Observability-Blocking Control using Sparser and Regional Feedback for Network Synchronization Processes

Abdullah Al Maruf and Sandip Roy

Abstract—The design of feedback control systems to block observability in a network synchronization model, i.e. to make the dynamics unobservable from measurements at a subset of the network’s nodes, is studied. First, a general design algorithm is presented for blocking observability at any specified group of \( m \) nodes, by applying state feedback controls at \( m + 2 \) specified actuation nodes. The algorithm is based on a method for eigenstructure assignment, which allows surgical modification of particular eigenvectors to block observability while preserving the remaining open-loop eigenstructure. Next, the topological structure of the network is exploited to reduce the number of controllers required for blocking observability; the result is based on blocking observability on the nodes associated with a vertex-cutset separating the actuation and measurement locations. Also, the design is modified to encompass regional feedback controls, which only use data from a subset of accessible nodes. The regional feedback design does not maintain the open-loop eigenstructure, but can be guaranteed to preserve stability via a time-scale argument. The results are illustrated with numerical examples.

I. INTRODUCTION

Assessing controllability and, dually, observability of complex dynamical networks has been a significant research focus of the network-controls community in the last few years [1]–[13]. A main outcome of this effort is a set of graph-theoretic necessary or sufficient conditions for observability and controllability, for canonical linear network models (e.g. synchronization models) [2], [8], [10]. This core graph-theoretic analysis has been extended in several directions, including toward evaluating Gramian-based metrics for control energy and state-estimation fidelity [3], [6], [8], studying input-output notions including output controllability and transfer-function zeros [5], [11], and addressing sensor/actuator placement [7], [12]. Analyses have also been undertaken for increasingly sophisticated linear models [9], and recently for some simple classes of nonlinear models [13]. These studies have been motivated by a number of applications, including cyber-threat assessment of multi-agent systems, infrastructure control and monitoring, and privacy-preserving design of network algorithms.

The studies on observability and controllability of network processes have largely focused on the relationship between a network’s native topology and its input-output behavior. A key further question of interest is whether a network’s control systems can be designed to facilitate or block observability/controllability from particular network locations, while preserving overall performance. Such design problems often arise when multiple control authorities or stakeholders have access to the network’s dynamics, and have cross-cutting goals in modulating the dynamics (whether cooperative or competitive). For instance, given increasing concern about cyber-attacks in the power grid, there is interest in designing wide-area control systems that not only damp oscillations but reduce the observability or controllability of adversaries [14]. Likewise, control initiatives in the air transportation system can be designed to improve or prevent estimation of flow dynamics [15], and control of water distribution systems to prevent malfeasance is of interest [16]. At a different scale, controller designs for multi-vehicle systems may need to consider security from intruders that can probe a subset of vehicles [17]. Therefore, there is a motivation to study the design of control systems in networks, to either facilitate or prevent state estimation. Here, we pursue a study in this direction, focused on designing controls at some network nodes to make the network’s dynamics unobservable at remote locations.

Specifically, in this study, we consider a linear network synchronization model [18]–[22] which can be actuated at a sparse set of network nodes. Our goal is to determine whether regional state feedback controls applied at these nodes can be used to enforce unobservability to monitoring at a set of remote nodes, while maintaining dynamical properties of the network. We study the design of such observability-blocking controllers, by applying and enhancing a method for eigenstructure assignment [23]–[25]. The main contributions of the study are as follows:

- A general state-feedback controller design algorithm is obtained, which blocks observability at a set of measurement nodes through assignment of a (or few) closed-loop eigenvector(s). Under broad conditions, the other open-loop eigenvectors and eigenvalues can be preserved.
- A topology-exploiting scheme which requires fewer actuation nodes to block observability is developed. It is shown that the required number of actuators depends on the cardinality of the vertex-cutset separating vertices corresponding to the actuation nodes and measurement nodes.
- The design scheme is further modified to accommodate regional feedback where only the states from a region or partition is used. In this case, the observability-blocking control scheme modifies the eigenstructure of the open-loop system, but stability can still be guaranteed via a
time-scale separation-based design.

The research described here connects to a wide literature on controller design for built network processes, such as stabilizer design to improve damping of inter-area oscillations in the power grid [26], mitigation of infection spreads [27], and pinning control of synchronization processes [28]. However, these various studies have been primarily focused on shaping the internal dynamics of network processes, while our interest here lies in shaping external or channel properties. We point the reader toward some preliminary work on shaping input-output dynamics of networks [12], [13], which is focused particularly on invariant zeros of network channels. The research described here also connects to the classical body of work on shaping the spectrum of linear systems (eigenvalues and eigenvectors) via feedback [23], [29], [30]. Relative to this literature, the contribution of our work lies in the development of pruned designs which exploit a graph structure overlaid on the linear dynamics.

The paper is organized as follows. In Section II, we formulate the observability-blocking controller design problem. In Section III, eigenstructure assignment via linear state feedback control, which is the basis for our approach, is briefly reviewed [23], [25]. The main results (design algorithms, proofs) are given in Section IV. Examples are given in Section V, and brief concluding remarks are included in Section VI. Some preliminary results in this direction were presented in [31]. Relative to [31], corrections and extensions of the design algorithms have been undertaken, the regional feedback design has been developed, new examples are included, and the motivation of the work has been updated.

