Control for a synchronization-desynchronization switch

Zhiwei He,[1,2] Xingang Wang,[3] Guo-Yong Zhang,[1,4] and Meng Zhan[1,*]

[1]*Wuhan Center for Magnetic Resonance, State Key Laboratory of Magnetic Resonance and Atomic and Molecular Physics,*
Wuhan Institute of Physics and Mathematics, Chinese Academy of Sciences, Wuhan 430071, China
[2]*University of the Chinese Academy of Sciences, Beijing 100049, China*
[3]*School of Physics and Information Technology, Shaanxi Normal University, Xi'an 710062, China*
[4]*College of Computer Science and Technology, Hubei Normal University, Huangshi 435002, China*

How to freely enhance or suppress synchronization of networked dynamical systems is of great importance in many disciplines. A unified precise control method for a synchronization-desynchronization switch, called the *pull-push control method*, is suggested. Namely, synchronization can be achieved when the original systems are desynchronous by pulling (or protecting) one node or a certain subset of nodes, whereas desynchronization can be accomplished when the systems are already synchronous by pushing (or kicking) one node or a certain subset of nodes. With this method, the controlled nodes should be chosen by the generalized eigenvector centrality of the critical synchronization mode of the Laplacian matrix. Compared with existing control methods for synchronization, it displays high efficiency, flexibility, and precision as well.

I. INTRODUCTION

Synchronization is omnipresent in nature and has been widely used in many manmade systems [1–9]. It has attracted many scientists and engineers from various fields to study and utilize it with great interest. During the past decade, the research interest has been shifted from the studies of the structures of complex networks, such as the discovery of small-world networks, scale-free networks, and several key properties of complex networks with complicated topology, to the studies of the dynamics of complex networks, including synchronization of coupled networked nonlinear systems [3].

In many cases, synchronization has been deemed to be very useful and necessary. Such cases include synchronous flashing by fireflies [10], sensory processing in cortical networks [11], and laser synchronization [12,13]. According to previous studies, the topology of networks plays an important role in synchronization, and the degree-heterogeneous networks are usually difficult to synchronize [14–16]. Many methods have been proposed to enhance network synchronizability, such as (i) slightly modifying the network structure, including removing nodes with maximal betweenness [17], dividing hub nodes [18], deleting overload edges [19], rewiring network edges [20,21], and shortening the average distance [22]; or (ii) properly introducing weight and directionality to each link (edge) [23–26]. In addition, the so-called pinning control, which usually refers to a local feedback method by inputting an external signal and stabilizing the network to a synchronous state (or a homogeneous steady state), has been verified to be very effective [27–31]. The original method has also been extended to other methods such as the adaptive pining control method [32,33] and the node-to-node pining control method [34].

On the other hand, synchronization may be harmful and undesirable in some other situations. For instance, the onset of abnormal synchronization may lead to congestion in traffic networks [35], the collapse of constructions [35], events such as the Millennium Bridge vibrations [36], epileptic seizures [37], and Parkinson's disease [38]. Several methods, such as fighting a charismatic leader and a set of political adversaries (contrarians) [39], have been proposed to suppress (or avoid) undesired synchronization. Despite all this progress, it remains unknown how to control networked dynamical systems and achieve a low-cost switch between synchronization and desynchronization under a unified framework.

For this problem, the key point is to precisely find the most effective node as the controller [40,41]. Some ranking strategies have been developed for node importance based solely upon stationary topology, including degree centrality, closeness centrality [42,43], betweenness centrality [44], and eigenvector centrality [45]. For the synchronization control problem, the degree-centrality-based strategy and the betweenness-centrality-based strategy have been widely verified as being helpful for enhancing network synchronizability [29–31]. To the best of our knowledge, although the eigenvector centrality has been widely used in many fields for ranking, such as Google's "PageRank" algorithm [46], analyzing connectivity patterns in functional magnetic resonance imaging (fMRI) data of the human brain [47], and finding a community structure in networks [48], it has not yet been used for any synchronization control problem.

Very recently, some of the authors and co-workers studied the desynchronous pattern of networked dynamics in detail [49,50]. We took a closer look at the nonzero time-averaged synchronization errors for each node and found that they are linearly related to the absolute value of the eigenvector element of the Laplacian matrix with the corresponding critical mode of the coupled systems. This indicates that although the desynchronous pattern is highly irregular and intermittent as time evolves, in the long-time period the average deviation away from the synchronization state for each node shows a

---

*Author to whom all correspondence should be addressed: zhanmeng@wipm.ac.cn

regular behavior, i.e., a linear dependence on the corresponding eigenvector element. Therefore, the instability strength of the nodes is exactly characterized by the strength of the eigenvector centrality of the Laplacian matrix. Here it is notable that the traditional eigenvector centrality relies on the eigenvector corresponding to the largest eigenvalue of the adjacency matrix, and it can be guaranteed to be non-negative by the Perron-Frobenius theorem. Here we use a slightly different form of the eigenvector centrality, relying on the absolute value of the eigenvector elements from the Laplacian matrix. The other key observation is that all of these first several unstable nodes are isolated and unconnected directly on the network.

In this paper, just relying on these interesting findings, we would like to study further the synchronization control problem, i.e., how to realize synchronization when the original systems are desynchronous and to realize desynchronization when the systems are already synchronous. The idea is simple: if the system stays at around the synchronization-desynchronization parameter region, we may pull the most unstable node with the aid of its neighbors to realize the synchronization of the whole system, and push the exact same node (the least stable node now) to destroy the synchronization. As a result, we may accomplish the synchronization-desynchronization switch precisely and efficiently with the help of the internal connection of the nodes.

II. MODEL AND METHOD

Consider the following paradigmatic model of diffusively coupled map lattices on complex networks [49]:

$$x_i(n+1) = F(x_i(n)) + \frac{\varepsilon}{d_i} \sum_{j=1}^{N} A_{ij}[H(x_j(n)) - H(x_i(n))],$$

$$i,j = 1, \dots, N, \tag{1}$$

where N represents the size of the network and ε is the uniform coupling strength. A is the adjacency matrix, and the elements $A_{ij} = A_{ji} = 1$ if nodes i and j are connected by a link, and $A_{ij} = A_{ji} = 0$ otherwise; self-connection is not allowed: $A_{ii} = 0$. $d_i = \sum_{j=1}^{N} A_{ij}$ denotes the degree of the ith node. Without losing generality, we employ the chaotic logistic map, $F(x) = 4x(1-x)$, as the local node dynamics, and using $H(x) = F(x)$ as the coupling function between them. According to the analytic method of the master stability function (MSF) for the synchronous manifold [51,52], the network is synchronizable within the regime $\varepsilon \in (\varepsilon_1, \varepsilon_2)$ with $\varepsilon_1 = 0.5/\lambda_2$ and $\varepsilon_2 = 1.5/\lambda_N$, where λ_2 and λ_N are the corresponding second and largest eigenvalues of the Laplacian matrix L, where

$$L_{ij} = \begin{cases} -\frac{1}{d_i} A_{ij} & \text{if } i \neq j, \\ 1 & \text{if } i = j, \end{cases} \tag{2}$$

and $0 = \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_N$.

When the coupling strength falls outside of the synchronization parameter regime, the network is desynchronized and synchronization errors of all nodes, $\delta x_i'(n) = |x_i(n) - \bar{x}(n)|$,

where $\bar{x}(n) = \sum_{i=1}^{N} x_i(n)/N$, should be nonzero. In this situation, we should intentionally protect the most unstable node, whose synchronization error is the largest. A proper strategy is pulling its amplitude with the average level of its neighbors, whose synchronization errors are not very large, namely

$$x_i(n+1) = \frac{1}{d_i} \sum_{j=1}^{N} A_{ij} x_j(n+1) \quad \text{if } i \in J, \tag{3}$$

where J is the set of nodes that should be controlled for every time step. We assume that all nodes can be chosen as controls. As we will see, such a slight management could have a significant impact on the synchronization stability of the whole system.

