedoglavic, A., 2002, J. Symb. Comput. **33**, 735.
Serre, J.-P., 1992, *Lie Algebras and Lie Groups* (Springer-Verlag, Berlin).
Shandilya, S. G., and M. Timme, 2011, New J. Phys. **13**, 013004.
Shields, R. W., and J. B. Pearson, 1976, IEEE Trans. Autom. Control **21**, 203.
Shinbrot, T., C. Grebogi, E. Ott, and J. A. Yorke, 1993, Nature (London) **363**, 411.
Siddhartha, J., and J. H. van Schuppen, 2001, Modelling and Control of Cell Reaction Networks, Tech. Rep. No. PNA-R0116 (Centrum Wiskunde & Informatica).
Slack, J. M. W., 2002, Nat. Rev. Genet. **3**, 889.
Slotine, J.-J., and W. Li, 1991, *Applied Nonlinear Control* (Prentice-Hall, Englewood Cliffs, NJ).
Slotine, J.-J. E., W. Wang, and K. E. Rifai, 2004, "Contraction analysis of synchronization in networks of nonlinearly coupled oscillators" (unpublished).
Smale, S., 1961, Ann. Math. **74**, 199.
Smale, S., 1967, Bull. Am. Math. Soc. **73**, 747.
Socolar, J. E. S., D. W. Sukow, and D. J. Gauthier, 1994, Phys. Rev. E **50**, 3245.
Song, Q., and J. Cao, 2010, IEEE Trans. Circuits Syst. I **57**, 672.
Sontag, E., 2008, Essays Biochem. **45**, 161.
Sontag, E. D., 1988, SIAM J. Control Optim. **26**, 1106.
Sontag, E. D., 1998, *Mathematical Control Theory: Deterministic Finite Dimensional Systems*, Textbooks in Applied Mathematics (Springer, New York).
Sontag, E. D., 2004, Syst. Biol. **1**, 9.
Sontag, E. D., and Y. Wang, 1991, in *Proceedings of the 90th IEEE Conference on Decision and Control* (IEEE Press, New York), pp. 720–725, doi:10.1109/CDC.1991.261406.
Sorrentino, F., M. di Bernardo, F. Garofalo, and G. Chen, 2007, Phys. Rev. E **75**, 046103.
Stépán, G., and L. Kollár, 2000, Math. Comput. Modell. **31**, 199, Proceedings of the Conference on Dynamical Systems in Biology and Medicine.
Strogatz, S. H., 1994, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering* (Westview Press, Cambridge, MA).
Su, H., X. Wang, and Z. Lin, 2009, IEEE Trans. Autom. Control **54**, 293.
Sukow, D. W., M. E. Bleich, D. J. Gauthier, and J. E. S. Socolar, 1997, Chaos **7**, 560.
Summers, T. H., F. Cortesi, and J. Lygeros, 2016, IEEE. Trans. Control Netw. Syst. **3**, 91.
Summers, T. H., and J. Lygeros, 2014, in *Proceedings of the 19th IFAC World Congress*, Vol. 47, pp. 3784–3789, doi:10.3182/20140824-6-ZA-1003.00226.
Sun, J., and A. E. Motter, 2013, Phys. Rev. Lett. **110**, 208701.
Sussmann, H., 1987, SIAM J. Control Optim. **25**, 158.
Sussmann, H., and V. Jurdjevic, 1972, J. Diff. Equ. **12**, 95.
Tabareau, N., J.-J. Slotine, and Q.-C. Pham, 2010, PLoS Comput. Biol. **6**, e1000637.
Tanner, H. G., 2004, in *43rd IEEE Conference on Decision and Control, CDC.* (IEEE, New York), Vol. 3, pp. 2467–2472, doi:10.1109/CDC.2004.1428782.
Tass, P., M. G. Rosenblum, J. Weule, J. Kurths, A. Pikovsky, J. Volkmann, A. Schnitzler, and H.-J. Freund, 1998, Phys. Rev. Lett. **81**, 3291.
Tegnér, J., M. S. Yeung, J. Hasty, and J. J. Collins, 2003, Proc. Natl. Acad. Sci. U.S.A. **100**, 5944.
Tero, A., S. Takagi, T. Saigusa, and K. Ito, 2010, Science **327**, 439.
Timme, M., and J. Casadiego, 2014, J. Phys. A **47**, 343001.
Toroczkai, Z., 1994, Phys. Lett. A **190**, 71.
Toroczkai, Z., and K. E. Bassler, 2004, Nature (London) **428**, 716.
Vicsek, T., A. Czirk, E. Ben-Jacob, I. Cohen, and O. Shochet, 1995, Phys. Rev. Lett. **75**, 1226.
Vicsek, T., and A. Zafeiris, 2012, Phys. Rep. **517**, 71.
Vinayagam, A., *et al.*, 2016, Proc. Natl. Acad. Sci. U.S.A. **113**, 4976.
Volterra, V., 1931, *Lecons sur la theorie mathematique de la lutte pour la vie* (Gauthier-Villars, Paris).