II. PROBLEM FORMULATION

We consider a standard model for network synchronization [18]–[22], which has been enhanced to represent: 1) actuation nodes where feedback controls can be applied by a system operator, and 2) measurements or outputs available to a stakeholder (e.g., an adversary). Our primary objective is to design feedback controllers at the actuation nodes such that the dynamics are unobservable to the measurement nodes. We focus on synchronization processes as a representative class, because they arise in many settings where estimation-prevention may be important (e.g., robotic systems, water-distribution systems).

Formally, the synchronization model is defined on a weighted digraph \( G(\mathcal{V}, \mathcal{E} : \mathcal{W}) \). Here, the vertex set \( \mathcal{V} \) contains \( n \) vertices labeled as 1, 2, \ldots, \( n \). Each directed edge in \( \mathcal{E} \) is specified as an ordered pair of vertices, i.e., \((i, k)\) indicates an edge from vertex \( i \) to vertex \( k \). The weight of each edge \((i, k)\) \( \in \mathcal{E} \) is denoted by \( w_{ik} \), and is assumed to be positive. For this study, we assume the network graph to be strongly connected. The synchronization dynamics is specified by the (asymmetric) Laplacian diffusion matrix \( L \) of the graph. Specifically, \( L \) is an \( n \times n \) matrix whose entries are as follows: each off-diagonal entry \( L_{ik} \) is equal to \( -w_{ik} \) for all \((i, k) \in \mathcal{E}, \) otherwise 0; each diagonal entry \( L_{kk} \) is equal to \( -\sum_{i=1,i\neq k}^{n} L_{ki} \).

The synchronization model comprises a network with \( n \) nodes labeled 1, 2, \ldots, \( n \), which correspond to the graph vertices. Each node \( i \) has associated with it a scalar state \( x_i(t) \). The network state is defined as \( x(t) = [x_1(t) \ x_2(t) \ \cdots \ x_n(t)]^T \). Actuation can be provided at a set of \( q \) nodes \( \{r_1, r_2, \ldots, r_q\} \), which we call actuation nodes (respectively, actuation vertices in the graph). Also, the states of \( m \) measurement nodes (measurement vertices in the graph) \( \{s_1, s_2, \ldots, s_m\} \) are observed. We stress that the actuation and measurement nodes may overlap. The model dynamics are then given by:

\[
\dot{x} = -Lx + Bu, \quad (1)
\]

\[
y = Cx, \quad (2)
\]

where \( u \in \mathbb{R}^q \) is a vector containing the input signals at the actuation nodes, \( y \in \mathbb{R}^m \) is a vector containing the observations at the measurement nodes, \( B = [e_{r_1} \ e_{r_2} \ \cdots \ e_{r_q}] \), \( C = [e_{s_1} \ e_{s_2} \ \cdots \ e_{s_m}]^T \), and \( e_i \) is a 0-1 indicator vector in \( \mathbb{R}^n \) with only the \( i \)th entry equal to 1. We assume the pair \((-L, B)\) is controllable.

In this study, we consider the design of linear feedback controllers at the actuation nodes, to modulate the observability of the network dynamics with respect to the measurements \( y \). As a nominal case, a state feedback control scheme is considered. The controller at each node \( r_i, i = 1, 2, \ldots, q \), is thus specified as \( u_{r_i} = -k_i^T x \), where \( k_i \) is the control gain. Assembling the state feedback models for each actuation node yields:

\[
u = -Fx \quad (3)
\]

where \( F = [k_{r_1} \ k_{r_2} \ \cdots \ k_{r_q}]^T \). Upon application of the feedback control, the closed-loop dynamics become:

\[
\dot{x} = -(L + BF) x, \quad (4)
\]

\[
y = Cx, \quad (5)
\]

Our first goal in this study is to design the state-feedback controller, defined by the gain matrix \( F \), to make the closed-loop model unobservable from the measurements, while preserving other dynamical properties of the open-loop system. That is, we seek to enforce that the pair \((C, -(L + BF))\) is unobservable, while maintaining as much of the open-loop eigenstructure as possible. Second, we seek to develop sparse controller designs which can be used to block observability (i.e., make the pair \((C, -(L + BF))\) unobservable) using a lesser number of actuation nodes, by exploiting the graph topology of the network-synchronization model. Third, we study whether regional feedback controls (which only use state measurements in a partition of the network) can be used to block observability, while still preserving stability.

III. PRELIMINARIES

Our proposed method for designing observability-blocking controllers draws on and extends techniques for eigenstructure assignment [23], [25]. The classical study in this direction by Moore et al [23] presents necessary and sufficient
conditions for full eigenstructure assignment when closed-loop eigenvalues are distinct, while several follow-up studies address the repeated-eigenvalue case and give alternative perspectives on assignment [24], [30]; these methods have found application in several domains such as aircraft control and malicious control of the power grid [32]–[34]. Our recent note [25] has clarified that the classical eigenstructure assignment method can be adapted for surgical eigenstructure assignment – i.e. placement of all eigenvalues and a subset of eigenvectors – subject to conditions only on the dictated eigenvectors. Key results on surgical and full eigenstructure assignment are summarized in a single proposition here.