On the other hand, consider the other side of the coupling parameter when the system is already synchronized, i.e., the coupling strength within the region $\varepsilon \in (\varepsilon_1, \varepsilon_2)$. Although all of the nodes are now synchronous, the synchronization stability of these nodes is heterogeneous, and their strength can be easily signaled by the eigenvector centrality of the Laplacian matrix. In this situation, we should intentionally push or kick the same worst node, which is nevertheless still stable, away from the synchronization state. Namely, we add a sufficiently large perturbation δ on it,

$$x_i(n+1) = x_i(n+1) + \delta \quad \text{if } i \in J, \tag{4}$$

where $0 < \delta < 0.5$ if $x_i(n+1) \leqslant 0.5$ and $-0.5 < \delta < 0$ if $x_i(n+1) > 0.5$ to make sure that $0 < x_i(n+1) < 1$ for the system stability.

In this paper, for comparison, the set J has been tested for several different centralities, such as the eigenvector centrality of the Laplacian matrix, the degree centrality, the closeness centrality, and the betweenness centrality. Here the eigenvector centrality of a node i is defined as $\text{ec}(i) = |e_{2,i}|$ or $|e_{N,i}|$, where $\mathbf{e}_2 = (e_{2,1}, e_{2,2}, \dots, e_{2,N})$ and $\mathbf{e}_N = (e_{N,1}, e_{N,2}, \dots, e_{N,N})$ are the eigenvectors of the Laplacian matrix L associated with the corresponding critical modes, λ_2 and λ_N, respectively. The closeness centrality of a node i is defined as $\text{cc}(i) = \frac{N-1}{\sum_{j \in N, j \neq i} l_{ij}}$, where l_{ij} is the shortest path length (distance) between nodes i and j. The betweenness centrality of node i is defined as $\text{bc}(i) = \sum_{i \neq j \neq k} \frac{\sigma_{jk}(i)}{\sigma_{jk}}$, where σ_{jk} is the total number of shortest paths from node j to k, and $\sigma_{jk}(i)$ is the number of shortest paths from node j to k going through node i.

Meanwhile, we consider three typical models of a complex network: the Barabási-Albert (BA) scale-free network [53], the small-world network [54], and the Erdös-Rényi (ER) random network [55]. All these networks are generated with the standard algorithms; the network size $N = 100$ considered is fixed. The node or the network is believed to be synchronous when the time-averaged synchronization error of node i, $\delta x_{i,T} = \sum_{n=1}^{T} \delta x_i'(n)/T$, with $T = 10^5$, or the averaged synchronization error of the system, $\delta x_{\text{net},T} = \sum_{i=1}^{N} \delta x_{i,T}/N$, is smaller than 10^{-8} after a sufficiently transient time $n = 4 \times 10^5$.
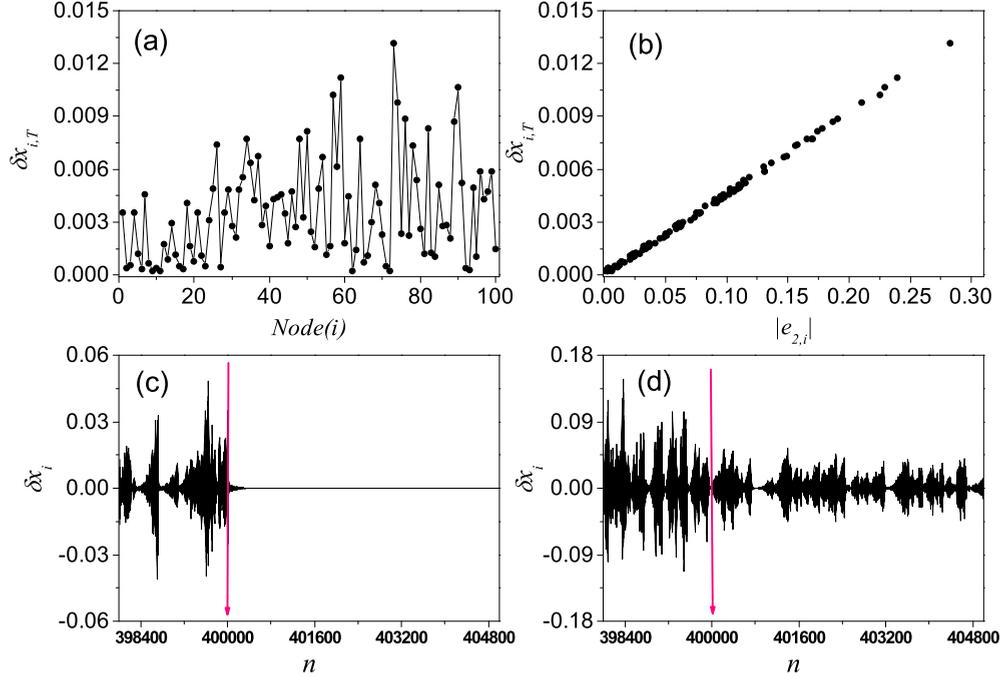
FIG. 1. (Color online) Controlling results for switch from desynchronization to synchronization. (a) The time-averaged synchronization error of all nodes, $\delta x_{i,T}$, at the coupling strength $\varepsilon = 0.86 < \varepsilon_1$ ($\varepsilon_1 \simeq 0.871$) under the desynchronization state. (b) $\delta x_{i,T}$ as a function of the eigenvector element, $|e_{2,i}|$, indicative of a monotonic linear relation. The three leading nodes (73,59,90) with the largest $\delta x_{i,T}$ correspond to the three largest eigenvector element. (c) and (d) The time evolution of the trajectory error $\delta x_i(n) = x_i(n) - \bar{x}(n)$ for $\varepsilon = 0.86$ and 0.84, respectively, with the pull control added on the 73rd node after $n = 4 \times 10^5$. Here a BA scale-free network with $N = 100$ and $\langle k \rangle = 16$ is exemplified.

III. NUMERICAL RESULTS

A. Pull control for the desynchronization-synchronization switch

First let us test whether the desynchronized system can be switched to synchronization by controlling only one node with our pull control method. As an example, a scale-free network with $N = 100$ and $\langle k \rangle = 16$ is treated, whose synchronization regime of the networked system stays at $(\varepsilon_1, \varepsilon_2) \simeq (0.871, 1.040)$. When $\varepsilon = 0.86 < \varepsilon_1$ is chosen without controlling, the time-averaged synchronization error for each node i, $\delta x_{i,T}$, is shown in Fig. 1(a), and it is also plotted as a function of eigenvector element $|e_{2,i}|$ in Fig. 1(b). Now λ_2 corresponds to the critical mode for synchronization. From these plots, we can see that the networked system is truly desynchronized but shows different strengths for different nodes ($\delta x_{i,T} > 0$), and their synchronization errors are linearly correlated with the absolute values of the corresponding eigenvector element ($\delta x_{i,T} \propto |e_{2,i}|$). Clearly in Fig. 1(a) the 73rd node is the most unstable. This reconfirms the findings of our recent work [49]. Figure 1(c) shows the time evolution of the trajectory error of the system, $\delta x_i(n) = x_i(n) - \bar{x}(n)$, when the pull control is added to the 73rd node [namely $J = \{73\}$ in Eq. (4)] after $n = 4 \times 10^5$, where we can see that δx_i quickly damps and vanishes after a very short transient, indicative of a successful control by using only one single node. However, when the coupling strength is farther away from the critical parameter ε_1, say, e.g., $\varepsilon = 0.84$, the pull control method does not work either, as shown in Fig. 1(d),

although the amplitude of desynchronous behavior has been greatly repressed (comparing the different amplitudes before and after the controller is added). This indicates that our control method also has its own application limitation.