Waddington, C. H., and H. Kacser, 1957, *The Strategy of the Genes: A Discussion of Some Aspects of Theoretical Biology* (Allen & Unwin, London).
Wang, J., K. Zhang, L. Xu, and E. Wang, 2011, Proc. Natl. Acad. Sci. U.S.A. **108**, 8257.
Wang, L., G. Chen, X. Wang, and W. K. S. Tang, 2016, Automatica **69**, 405.
Wang, L., H. P. Dai, H. Dong, Y. Y. Cao, and Y. X. Sun, 2008, Eur. Phys. J. B **61**, 335.

Wang, P., C. Song, H. Zhang, Z. Wu, X.-J. Tian, and J. Xing, 2014, Interface Focus **4**, 20130068.
Wang, W., and J.-J. Slotine, 2005, Biol. Cybern. **92**, 38.
Wang, W., and J.-J. Slotine, 2006, IEEE Trans. Autom. Control **51**, 1156.
Wang, W.-X., X. Ni, Y.-C. Lai, and C. Grebogi, 2012, Phys. Rev. E **85**, 026115.
Wang, X. F., and G. Chen, 2002a, Physica A (Amsterdam) **310**, 521.
Wang, X. F., and G. Chen, 2002b, Int. J. Bifurcation Chaos **12**, 187.
Wang, Z., L. Huang, Y. Wang, and Y. Zuo, 2010, Commun. Nonlinear Sci. Numer. Simul. **15**, 4202.
Watts, D. J., and S. H. Strogatz, 1998, Nature (London) **393**, 440.
Whalen, A. J., S. N. Brennan, T. D. Sauer, and S. J. Schiff, 2015, Phys. Rev. X **5**, 011005.
Wu, C., and L. Chua, 1994, Int. J. Bifurcation Chaos **04**, 979.
Wu, C., and L. Chua, 1995a, IEEE Trans. Circuits Syst. I **42**, 494.
Wu, C., and L. Chua, 1995b, IEEE Trans. Circuits Syst. I **42**, 775.
Wu, C., and L. Chua, 1995c, IEEE Trans. Circuits Syst. I **42**, 430.
Wuchty, S., 2014, Proc. Natl. Acad. Sci. U.S.A. **111**, 7156.
Xiao, Y.-D., S.-Y. Lao, L.-L. Hou, and L. Bai, 2014, Phys. Rev. E **90**, 042804.
Xie, G., and L. Wang, 2003, Syst. Control Lett. **48**, 135.
Xie, G., D. Zheng, and L. Wang, 2002, IEEE Trans. Autom. Control **47**, 1401.
Yamada, T., and L. R. Foulds, 1990, Networks **20**, 427.
Yan, G., J. Ren, Y.-C. Lai, C.-H. Lai, and B. Li, 2012, Phys. Rev. Lett. **108**, 218703.
Yan, G., G. Tsekenis, B. Barzel, J.-J. Slotine, Y.-Y. Liu, and A.-L. Barabási, 2015, Nat. Phys. **11**, 779.
Yang, Y., J. Wang, and A. Motter, 2012, Phys. Rev. Lett. **109**, 258701.
Yu, D., 2010, Automatica **46**, 2035.
Yu, D., and U. Parlitz, 2010, Phys. Rev. E **82**, 026108.
Yu, D., M. Righero, and L. Kocarev, 2006, Phys. Rev. Lett. **97**, 188701.
Yu, W., G. Chen, and J. Lü, 2009, Automatica **45**, 429.
Yu, W., G. Chen, J. Lü, and J. Kurths, 2013, SIAM J. Control Optim. **51**, 1395.
Yuan, Z., C. Zhao, Z. Di, W.-X. Wang, and Y.-C. Lai, 2013, Nat. Commun. **4**, 2447.
Yuan, Z., C. Zhao, W.-X. Wang, Z. Di, and Y.-C. Lai, 2014, New J. Phys. **16**, 103036.
Zañudo, J. G. T., and R. Albert, 2015, PLoS Comput. Biol. **11**, e1004193.
Zdeborová, L., and M. Mézard, 2006, J. Stat. Mech. 05, P05003.
Zhang, Y., A. Garas, and F. Schweitzer, 2016, Phys. Rev. E **93**, 012309.
Zhao, C., W.-X. Wang, Y.-Y. Liu, and J.-J. Slotine, 2015, Sci. Rep. **5**, 8422.
Zhao, J.-H., Y. Habibulla, and H.-J. Zhou, 2015, J. Stat. Phys. **159**, 1154.
Zhou, H., and Z. Ou-Yang, 2003, "Maximum matching on random graphs," arXiv:cond-mat/0309348v1.
Zhou, J., J.-A. Lu, and J. Lü, 2008, Automatica **44**, 996.
Zhou, T., 2015, Automatica **52**, 63.
Zou, Y., and G. Chen, 2008, Europhys. Lett. **84**, 58005.