To present these results on eigenstructure assignment, we require some further notations. Formally, let us consider the closed-loop dynamics of a controllable linear system with an applied state feedback controller: \( \dot{x} = (A + BF)x \) where \( x \in \mathbb{R}^n \), \( A \in \mathbb{R}^{n \times n} \), \( B \in \mathbb{R}^{n \times q} \) and \( F \in \mathbb{R}^{q \times n} \) denote the state, open-loop state matrix, input matrix and gain matrix, respectively. Then, for any given complex \( \lambda \) we construct a matrix \( N(\lambda) = [N_1(\lambda)^T \quad N_2(\lambda)^T]^T \) where \( N(\lambda) \in \mathbb{C}^{(n+q) \times q} \), \( N_1(\lambda) \in \mathbb{C}^{n \times q} \), \( N_2(\lambda) \in \mathbb{C}^{q \times q} \), to have the following properties: the columns of \( N(\lambda) \) are linearly independent and span the null space of \( S(\lambda) = [(A - \lambda I_n) B] \). In this notation, the main results developed in [23] and [25] can be summarized as follows:

**Proposition 1:** Assume that \( \{\lambda_1, \lambda_2, \ldots, \lambda_n\} \) is any self-conjugate set of distinct complex numbers and the pair \( (A, B) \) is controllable. Then a real valued state feedback controller gain matrix \( F \) can be designed to: 1) place any \( r \) closed-loop eigenvalues at \( \lambda_i \in \{\lambda_1, \lambda_2, \ldots, \lambda_n\} \) and their corresponding eigenvectors at any \( \hat{v}_i = [\hat{v}_{i1}, \hat{v}_{i2}, \ldots, \hat{v}_{iq}] \), and 2) place the remaining \( n - r \) closed-loop eigenvalues at \( \lambda_i \in \{\lambda_{i+1}, \lambda_{i+2}, \ldots, \lambda_{n}\} \) provided that the following conditions hold: (i) \( \hat{v}_i \) is a set of linearly independent vectors in \( \mathbb{C}^q \), (ii) \( \hat{v}_i = \hat{v}_k \) whenever \( \lambda_i = \lambda_k \) where \( i, k \in \{1, 2, \ldots, r\} \), (iii) \( \hat{v}_i \in \text{column space of } N_1(\lambda_i) \) for all \( i \in \{1, 2, \ldots, r\} \), (iv) \( N_2(\lambda_i) \) is a self-conjugate set. Furthermore, for this feedback design, the closed-loop eigenvectors \( \hat{v}_i \in \{\hat{v}_{i+1}, \hat{v}_{i+2}, \ldots, \hat{v}_{iq}\} \) corresponding to the eigenvalues in \( \lambda_i \) satisfy: \( \hat{v}_i \in \text{column space of } N_1(\lambda_i) \) for all \( i \in \{r + 1, r + 2, \ldots, n\} \).

Generalizations of the result for the repeated eigenvalue case have been presented in [24], [25]; details are omitted here to simplify the presentation.

**IV. MAIN RESULTS**

The design of observability-blocking controllers is undertaken in three steps. First, we develop a general algorithm for blocking observability via state feedback control, when the number of actuation nodes \( q \) exceeds the number of measurement nodes \( m \) by two (\( q = m + 2 \)), see Section IV.A; the algorithm is general in the sense that it does not depend on the network graph or even the Laplacian form of the state matrix. Then, we demonstrate that a smaller set of actuation nodes can be used, if the network’s graph has a low-cardinality separating cutset between the actuation vertices and measurement vertices. In both cases, the controllers entirely preserve the open-loop dynamics, except perhaps a small set of eigenvectors. Lastly, we show that regional feedback controllers, which only use states from a region or partition of the network, are also able to block observability. These regional controllers distort the open-loop eigenstructure in general, but can be designed to guarantee stability.

**A. General Design of Observability-Blocking Controllers**

We present an algorithm for designing a state-feedback controller which makes the the pair \( (C, -(L + BF)) \) observable. The design is based on the eigenstructure assignment method reviewed above. Specifically, the algorithm places one eigenvector of the closed-loop state matrix to block observability, while aiming to keep all eigenvalues and most of the remaining eigenvectors unchanged. Precisely, the algorithm places one eigenvector \( v \) of \( -(L + BF) \) to satisfy \( Cv = 0 \) (while preserving the remainder of the eigenstructure), whereupon unobservability follows from the Popov-Belevitch-Hautus or PBH test [29].

In the ensuing development, for notational convenience and without loss of generality, the last \( m \) nodes (nodes \( n - m + 1, \ldots, n \)) are assumed to be the measurement nodes. Hence, \( C = [e_{n-m+1} \cdots e_n]^T \), and the the pair \( (C, -(L + BF)) \) is unobservable from the PBH test if and only if \( (L + BF) \) has an eigenvector whose final \( m \) entries are zero.

We first present the design algorithm for the case where the open-loop eigenvalues are distinct and real, and then use this result to obtain algorithms for more general settings. For the case with distinct and real eigenvalues, we label the eigenvalues of \( L \) as \( \lambda_1, \lambda_2, \ldots, \lambda_n \), and note that corresponding eigenvectors \( v_1, v_2, \ldots, v_n \) can be found which are linearly independent and span \( \mathbb{R}^n \). We also define the sets \( L_0 \) and \( V_0 \) which contain the open-loop eigenvalues and eigenvectors respectively.