To quantitatively characterize the application region of the pull control method, in Fig. 2(a) the averaged synchronization errors, $\delta x_{\mathrm{net},T}$, in both the uncontrolled system and the system adding control on the 73rd node are compared. Both curves decrease monotonically and vanish at certain critical values: ε_1 ($\varepsilon_1 \simeq 0.871$) for the uncontrolled system and $\varepsilon_c \approx 0.854 (\varepsilon_c < \varepsilon_1)$ for the controlled system. This comparison clearly demonstrates that the synchronization region has been extended to include $\varepsilon \in (\varepsilon_c, \varepsilon_1)$ for the original desynchronous parameter region. We may call $\varepsilon \in (\varepsilon_c, \varepsilon_1)$ the controllable region.

In Fig. 2(b), we further test the efficiency of the pull method by calculating the critical values ε_c for controlling on different nodes. We also calculate the degree centrality (d_i), the closeness centrality [cc(i)], the betweenness centrality [bc(i)], and the eigenvector centrality [ec(i)] for all nodes of the network, and we mark their corresponding nodes with the largest values in Fig. 2(b), e.g., the 17th node for d_i, the 17th node for cc(i), the 16th node for bc(i), and the 73rd node for ec(i). It can be seen that the smallest value ε_c for the largest controllable region corresponds to the node with the largest eigenvector centrality. Furthermore, the critical coupling values in Fig. 2(c) are rearranged in decreasing order for the value of $|e_{2,i}|$ from large to small, where the node with a larger value of $|e_{2,i}|$ corresponds roughly to smaller ε_c. This
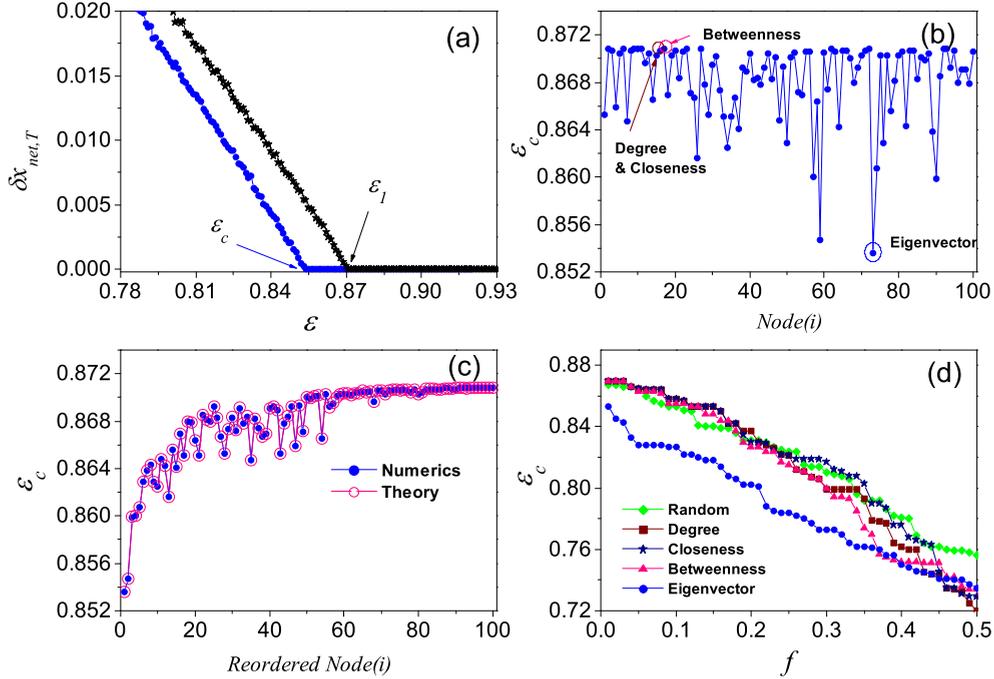
FIG. 2. (Color online) (a) The averaged synchronization error $\delta x_{\text{net},T}$ vs the coupling strength ε for the uncontrolled system and the system with the node 73 controlled, showing an extended coupling parameter region for the synchronization from ε_1 to ε_c. (b) The critical value ε_c for each node as the controller. The nodes with the largest degree (17th node), the largest closeness centrality (17th node), the largest betweenness centrality (16th node), and the largest value of $|e_{2,i}|$ (73rd node) are marked. (c) The numerical (blue dotted line) and theoretical (pink open circles) results of the critical value ε_c after all nodes are rearranged according to their eigenvector element, $|e_{2,i}|$, from large to small, showing a roughly monotonic relation. (d) The critical value ε_c vs the ratio f for different control strategies, including the random strategy (green diamond line), the degree centrality strategy (wine square line), the closeness centrality strategy (navy star line), the betweenness centrality strategy (pink triangle line), and the eigenvector centrality strategy (blue dotted line). Among all these strategies, the eigenvector centrality method exhibits the highest efficiency.

confirms that the network should be easier to be controlled to synchronization if the node with a larger synchronization error is targeted.

In addition, the efficiency of the pull method has also been tested when several (not only one) nodes are controlled. Figure 2(d) plots the critical value ε_c versus the ratio of nodes f that are controlled, for different control strategies, including the random, degree-centrality, closeness-centrality, betweenness-centrality, and eigenvector-centrality ones. From these curves, we know that (i) no matter which strategy is chosen, the networked system can always be synchronized by controlling more nodes; (ii) the eigenvector-centrality-based strategy is always the most effective one among all these strategies, as the critical value ε_c is always the smallest ($f < 0.5$); and (iii) for all other strategies, we cannot see much of a difference.

To proceed further, we test the efficiency of the pull method on the other side of the coupling parameter region $\varepsilon > \varepsilon_2$. In Fig. 3, we perform similar simulations to those in Fig. 2. Here the largest controllable region has been found to be extended to the critical coupling strength ε_c when the node with the largest value of $|e_{N,i}|$ is chosen, as shown in Figs. 3(a)–3(c). Again the eigenvector-centrality-based strategy, relying on $ec(i) = |e_{N,i}|$ here, is still the most effective strategy, even when nodes with the centrality degree in descending order are selected successively, as shown in Fig. 3(d).

To test the generality of this control method for other kinds of complex networks, we perform extensive simulations on the small-world network and the ER random network, and we find that the qualitative results are nearly the same. See, for example, the small-world network in Figs. 4(a) and 4(b) and the ER network in Figs. 4(c) and 4(d), exhibiting only a little mismatch in Fig. 4(c), where controlling on the node with the highest eigenvector centrality does not exactly induce the largest controllable region. A possible reason for this is that because the random network with comparatively homogeneous degrees is now considered, the first several worst nodes may not be unconnected completely.

Consequently, based on the above extensive simulations, we may conclude that the pull control method is highly effective, and it is capable of switching the desynchronized network to a synchronized one with the help of the system self-organized behavior. Next, we will examine the results for the synchronization-desynchronization switch.