The design algorithm is based on selecting any one of the eigenvalues of \( L \) and its associated eigenvector, say \( \lambda_p \) and \( v_p \) where \( p \in \{1, 2, \ldots, n\} \). The goal of the design is to construct a feedback control which uses \( q = m + 1 \) actuation nodes, such that the eigenvector \( v_p \) is modified to a vector \( \hat{v}_p \) whose entries corresponding to the measurement nodes are all zeros, while most of the remaining eigenvectors and all the eigenvalues are maintained. In this way, the mode \( -(\lambda_p) \) of the system dynamics is made unobservable. This can be done according to the following steps:

**Algorithm 1:**

1. Select one eigenvalue \( \lambda_p \) of \( L \) and its associated eigenvector \( v_p \), where \( p \in \{1, 2, \ldots, n\} \).
2. Compute a matrix \( N(\lambda_p) \in \mathbb{R}^{(n+q) \times q} \), whose columns are linearly independent and span the null space of \( S(\lambda_p) = [(L - \lambda_p I_n) B] \). Then partition \( N(\lambda_p) \) as \( N(\lambda_p) = [N_1(\lambda_p)^T \quad N_2(\lambda_p)^T]^T \), where \( N_1(\lambda_p) \in \mathbb{R}^{n \times q} \) and \( N_2(\lambda_p) \in \mathbb{R}^{q \times q} \) as described in Section III. Therefore \( N_1(\lambda_p) \) and \( N_2(\lambda_p) \) satisfy:

\[
((L - \lambda_p I_n) B) \begin{bmatrix} N_1(\lambda_p) \\ N_2(\lambda_p) \end{bmatrix} = 0. \tag{6}
\]
3) Partition $N_1(\lambda_p)$ as $N_1(\lambda_p) = [N_3(\lambda_p)^T \ N_4(\lambda_p)^T]$, where $N_3(\lambda_p) \in \mathbb{R}^{(n-m)+q}$ and $N_4(\lambda_p) \in \mathbb{R}^{m\times q}$. Then find a vector $h_p \neq 0$ which lies in the null space of $N_4(\lambda_p)$, i.e. which satisfies:
\[ N_4(\lambda_p) \ h_p = 0. \] (7)

4) Compute the vectors $\hat{v}_p$ and $z_p$ as:
\[ \hat{v}_p = N_1(\lambda_p) \ h_p, \] (8)
\[ z_p = N_2(\lambda_p) \ h_p. \] (9)

5) Form the open-loop modal matrix $V_0 \in \mathbb{R}^{n\times n}$ (i.e. $V_0 = [v_1v_2 \cdots v_n]$), and also a zero matrix $Z_0$ in $\mathbb{R}^{q\times n}$. Then check whether the $n$ vectors in $V_0 = V_0 \cup \{\hat{v}_p\}\{v_p\}$ are linearly independent. If yes, then follow Step 6 below to construct the closed-loop modal matrix $V$ and associated $Z$ matrix. Otherwise, skip Step 6, and follow steps 7 through 9 to construct the $V$ and $Z$ matrices.

6) Construct the matrix $V \in \mathbb{R}^{n\times n}$ from $V_0$ by replacing the column containing $v_p$ with $\hat{v}_p$. Similarly, construct the matrix $Z \in \mathbb{R}^{q\times n}$ from $Z_0$ by replacing the corresponding column with $z_p$. Therefore, $V = [v_1v_2 \cdots v_{p-1} \ \hat{v}_p \ v_{p+1} \cdots v_n]$ and $Z = [0 \cdots 0 \ z_p \ 0 \cdots 0]$. Then jump to Step 10.

7) Choose another eigenvalue and eigenvector pair $(\lambda_k, v_k)$ such that the $n - 1$ vectors in $V_1 = V_0 \cup \{\hat{v}_p\}\{v_p\}$ are linearly independent.

8) Then find a vector $v_k$ in the column space of $N_1(\lambda_k)$ such that the $n$ vectors in $V = V_0 \cup \{v_p, v_k\}\{v_p, v_k\}$ are linearly independent. Then compute $z_k$ as:
\[ z_k = N_2(\lambda_k) \ h_k \] such that $\hat{v}_k = N_1(\lambda_k) \ h_k. \] (10)

9) Construct the matrix $V \in \mathbb{R}^{n\times n}$ from $V_0$ by replacing the columns containing $v_p$ and $v_k$ with $\hat{v}_p$ and $\hat{v}_k$ respectively. Similarly, construct the matrix $Z \in \mathbb{R}^{q\times n}$ from $Z_0$ by replacing the corresponding columns with $z_p$ and $z_k$. Hence $V = [v_1v_2 \cdots v_{p-1} \ \hat{v}_p \ v_{p+1} \cdots \ v_k-1 \ \hat{v}_k \ v_{k+1} \cdots v_n]$ and $Z = [0 \cdots 0 \ z_p \ 0 \cdots 0 \ z_k \ 0 \cdots 0]$. When $k < p$, jump to Step 10.

10) Finally the gain matrix $F$ for the observability-blocking controller is uniquely obtained by (11):
\[ F = Z \ V^{-1}. \] (11)

As demonstrated in the following theorem, the controller designed according to the Algorithm 1 guarantees that the pair $(C, -(L + BF))$ is unobservable. Furthermore, it preserves all open-loop eigenvalues, and all but one $(v_p)$ or possibly two $(v_p, v_k)$ eigenvectors.

**Theorem 1:** Consider the network synchronization model. Assume that: 1) the eigenvalues of $L$ are distinct and real, 2) the number of actuation nodes is one more than the number of measurement nodes ($q = m + 1$), and 3) the pair $(-L, B)$ is controllable. Then the gain matrix $F$ of the state feedback controller obtained using Algorithm 1 blocks the observability of the model, specifically the pair $(C, -(L + BF))$ has an unobservable mode at $(-\lambda_p)$. Furthermore, if the set $V_0$ obtained in Step 5 contains $n$ linearly independent vectors, then all eigenvalues and all except one eigenvector $(v_p)$ of the open-loop model are maintained in the closed-loop model. Otherwise, all eigenvalues and all except two eigenvectors $(v_p, v_k)$ of the open-loop model are maintained in the closed-loop model.