B. Push control for the synchronization-desynchronization switch

We still start from the same BA scale-free network used in Fig. 1, but we consider the case for the coupling strength $\varepsilon = 0.93$ being a little larger than ε_1 ($\varepsilon_1 \simeq 0.871$). In the absence of control, the system is synchronous. We set $\delta = 0.3$ if $x_i(n + 1) \leqslant 0.5$ and $\delta = -0.3$ if $x_i(n + 1) > 0.5$ in Eq. (4), and we
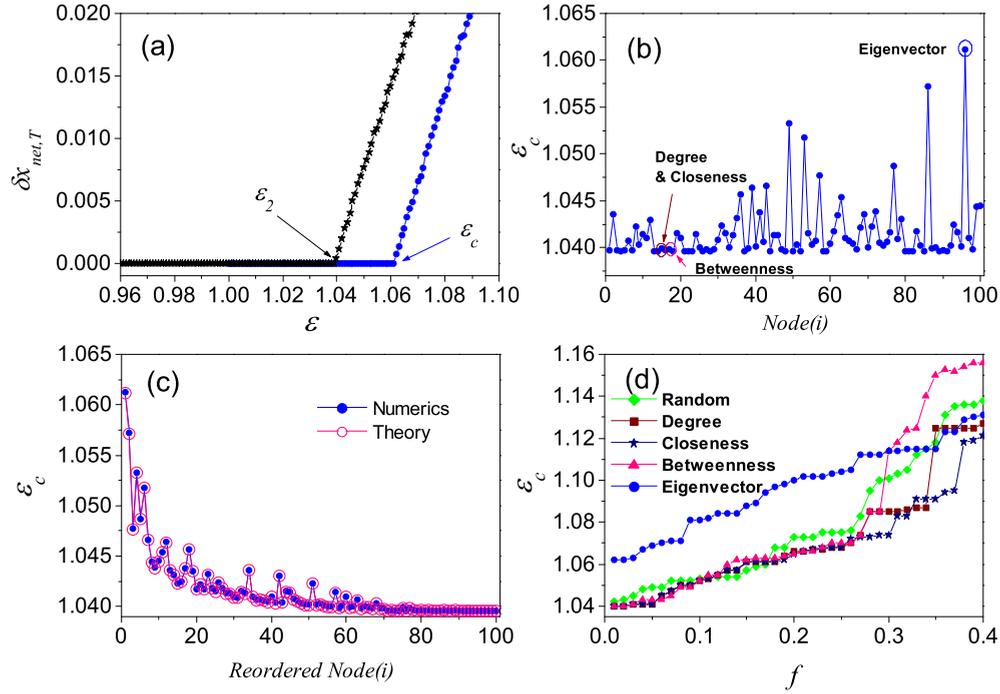
FIG. 3. (Color online) Similar to Fig. 2 but for the other side of the parameter region, $\varepsilon \gtrsim \varepsilon_2$. The network used here is the same as the one used in Fig. 2 but with the critical eigenvector element $|e_{N,i}|$ instead.

show the time evolutions of the trajectory errors of all nodes, $\delta x_i(n) = x_i(n) - \bar{x}(n)$, in Fig. 5(a). Before $n = 4 \times 10^5$, the system is free, exhibiting perfect synchronization. When one node is pushed at the time $n = 4 \times 10^5$, the system becomes desynchronous immediately and damps gradually back to the

synchronous state for a sufficiently long time. Hence for a better control efficiency, we would choose the node having the longer transient time.

To find out which node has the longest transient period, we calculate the transient periods for different nodes for different
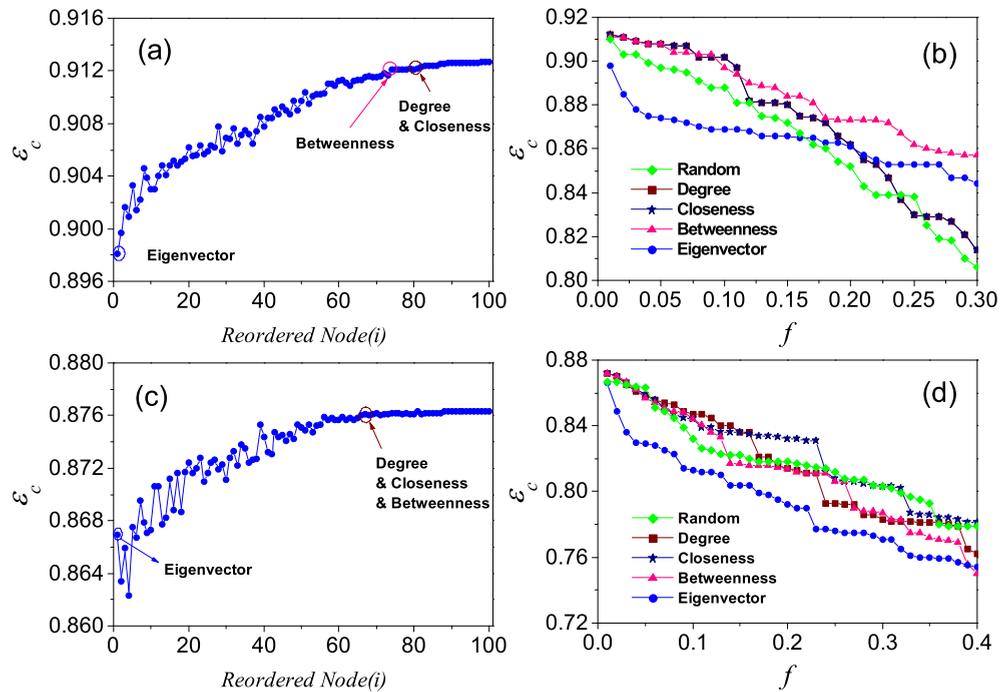
FIG. 4. (Color online) (a) and (b) Similar to Figs. 2(c) and 2(d) but for a small world network with $N = 100$, $\langle k \rangle = 40$, and $p = 0.3$; the synchronization regime is within $(\varepsilon_1, \varepsilon_2) \simeq (0.913, 1.224)$. (c) and (d) An ER random network with $N = 100$ and $\langle k \rangle = 16$ is considered; the synchronization regime is within $(\varepsilon_1, \varepsilon_2) \simeq (0.876, 1.042)$. Qualitative results are the same for different types of complex networks.
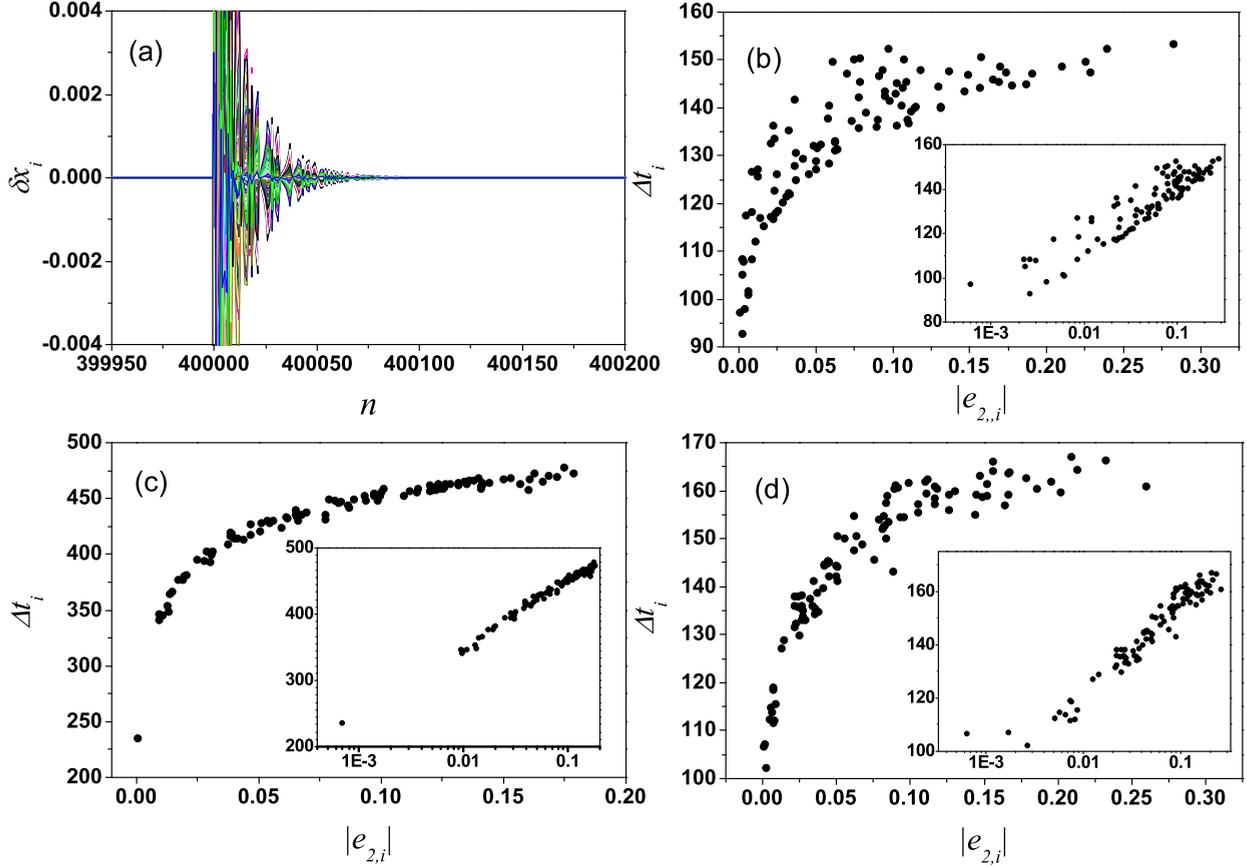
FIG. 5. (Color online) Controlling results for the switch from synchronization to desynchronization. (a) The time evolution of the trajectory error δx_i with the 73rd node controlled after $n = 4 \times 10^5$; $\varepsilon = 0.93 < \varepsilon_2$ ($\varepsilon_2 \simeq 1.040$). (b), (c), and (d) The averaged transient period, $\triangle t_i$, over 1000-time realizations as a function of the eigenvector element, $|e_{2,i}|$, in a BA scale-free network, a small-world network, and an ER random network, respectively. The insets of each subfigure with a semilogarithmic plot indicate the exponential relation $e^{\triangle t_i} \propto |e_{2,i}|$.

types of networks. In Figs. 5(b)–5(d), the transient periods for the ith node controlled, $\triangle t_i$, have been averaged over 1000 realizations with different initial conditions. Their dependence on the eigenvector element, $|e_{2,i}|$, in the BA scale-free network, the small-world network, and the ER random network are shown in Figs. 5(b), 5(c), and 5(d), respectively. Based on these plots, one can easily see that a larger value of $|e_{2,i}|$ corresponds roughly to a longer transient period. With the semilogarithmic plots in the insets of these subfigures, we even find an exponential relationship between $\triangle t_i$ and $|e_{2,i}|$, i.e., $e^{\triangle t_i} \propto |e_{2,i}|$. Thus as with the pull control method, the push control method would be more effective by controlling the node with a larger value of $|e_{2,i}|$ (or larger eigenvector centrality [ec(i) = $|e_{2,i}|$]). The underlying mechanism is the same.

In addition, we have tested the case in which the coupling strength ε is a little smaller than ε_2 for different kinds of networks, and we found similar results (not shown here). We have also observed that if the node is pushed periodically with the control period less than the transient period, then the system can stay at the desynchronization state all the time. Thus we can conclude that the synchronization-desynchronization switch activated by the simple push control method is also very efficient.

C. Some other systems

So far, we have mainly presented the numerical results for the coupled logistic map systems. In this subsection, we will give more examples to justify the validity and efficiency of our control method, including the Rulkov map system [56] and the time-continuous Rössler system [57].

The Rulkov map system is important in neuroscience as it can properly describe the neuron's irregular bursting behavior. The coupled Rulkov map system on the complex networks can be described as follows:

$$x_i(n + 1) = F(x_i(n), y_i(n)) + \frac{\varepsilon}{d_i} \sum_{j=1}^{N} A_{ij}[H(x_j(n), y_j(n))$$

$$- H(x_i(n), y_i(n))],$$

$$y_i(n + 1) = y_i(n) - \sigma x_i(n) - \beta, \qquad (5)$$

where $F(x, y) = \frac{\alpha}{1+x^2} + y$, and without losing generality, we use $H(x, y) = F(x, y)$ as the coupling function between nodes. Here $x_i(n)$ and $y_i(n)$ represent the fast and slow dynamical variables of the ith node, respectively. The slow evolution of $y_i(n)$ appears due to the small values of the two positive parameters σ and β, say, e.g., $\sigma = \beta = 0.001$. If $\alpha > 4.0$, the
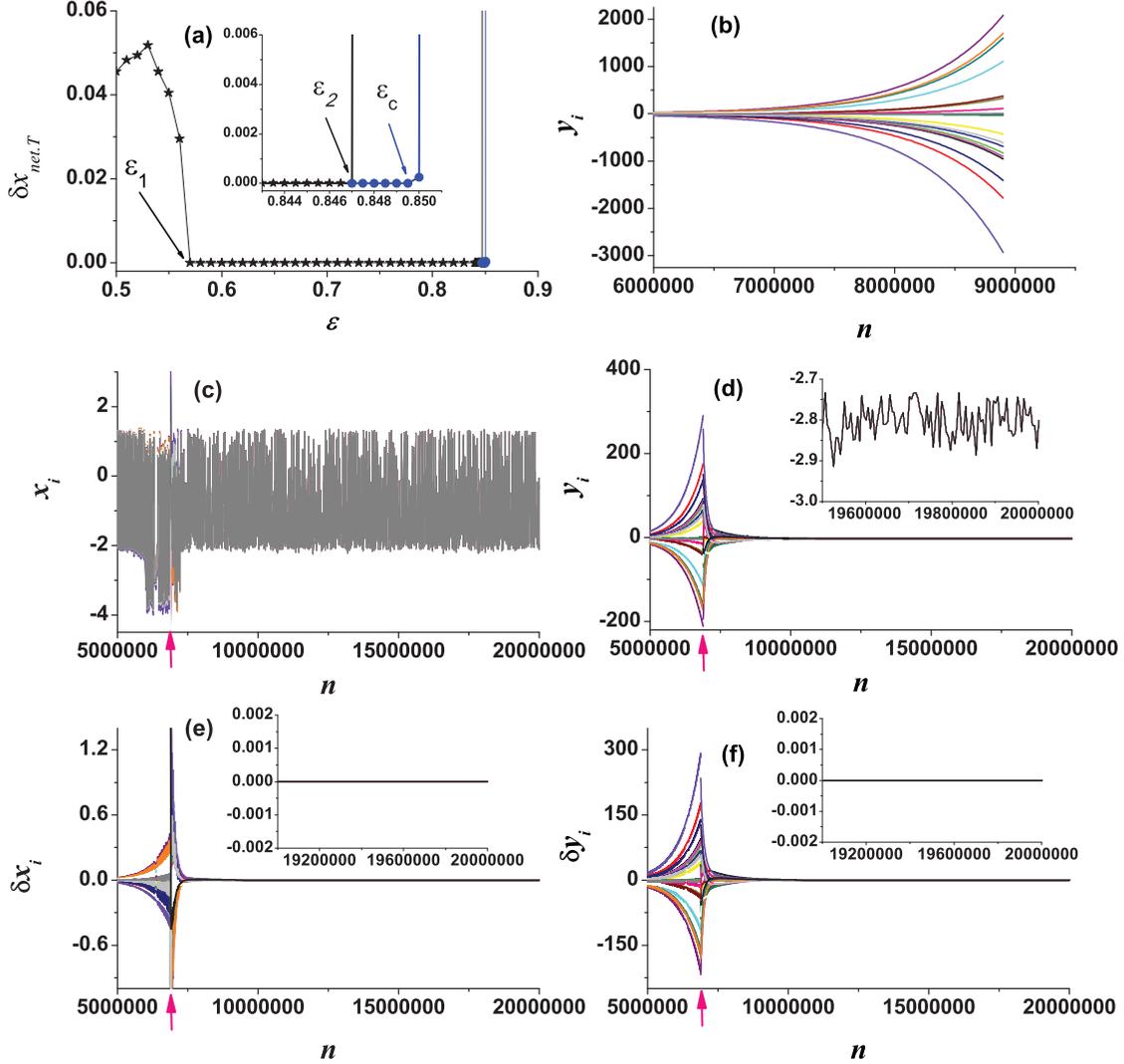
FIG. 6. (Color online) (a) Comparison of the averaged synchronization error $\delta x_{\mathrm{net},T}$ vs the coupling strength ε for the uncontrolled Rulkov map system and the controlled system on the 88th node, showing an extended coupling parameter region for the synchronization from $\varepsilon_2 = 0.8470$ to $\varepsilon_c = 0.8495$ (see details in the inset). (b) The divergent, desynchronous time series of y_i for the uncontrolled system; $\varepsilon = 0.8490$. (c) and (d) The synchronous time series of x_i and y_i, respectively, with the pull control added on the 88th node after $n = 6.886\,37 \times 10^6$ (arrow); $\varepsilon = 0.8490$. (e) and (f) The corresponding time evolutions of the trajectory error δx_i and δy_i. A BA scale-free network with $N = 100$ and $\langle k \rangle = 60$ is considered here.