**Proof:**

First note that $N_4(\lambda_p) \in \mathbb{R}^{m\times q}$, as defined in Step 3 of Algorithm 1, is a rank deficient matrix since $q = m + 1$. Therefore, a nonzero vector $h_p$ satisfying (7) is guaranteed to exist. Now consider two cases separately for the proof:

Case 1 where $V_0$ is a set of $n$ linearly independent vectors, and Case 2 where $V_0$ is not a set of $n$ linearly independent vectors.

For the Case 1, notice that $V$ defined in Step 6 is invertible. Then consider the gain matrix $F$ obtained in Step 10 in this case. From (11), it follows that $Fv_i = 0 \ \forall i \in \{1, \ldots, n\}\{p\}$, therefore all the open-loop eigenvalue and eigenvector pairs beside $(\lambda_p, v_p)$ are maintained in the closed-loop model. Now we will show that $(\lambda_p, v_p)$ is an eigenvalue and eigenvector pair of $L + BF$, using similar arguments to [23]. By multiplying (6) with $h_p$ and using (8), (9) and (11) we easily find that $(L + BF)\hat{v}_p = \lambda_p \hat{v}_p$. Hence, all the eigenvalues and eigenvectors of $L$ except $v_p$ is maintained in $L + BF$ for Case 1 where $\hat{v}_p$ is the new closed-loop eigenvector.

Now consider the Case 2. For this case, first note that $N_1(\lambda_p)$ is full rank because $B$ is full rank [23], [31]. Therefore $\hat{v}_p$ obtained from (8) in Step 4 is non-zero. Since $\hat{v}_p$ is a linear combination of the open-loop eigenvectors other than $v_p$, at least one eigenvector $v_k$ can be found in Step 7 such that $V_1$ is a set of $(n - 1)$ linearly independent vectors. Now consider $L_1 = L_0 \{\lambda_k\}$ and $L_2 = \{\lambda_k\}$. Note that $L_1, V_1$ and $L_2$ satisfy all the conditions given in Proposition 1. Therefore by applying Proposition 1, we can guarantee the existence of $\hat{v}_k$ in Step 8 such that $v_k$ is in the column space of $N_1(\lambda_k)$ and $V = V_0\cup \{v_p, v_k\}\{v_p, v_k\}$ is a set of $n$ linearly independent vectors. Hence in the Case 2, $V$ defined in Step 9 is invertible. Using arguments similar to those for Case 1, we can easily show that all the eigenvalues and eigenvectors of $L$ except $v_p$ and $v_k$ are maintained in $L + BF$ for Case 2 where $\hat{v}_p$ and $\hat{v}_k$ are the new closed-loop eigenvectors.

Finally note that the last $m$ entries of $\hat{v}_p$ are zero according to (7), hence the mode $(-\lambda_p)$ of the closed-loop system is unobservable, which completes our proof.

In the above algorithm, the check on linear independence of the vectors contained in $V_0$ in Step 5 is needed, as it dictates the construction of closed-loop modal matrix $V$.

When $V_0$ is not a set of linearly independent vectors, $V$ as defined in Step 6 is not invertible. In this case, we follow Steps 7, 8 and 9 instead to construct an invertible $V$ in order to obtain the gain matrix $F$. We note that our preliminary work in this direction [31] had an inaccuracy, in that this condition on $V_0$ and resulting additional steps were not indicated. The following example demonstrates this...
situation can occur, i.e. \( \hat{V}_0 \) may not be a set of \( n \) linearly independent vectors. However, we note that the situation arises only for specially constructed examples, and hence the additional steps are unnecessary in typical cases.

**Example:** Consider a network model with parameters \( L = [2, 0, -1, -1; 0, 3, -3, 0; -1, 5, -3, -1; 0, -1, 2] \), \( B = [1, 0, 0; 0, 1, 0; 0, 0, 1] \), and \( C = [0, 0, 1, 0; 0, 0, 0, 1] \). The Laplacian has eigenvalues 0, 2.4384, 3 and 6.5616. If we choose \( \lambda_p = 3 \), the vectors in \( V_0 \) are linearly dependent. This happens because \( \hat{v}_p \) obtained from (8) is in the range space of the eigenvectors associated with the eigenvalues at 0, 2.4384 and 6.5616. Now according to our algorithm we can choose \( \lambda_k = 6.5616 \) in Step 7. Then we choose \( \hat{v}_k = [-1.0569; 0.6022; -0.4802; 0.3370] \) from the null space of \( N_1(6.5616) \) such that \( \hat{V} \) is a set of four linearly independent vectors. This choice yields \( F = [3.5616, 0, 0, -3.5616; -0.5051, 0, 0, 0.5051; 2.4174, 1, -3.5616, 0.1442] \). This observability-blocking controller maintains the open-loop eigenstructure except eigenvectors \( v_p \) and \( v_k \), as stated in the Theorem 1.

Next, we modify Algorithm 1 to address the case that \( L \) has complex eigenvalues. The modified algorithm requires \( q = m + 2 \) actuation nodes to be used, in the case where the eigenvectors associated with a complex-conjugate pair are modified to achieve unobservability. Here is the algorithm:

**Algorithm 2:**

1) Select one eigenvalue \( \lambda_p \) of \( L \) and its associated eigenvector \( v_p \), where \( p \in \{1, 2, \ldots, n\} \). Then if \( \lambda_p \) is real, follow the steps under Sub-AlGORITHM 1 to obtain observability-blocking controller. Otherwise follow the steps under Sub-AlGORITHM 2.

**Sub-AlGORITHM 1:**

2)-4) Steps 2 through 4 are exactly the same as Algorithm 1.

5) Form the open-loop modal matrix \( V_0 \in \mathbb{R}^{n \times n} \) and the zero matrix \( Z_0 \in \mathbb{R}^{q \times n} \). Then check whether the \( n \) vectors in \( \tilde{V}_0 = V_0 \cup \{\hat{v}_p\} \backslash \{v_p\} \) are linearly independent. If yes, then follow Step 6 below to construct the closed-loop modal matrix \( V \) and associated \( Z \) matrix. Otherwise, skip Step 6, and follow steps 7 through 9 to construct the \( V \) and \( Z \) matrices.

6) Construct the matrix \( V \in \mathbb{R}^{n \times n} \) from \( V_0 \) by replacing the column containing \( v_p \) with \( \hat{v}_p \). Similarly, construct the matrix \( Z \in \mathbb{R}^{q \times n} \) from \( Z_0 \) by replacing the corresponding column with \( z_p \). Then jump to Step 10.

7) Find the largest-cardinality subset \( V_1 \) of \( \tilde{V}_0 \) such that \( v_p \in V_1 \) and \( V_1 \) is a self-conjugate set of linearly independent vectors.

8) Find \( v_k \) in the column space of \( N_1(\lambda_k) \) for all \( v_k \in V_0 \backslash V_1 \) such that the set \( \tilde{V} = V_1 \cup \{\hat{v}_k; v_k \in V_0 \backslash V_1\} \) is a self conjugate set of \( n \) linearly independent vectors. While doing so, maintain \( \tilde{v}_k = v_k \) whenever \( v_k = v_k \) and \( v_k \in V_0 \backslash V_1 \). Next, find the corresponding \( z_k \) based on (10) for all \( v_k \in V_0 \backslash V_1 \) (i.e. \( z_k = N_2(\lambda_k) h_k \), where \( h_k \) solves \( \hat{v}_k = N_1(\lambda_k) h_k \). Note that \( z_{k_2} = z_{k_1} \) whenever \( v_{k_2} = v_{k_1} \).

9) Construct \( V \) from \( V_0 \) by replacing the columns containing \( v_p \) and all \( v_k \in V_0 \backslash V_1 \) with \( \hat{v}_k \) and corresponding \( \hat{v}_k \) respectively. In the same manner construct \( Z \) from \( Z_0 \) by replacing the corresponding columns of \( Z_0 \) with \( z_p \) and all \( z_k \) obtained in step 8 respectively.

10) Finally, compute the gain matrix \( F \) using (11).

**Sub-AlGORITHM 2:**

2)-4) Steps 2 through 4 remain exactly the same as Algorithm 1. Additionally in Step 4 obtain \( \hat{v}_p \) and associated \( z_p \) by taking complex conjugates of \( v_p \) and \( z_p \) respectively.

5) Form the open-loop modal matrix \( V_0 \in \mathbb{R}^{n \times n} \) and the zero matrix \( Z_0 \in \mathbb{R}^{q \times n} \). Then check whether \( V_0 = V_0 \cup \{\hat{v}_p, \hat{v}_p\}\backslash\{v_p, v_p\} \) is a set of \( n \) linearly independent vectors. If \( V_0 \) is a set of linearly independent vectors, follow Step 6; otherwise skip Step 6 and follow Steps 7 through 9 to find \( V \) and \( Z \).

6) Construct the matrix \( V \) from \( V_0 \) by replacing the columns having \( v_p \) and \( v_p \) with \( \hat{v}_p \) and \( \hat{v}_p \) respectively. Similarly, construct the matrix \( Z \) from \( Z_0 \) by replacing the corresponding columns of \( Z_0 \) with \( z_p \) and \( z_p \) respectively. Then jump to Step 10.

7) Find the largest-cardinality subset \( V_1 \) of \( \tilde{V}_0 \) such that \( v_p \), \( \hat{v}_p \in V_1 \) and \( V_1 \) is a self-conjugate set of linearly independent vectors.

8) This step is the same as the Step 8 of Sub-AlGORITHM 1. Thus, find \( \hat{v}_k \) in the column space of \( N_1(\lambda_k) \) for all \( v_k \in V_0 \backslash V_1 \) such that the set \( \tilde{V} = V_1 \cup \{\hat{v}_k; v_k \in V_0 \backslash V_1\} \) is a self conjugate set of \( n \) linearly independent vectors, and find the corresponding \( z_k \).

9) Construct \( V \) from \( V_0 \) by replacing the columns containing \( v_p \), \( v_p \) and all \( v_k \in V_0 \backslash V_1 \) with \( \hat{v}_p \), \( \hat{v}_p \) and corresponding \( \hat{v}_k \) respectively. In the same manner, construct \( Z \) from \( Z_0 \) by replacing the corresponding columns of \( Z_0 \) with \( z_p \), \( z_p \) and all \( z_k \) obtained in step 8 respectively.

10) As before, compute the gain matrix \( F \) using (11).