individual node is capable of exhibiting chaotic oscillation; in this paper, we choose $\alpha = 4.1$. To test the effectiveness of the pull control method, a scale-free network with $N = 100$ and $\langle k \rangle = 60$ is constructed, whose synchronization regime of the networked system is found to stay at $(\varepsilon_1, \varepsilon_2) \simeq (0.560, 0.8470)$ [Fig. 6(a)]. Figure 6(b) clearly shows that the uncontrolled system is divergent when the coupling strength is chosen outside of the synchronization regime; $\varepsilon = 0.8490$. In contrast, the system dynamics changes and becomes quickly converged after only one controller is added on the 88th node on both $x_{88}(n)$ and $y_{88}(n)$ variables in Eqs. (5); all these system variables and the trajectory errors are shown in Figs. 6(c), 6(d), 6(e), and 6(f), respectively. The method for choosing the 88th node is still based on our eigenvalue and eigenvector analysis of the critical spatial modes of the network relying on the master stability function theory. In addition, the controllable

region is slightly enlarged based on our numerics: $(\varepsilon_2, \varepsilon_c) \simeq (0.8470, 0.8495)$; see Fig. 6(a) and its inset for the zoomed-in view.

As another example, the model of coupled classical chaotic Rössler oscillators on complex networks is studied,

$$\dot{x}_i = -y_i - z_i + \frac{\varepsilon}{d_i} \sum_{j=1}^{N} A_{ij}[H(x_j) - H(x_i)],$$

$$\dot{y}_i = x_i + 0.2y_i, \quad \dot{z}_i = 0.2 + (x_i - 5.7)z_i, \tag{6}$$

with the connecting function $H(x) = x$ and the classical parameter set for the chaotic Rössler oscillator chosen for simplicity. A scale-free network with $N = 100$ and $\langle k \rangle = 8$ has been generated. For this network, in the absence of the control, the synchronization of the coupled systems appears within
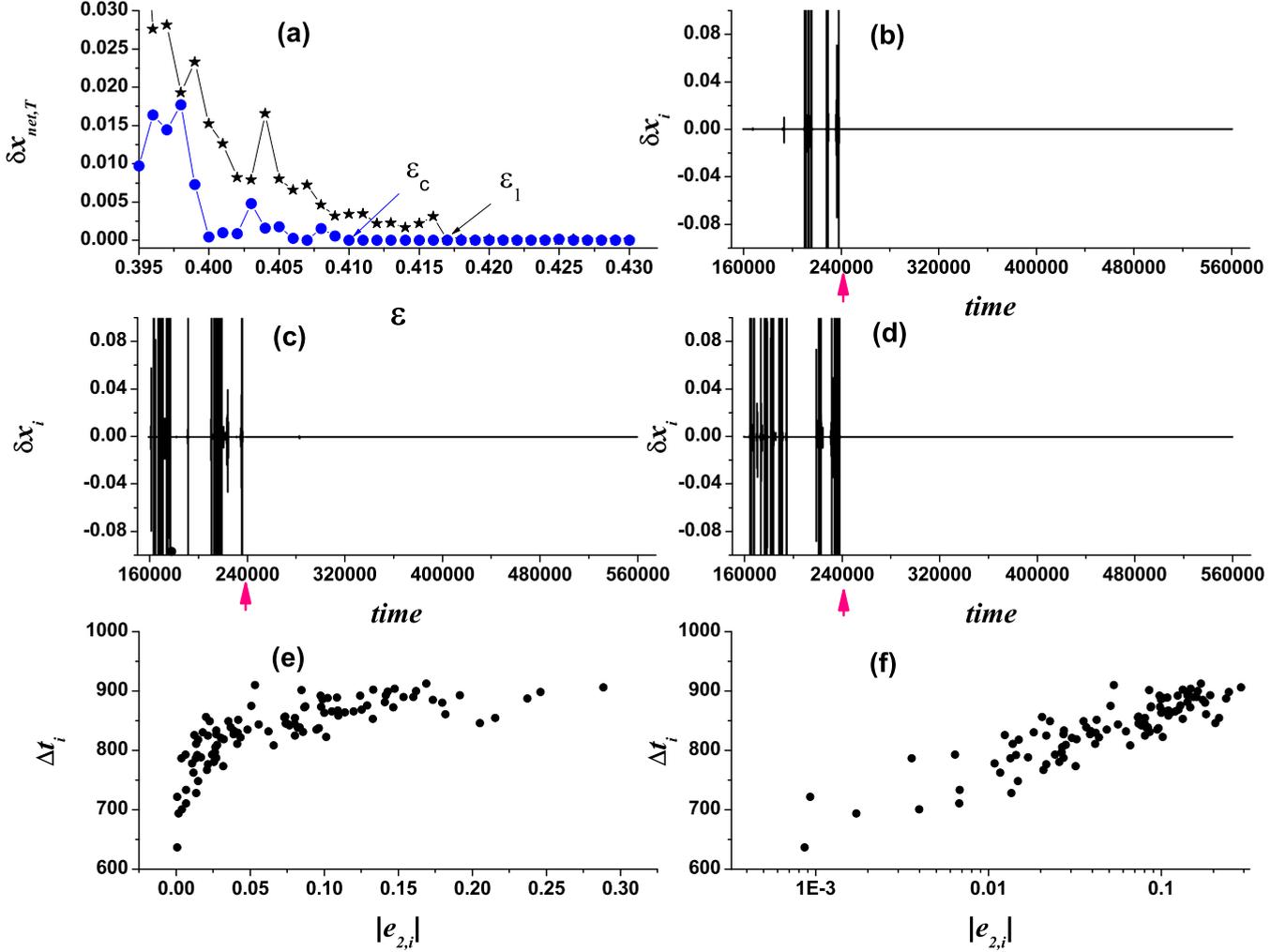
FIG. 7. (Color online) (a) Comparison of the averaged synchronization error $\delta x_{net,T}$ vs the coupling strength ε for the uncontrolled Rössler system and the controlled system on the 98th node, showing an extended coupling parameter region for the synchronization from $\varepsilon_1 = 0.417$ to $\varepsilon_c = 0.410$. (b), (c), and (d) The time evolutions of the trajectory error δx_i for the desynchronization-synchronization switch under $\varepsilon = 0.412$ with the pull control added on the 98th node, $\varepsilon = 0.405$ with the pull control added on the five leading (least stable) nodes, and $\varepsilon = 0.395$ with the pull control added on the ten leading nodes after $t = 240\,000$, respectively. The controller is added with a very small time step $\Delta\tau = 0.0005$. (e) The average transient period, Δt_i, over 600-time realizations as a function of the eigenvector element, $|e_{2,i}|$; $\varepsilon = 0.5$ ($\varepsilon \gtrsim \varepsilon_1$). (f) The semilogarithmic plot of (e), indicative of the same exponential relation: $e^{\Delta t_i} \propto |e_{2,i}|$. Here a BA scale-free network with $N = 100$ and $\langle k \rangle = 8$ is exemplified.