Now we formalize the outcome of Algorithm 2 in the following theorem:

**Theorem 2:** Consider the network synchronization model. Assume that: 1) the eigenvalues of \( L \) are distinct, 2) the number of actuation nodes is two more than the number of measurement nodes \((q = m + 2)\), and 3) the pair \((-L, B)\) is controllable. Then the gain matrix \( F \) of the state feedback controller obtained using Algorithm 2 blocks the observability of the model, specifically the pair \((C, -(L + BF))\) has an unobservable mode at \((-\lambda_p)\). Furthermore, all the open-loop eigenvalues and the open-loop eigenvectors in the set \( V_0 \cap V_1 \) are maintained in the closed-loop model.

**Proof:**

The proof is similar to that of Theorem 1, and thus is presented briefly to avoid repetition. We consider two cases separately for the proof: Case 1 where \( \lambda_p \) is real and Case 2 where \( \lambda_p \) is not real.

Recall that the steps of Sub-AlGORITHM 1 are followed for the Case 1. The key steps of Sub-AlGORITHM 1, the Step 7 and
Step 8, differ from those of Algorithm 1; hence, here we focus on showing that $V_1$ and $V$ can be found respectively in Step 7 and Step 8 of Sub-Algorithm 1, verifying that the algorithm achieves a valid design. Since the open-loop eigenvectors are in conjugate pairs, it is immediate that in Step 7 a self-conjugate set $V_1$ can be found consisting of at least $(n - 3)$ open-loop eigenvectors. Now we guarantee existence of $V$ in Step 8 by using Proposition 1. To do so, consider $L_1$ to be the set of eigenvalues of $\mathbf{L}$ associated with the eigenvectors in $V_1$, and let $L_2 = L_n \setminus L_1$. Since $L_1, V_1$ and $L_2$ satisfy all the conditions of Proposition 1, therefore Proposition 1 can be applied which guarantees existence of $\hat{v}_k$ in the column space of $\mathbf{N}_1(\lambda_k)$ for all $\lambda_k \in L_2$ such that $V_1 \cup \{\hat{v}_k | \lambda_k \in L_2\}$ is a self-conjugate set of $n$ linearly independent vectors. Hence, $V$ in Step 8 can always be found. The remainder of the proof for this case is similar to that of Theorem 1.

Now consider the Case 2. Recall that the steps of Sub-Algorithm 2 are followed for this case. For this case we again will show the existence of $V_1$ and $V$ in the Step 7 and Step 8 of Sub-Algorithm 2, respectively. If the vectors $\hat{v}_p$ and $\hat{\nu}_p$ are linearly independent vectors, it is straightforward that the set $V_1$ consisting of at least $(n - 6)$ open-loop eigenvectors can always be found in Step 7. Then, it remains to show that linearly independent vectors $\hat{v}_p$ and $\hat{\nu}_p$ exist in Step 4 for any $\lambda_p$. Since $q = m + 2$, there exist two linearly independent vectors in the null space of $\mathbf{N}_{4}(\lambda_p)$. As $\mathbf{N}_1(\lambda_p)$ is full-rank, therefore using suitable linear combination of the two linearly independent vectors in the null space of $\mathbf{N}_4(\lambda_p)$ a complex vector $\hat{v}_p$ can always be obtained in Step 4 such that $\hat{v}_p$ and $\hat{\nu}_p$ are linearly independent. Thus, $V_1$ exists in Step 7. Now the arguments in Case 1 can be directly applied to guarantee the existence of $V$ in Step 8 of Sub-Algorithm 2. The remainder of the proof for this case is also similar to that of Theorem 1.

Finally note that $\mathbf{F}$ obtained in Algorithm 2 is a real valued matrix. This can be verified by standard manipulation of linear equations with conjugated coefficients (see [23] for further details).

According to the theorem, the closed-loop model preserves the open-loop eigenvectors contained in the set $V_0 \cap V_1$. From the arguments in the proof it is clear that $V_0 \cap V_1$ contains at least $(n - 6)$ open-loop eigenvectors (and at least $(n - 3)$ open-loop eigenvectors when $\lambda_0$ is real). Therefore Algorithm 2 modifies only six open-loop eigenvectors at the worst.

The matrices $\mathbf{V}$ and $\mathbf{Z}$ in Algorithm 2 are in general complex valued. Thus, calculation of $\mathbf{F}$ using (11) may not be computationally appealing as it involves inversion of complex valued matrix. However, an equivalent computation of $\mathbf{F}$ is possible, which involves manipulation of real valued matrices only and thus is computationally simpler. Here is the computation: $\mathbf{F} = \mathbf{Z}_{\text{mod}} \mathbf{V}_{\text{mod}}^{-1}$ where $\mathbf{V}_{\text{mod}}$ and $\mathbf{Z}_{\text{mod}}$ are real matrices constructed from $\mathbf{V}$ and $\mathbf{Z}$ by replacing their complex conjugate columns with their real and imaginary parts, respectively.

It should be noted that if our chosen $\lambda_p$ is real, we can use $q = m + 1$ actuation nodes in Algorithm 2. However, the following example shows that Algorithm 2 may fail if $\lambda_p$ is not real and $q = m + 1$ actuation nodes are used. Therefore $q = m + 2$ actuation nodes are required in general.

**Example:** Consider the network model with parameters $\mathbf{L} = [0.9, -0.8, -0.1, 0; -0.1, 0.9, -0.8, 0; -0.8, -0.1, 1, -0.1; 0, 0, -0.2, 0.2]$, $\mathbf{B} = [1, 0, 0; 0, 1, 0; 0, 0, 0; 0, 0, 1]$ and $\mathbf{C} = [0, 0, 1, 0; 0, 0, 0, 1]$. Here, $m = 2$ and $q = 3$. For this network model, if we choose any of the complex eigenvalues of $\mathbf{L}$ as $\lambda_p$ in the Step 1, then $\hat{v}_p$ obtained in the Step 4 will be real.[1] Therefore, $V_1$ in Step 7 of Sub-Algorithm 2 would not exist, as $\hat{v}_p = \hat{V}_p$. Hence, Algorithm 2 fails when $q = m + 1$ actuation nodes are used.