the regime $\varepsilon \in (\varepsilon_1, \varepsilon_2)$, $\varepsilon_1 = 0.155/\lambda_2 \simeq 0.417$, and $\varepsilon_2 = 4.512/\lambda_N \simeq 2.781$, where λ_2 and λ_N are the corresponding second and largest eigenvalues of the Laplacian matrix L in Eq. (2). For the pull control, one controller is added on all three variables x_i, y_i, z_i of the 98th node with a very small time step $\Delta\tau$ ($\Delta\tau = 0.0005$). Again the controlled node, namely the least stable node, should correspond exactly to the node with the largest absolute eigenvector element of $|e_{2,i}|$. In this situation, it is found that its controllable region has been extended from $\varepsilon_c \simeq 0.410$ to $\varepsilon_1 \simeq 0.417$ [Fig. 7(a)]. The time evolution of trajectory error δx_i at $\varepsilon = 0.412$ with only one controller is shown in Fig. 7(b), where the systems are desynchronous first and then become synchronized when the control is switched on after $t = 240\,000$. In fact, the systems can also become synchronous under a much smaller coupling strength ε by controlling more nodes. Two examples are the

controls on the first five leading nodes at $\varepsilon = 0.405$ [Fig. 7(c)] and the first ten leading nodes at $\varepsilon = 0.395$ [Fig. 7(d)]; both are successful. This is quite similar to what we have found for the coupled logistic maps in Figs. 4(b) and 4(d). We have also tested the push control method at $\varepsilon = 0.5$ when the systems are originally synchronous. We can push one node of the network by $z_i(t_0) = z_i(t_0) + \delta$ and $\delta = 20$. Similarly, we calculated the averaged transient periods for the ith node controlled, Δt_i, over 600 realizations with different initial conditions. We plot their values as a function of the eigenvector element, $|e_{2,i}|$, in Fig. 7(e) and we show its semilogarithmic plot in Fig. 7(f). From these patterns, we can again easily find that the exponential relationship $e^{\Delta t_i} \propto |e_{2,i}|$ persists.

Therefore, we may conclude that the control method is very generic and can work for many different coupled dynamical systems.

IV. MATHEMATICAL ANALYSIS

We have given some simulation results of the pull-push control for the synchronization-desynchronization switch for several classical chaotic systems. It might be easy to understand the impact of the push control method, as one node of the coupled systems has been kicked away from the synchronous manifold. But how does the pull control method with the action on the precisely selected node also work? It still requires proof by theoretical analysis, which is shown as follows.

Below, we consider the simplest case for only one node being controlled, i.e., $J = \{i_0\}$, with i_0 denoting the controlled node. The analytical result can easily be extended to more complicated cases. Then Eq. (3) can be written as

$$x_{i_0}(n+1) = \frac{1}{d_{i_0}} \sum_{j=1}^{N} A_{i_0 j} x_j(n+1). \qquad (7)$$

Inserting it into Eq. (1), we obtain

$$x_{i_0}(n+1) = F(x_{i_0}(n)) + \frac{\varepsilon}{d_{i_0}} \sum_{j \neq i_0} A_{i_0 j} \left[H(x_j(n)) - H\left(\frac{1}{d_{i_0}} \sum_{j=1}^{N} A_{i_0 j} x_j(n) \right) \right] \qquad (8)$$

and

$$x_i(n+1) = F(x_i(n)) + \frac{\varepsilon}{d_i} \sum_{j \neq i_0} A_{ij}[H(x_j(n)) - H(x_i(n))] + \frac{\varepsilon}{d_i} A_{ii_0} \left[H\left(\frac{1}{d_{i_0}} \sum_{j=1}^{N} A_{i_0 j} x_j(n) \right) - H(x_i(n)) \right]$$
$$\text{if } i \neq i_0. \qquad (9)$$

For the connecting function $H(x)$, if $H(\gamma x) \simeq \gamma H(x)$ holds (here γ is a constant), then the following approximation holds:

$$H\left(\frac{1}{d_{i_0}} \sum_{j=1}^{N} A_{i_0 j} x_j(n) \right) \simeq \frac{1}{d_{i_0}} \sum_{j=1}^{N} A_{i_0 j} H(x_j(n)), \qquad (10)$$

which yields

$$x_{i_0}(n+1) = F(x_{i_0}(n)), \qquad (11)$$

$$x_i(n+1) = F(x_i(n)) + \frac{\varepsilon}{d_i} \sum_{j \neq i_0} \left(A_{ij} + \frac{1}{d_{i_0}} A_{ii_0} A_{i_0 j} \right) \times [H(x_j(n)) - H(x_i(n))] \quad \text{if } i \neq i_0. \qquad (12)$$

Together with Eqs. (7), (11), and (12), the new Laplacian matrix L' can be expressed as

$$L'_{ij} = \begin{cases} -\frac{1}{d_i}\left(A_{ij} + \frac{1}{d_{i_0}} A_{ii_0} A_{i_0 j} \right) & \text{if } i, j \neq i_0 \text{ and } i \neq j, \\ 1 & \text{if } i, j \neq i_0 \text{ and } i = j, \\ 0 & \text{if } i = i_0 \text{ or } j = i_0. \end{cases} \qquad (13)$$

Comparing this new Laplacian matrix L' with the original matrix L in Eq. (2), we find that the control gives rise to
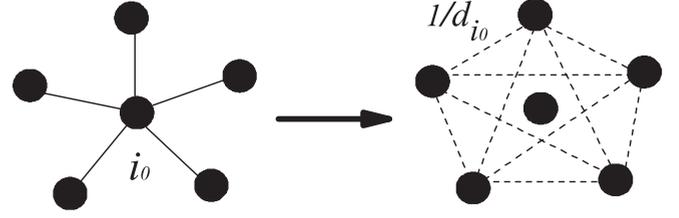
FIG. 8. Schematic show for the equivalent effect under the pull control: the controlled node (the i_0th node) removed from the network and an identical weighted interaction $1/d_{i_0}$ added between all of the neighbors of the controlled node i_0. See text for more details.

two key changes: (i) The controlled node has been completely isolated from the network as $L'_{i_0 j} = L'_{ii_0} = 0$ for all $j = i_0$ or $i = i_0$, and (ii) the interactions between its neighbors have been strengthened by an identical weight $\frac{1}{d_{i_0}}$; see the first line of L'_{ij} in Eq. (13), which implies that the synchronization stability of the remaining nodes should have been improved. For a schematic representation of these changes, see Fig. 8. Therefore, it is now easy to understand why the pull control method is so effective and also why it can only work for the slightly unstable synchronous state.

To test the validity of the above mathematical results, we also calculate the critical value ε_c of the controlled system on the basis of the master stability function method with the new Laplacian matrix L' in Eq. (13); the analytical results with the pink open circle lines are superimposed in Figs. 2(c) and 3(c), showing a perfect fit with the numerical results by black filled circles in the same panel.

V. CONCLUSION

In conclusion, we have proposed a precise pull-push control method for the synchronization-desynchronization switch on complex networks under a unified framework. For the pull control method, the synchronization region can be enlarged for both $\varepsilon < \varepsilon_1$ and $\varepsilon > \varepsilon_2$ by controlling only one node or a subset of nodes. Meanwhile, for the push control method, the desynchronization can be achieved by kicking one node or a subset of nodes. Based on the generalized eigenvector centrality from the Laplacian matrix, the pull-push control method has been found to be very effective. In addition, we have also tested whether the control method can be applied to some other nonlinear systems, such as the Rulkov map system and the time-continuous Rössler system, and we showed that it is indeed very efficient.

Finally, it is worth mentioning the following: (i) The pull control method, which uses the internal signals of the neighbors of the bad node, is essentially different from the pinning control method, which uses the external signal. Hence our method is a precise manipulation, which is achieved by changing the synchronization stability of the whole system but leaving the network structure unchanged. (ii) The pull-push method, which is concerned with the control of the synchronization or desynchronization state around the critical coupling strength, is also fundamentally different from the existing synchronization enhancement methods, which mainly address the relationship between the network structures and

the synchronization ability. (iii) The pull-push control method for the synchronization-desynchronization switch provides a unified framework for the stability change of the coupled systems with only a slight change of the controlled node; the generalized eigenvector centrality provides common information for the selection of the nodes. (iv) In this paper, we only considered the cases for instantly coupled elements and without time delays. It might be very interesting to study the delay effect on our control method in the future [6–9]. (v) Last but not least, since the control of both synchro-

nization and desynchronization of complex networks is of importance and significance in various fields, we hope that the insights and method presented here may enhance our understanding of collective behaviors in coupled complex systems in general, and that this simple, self-organized, effective, and precise control method may have applications in these fields.

ACKNOWLEDGMENTS

[1] A. Pikovsky, M. Rosenblum, and J. Kurths, *Synchronization: A Universal Concept in Nonlinear Sciences* (Cambridge University Press, Cambridge, 2001).

[2] E. Mosekilde, Y. Maistrenko, and D. Postnov, *Chaotic Synchronization: Applications to Living Systems* (World Scientific, Singapore, 2002).

[3] Y. Bar-Yam, *Dynamics of Complex Systems* (Addison-Wesley, Reading, MA, 1997).

[4] S. Bregni, *Synchronization of Digital Telecommunications Networks* (Wiley, Chichester, 2002).

[5] A. E. Motter, S. A. Myers, M. Anghel, and T. Nishikawa, Nat. Phys. **9**, 191 (2013).

[6] J. Kestler, E. Kopelowitz, I. Kanter, and W. Kinzel, Phys. Rev. E **77**, 046209 (2008).

[7] S. Heiligenthal, T. Dahms, S. Yanchuk, T. Jüngling, V. Flunkert, I. Kanter, E. Schöll, and W. Kinzel, Phys. Rev. Lett. **107**, 234102 (2011).

[8] I. Kanter, E. Kopelowitz, R. Vardi, M. Zigzag, W. Kinzel, M. Abeles, and D. Cohen, Europhys. Lett. **93**, 66001 (2011).

[9] R. Vardi, R. Timor, S. Marom, M. Abeles, and I. Kanter, Phys. Rev. E **87**, 012724 (2013).

[10] J. Buck and E. Buck, Science **159**, 1319 (1968).

[11] P. Uhlhaas, G. Pipa, B. Lima, L. Melloni, S. Neuenschwander, D. Nikolić, and W. Singer, Front. Integr. Neurosci. **3**, 17 (2009).

[12] F. C. Hoppensteadt and E. M. Izhikevich, Phys. Rev. E **62**, 4010 (2000).

[13] M. Nixon, M. Fridman, E. Ronen, A. A. Friesem, N. Davidson, and I. Kanter, Phys. Rev. Lett. **108**, 214101 (2012).

[14] M. Barahona and L. M. Pecora, Phys. Rev. Lett. **89**, 054101 (2002).

[15] T. Nishikawa, A. E. Motter, Y.-C. Lai, and F. C. Hoppensteadt, Phys. Rev. Lett. **91**, 014101 (2003).

[16] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, Physics Rep. **424**, 175 (2006).

[17] H. Hong, B. J. Kim, M. Y. Choi, and H. Park, Phys. Rev. E **69**, 067105 (2004).

[18] M. Zhao, T. Zhou, B. H. Wang, and W. X. Wang, Phys. Rev. E **72**, 057102 (2005).

[19] C. Y. Yin, W. X. Wang, G. R. Chen, and B. H. Wang, Phys. Rev. E **74**, 047102 (2006).

[20] A. Hagberg and D. A. Schult, CHAOS **18**, 037105 (2008).

[21] M. Jalili and A. A. Rad, CHAOS **19**, 028101 (2009).

[22] T. Zhou, M. Zhao, and B.-H. Wang, Phys. Rev. E **73**, 037101 (2006).

[23] A. E. Motter, C. S. Zhou, and J. Kurths, in *Science of Complex Networks: From Biology to the Internet and WWW: CNET 2004*, edited by J. F. F. Mendes, S. N. Dorogovtsev, A. Povolotsky, F. V. Abreu, and J. G. Oliveira, AIP Conf. Proc. No. 776 (AIP, New York, 2005), p. 201.

[24] X. G. Wang, Y.-C. Lai, and C.-H. Lai, Phys. Rev. E **75**, 056205 (2007).

[25] M. H. Li, X. G. Wang, Y. Fan, Z. Di, and C.-H. Lai, CHAOS **21**, 025108 (2011).

[26] G. W. Wei, M. Zhan, and C.-H. Lai, Phys. Rev. Lett. **89**, 284103 (2002).

[27] H. Gang and Qu Zhilin, Phys. Rev. Lett. **72**, 68 (1994).

[28] J. H. Xiao, G. Hu, J. Z. Yang, and J. H. Gao, Phys. Rev. Lett. **81**, 5552 (1998).

[29] X. F. Wang and G. R. Chen, Physica A **310**, 521 (2002).

[30] X. Li, X. F. Wang, and G. R. Chen, IEEE Trans. Circ. Sys. **51**, 2074 (2004).

[31] Z. H. Rong, X. Li, and W. L. Lu, in *Proceedings of IEEE International Symposium on Circuits and Systems* (IEEE, Piscataway, NJ, 2009), pp. 1689–1692.

[32] J. Zhou, J.-a. Lu, and J. Lü, IEEE Trans. Autom. Control **51**, 652 (2006).

[33] J. Zhou, J.-a. Lu, and J. Lü, Automatica **44**, 996 (2008).

[34] M. Porfiri and F. Fiorilli, CHAOS **19**, 013122 (2009).

[35] S. Floyd and V. Jacobson, IEEE/ACM Trans. Netw. **1**, 397 (1993).

[36] S. H. Strogatz, D. M. Abrams, A. McRobie, B. Eckhardt, and E. Ott, Nature (London) **438**, 43 (2005).

[37] K. Lehnertz and C. E. Elger, Phys. Rev. Lett. **80**, 5019 (1998).

[38] L. Glass, Nature (London) **410**, 277 (2001).

[39] V. H. P. Louzada, N. A. M. Araújo, J. S. Andrade, Jr., and H. J. Herrmann, Sci. Rep. **2**, 658 (2012).

[40] T. P. Chen, X. W. Liu, and W. L. Lu, IEEE Trans. Circ. Sys. **54**, 1317 (2007).

[41] M. Zhan, J. Gao, Y. Wu, and J. H. Xiao, Phys. Rev. E **76**, 036203 (2007).

[42] L. C. Freeman, Social Netw. **1**, 215 (1979).

[43] M. A. Beauchamp, Behavioral Sci. **10**, 161 (1965).

[44] L. C. Freeman, Sociometry **40**, 35 (1977).

[45] P. Bonacich, Am. J. Sociol. **92**, 1170 (1987).

[46] A. N. Langville and C. D. Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings* (Princeton University Press, Princeton, NJ, 2006).

[47] G. Lohmann, D. S. Margulies, A. Horstmann, B. Pleger, J. Lepsien, D. Goldhahn, H. Schloegl, M. Stumvoll, A. Villringer, and R. Turner, PloS ONE **5**, e10232 (2010).

[48] M. E. J. Newman, Phys. Rev. E **74**, 036104 (2006).

[49] C. B. Fu, H. Zhang, M. Zhan, and X. G. Wang, Phys. Rev. E **85**, 066208 (2012).

[50] C. B. Fu, W. J. Lin, L. Huang, and X. G. Wang, Phys. Rev. E **89**, 052908 (2014).

[51] L. M. Pecora and T. L. Carroll, Phys. Rev. Lett. **80**, 2109 (1998).

[52] J. Z. Yang, G. Hu, and J. H. Xiao, Phys. Rev. Lett. **80**, 496 (1998).

[53] A.-L. Barabási and R. Albert, Science **286**, 509 (1999).

[54] D. J. Watts and S. H. Strogatz, Nature (London) **393**, 440 (1998).

[55] P. Erdös and A. Rényi, Publ. Math. Inst. Hung. Acad. Sci. **5**, 17 (1960).

[56] N. F. Rulkov, Phys. Rev. Lett. **86**, 183 (2001).

[57] M. G. Rosenblum, A. S. Pikovsky, and J. Kurths, Phys. Rev. Lett. **76**, 1804 (1996).