Next, we discuss the extension of our observability-blocking controller design algorithm for the case when $\mathbf{L}$ has repeated eigenvalues. The extension is based on the results on eigenstructure assignment for repeated eigenvalues [24], [25]. Below we briefly explain how this can be achieved by highlighting how the algorithms presented for the distinct eigenvalue case would change.

**Brief discussion on the repeated eigenvalue case:**

Repeated eigenvalues may be defective (have smaller geometric multiplicity than algebraic multiplicity), and hence may have Jordan chains which include generalized eigenvectors. In this case, the controller design algorithm must be modified to maintain the eigenvector and generalized eigenvector sequences of the Jordan chains. Thus, when any eigenvector/generalized eigenvector in a Jordan chain is modified in Step 4 or Step 8, all the following generalized eigenvectors in that Jordan chain must also be modified accordingly in that step. The modified/target closed-loop generalized eigenvectors and associated z vectors can be obtained through condition 3 of [24], which characterizes the assignability of generalized eigenvectors via state feedback. As the sequences in the Jordan chains are maintained, the technique for surgical eigenstructure assignment in [25] can again be applied to obtain target closed-loop eigenvectors/generalized eigenvectors in Step 8. Finally the gain matrix of the observability-blocking controller can be obtained using (11). It should be noted that, like Algorithm 2, conjugate open-loop eigenvectors/generalized eigenvectors must also be modified accordingly when $\mathbf{L}$ has complex eigenvalues.

The controller designed according to the above discussion blocks observability and is guaranteed to maintain all the open-loop eigenvalues in the closed-loop model. However, in general, it is difficult to characterize how much of the open-loop eigenvectors/generalized eigenvectors will be maintained. For the sake of completeness, we formalize the discussion in the following theorem. The proof is omitted (it is similar to the previous proofs but require the results in [24], [25]).

**Theorem 3:** Consider the network synchronization model.

---

[1] It can be shown that when each node in the network is either an actuation node or a measurement node, $\hat{v}_p$ obtained in Step 4 is real.
Assume that: 1) the number of actuation nodes is two more than the number of measurement nodes \((q = m + 2)\), and 2) the pair \((-L, B)\) is controllable. Then a state feedback gain matrix \(F\) can be designed so that observability is blocked (i.e. the pair \((C, -(L + BF))\) is unobservable, where any chosen open-loop mode \((-\lambda_p)\) can be made unobservable) and the open-loop eigenvalues are maintained in the closed-loop model.

Remark: The algorithms presented so far in this section are based on an implementation of the eigenstructure assignment techniques in [23]–[25]. Relative to [23]–[25], the contribution and effort in this study lie in designing the target eigenvectors to block observability and preserve most of the eigenstructure, while meeting the criteria for assignability given in [23]–[25]. From another viewpoint, [23]–[25] give implicit conditions that can be checked for assignability; our effort here shows how the target eigenstructure set can be designed to meet these conditions.

Several further remarks on the observability-blocking controller design algorithm are worthwhile:

1) The presented design method is general, in the sense that it does not depend on the state matrix having a Laplacian form, nor on the specific graph topology. In Section IV.B and IV.C, the design will be specialized for sparser and regional feedback controls, by exploiting the graph topology and Laplacian form of the network dynamics.

2) If the network graph is undirected or is a tree, observability can be blocked using \(q = m + 1\) actuation nodes, as the Laplacians have only real eigenvalues in these cases.

3) The controller designed using the algorithm maintains almost the entire open-loop eigenstructure. Such a structured modification of the eigenstructure allows for sequential design of feedback controllers for additional goals beyond unobservability. As an example, controllers could be first designed for a performance goal (e.g. eigenvalue placement), and then a second control loop could be added to block observability while maintaining the previously placed eigenvalues. In addition, maintenance of eigenstructure may be appealing from a privacy standpoint, in the sense that other stakeholders may not be able to distinguish the presence of the observability-blocking controller.

4) The design method allows the control designer to choose which mode is made unobservable. The mode can be chosen, for example, based on the achieved unobservable subspace or the extent to which the open-loop dynamics is modified.

5) Multiple modes of the system dynamics can be made unobservable using the design method, by modifying Step 4 so that all the corresponding eigenvectors have the necessary zero patterns. Alternatively, sequential design can also be used to enforce unobservability on multiple modes. At most \((n - m)\) modes can be made unobservable.

6) The presented design method can be extended for uncontrollable systems. The main idea is that all the target eigenvectors/generalized eigenvectors must be chosen to maintain the open-loop left eigenvectors associated with all uncontrollable eigenvalues.

7) We can also design a controller to enable observability of a mode \((-\lambda_p)\) for which the open-loop model is unobservable. In that case, \(h_p\) must be selected in Step 3 in such a way that \(N_4(\lambda_p) h_p \neq 0\). By using \((n - m + 1)\) actuation channels, we can guarantee the existence of such an \(h_p\), however many fewer controllers often suffice. Once \(h_p\) is selected, the remainder of the design is similar to that for the observability-blocking control.

B. Sparser Observability-Blocking Using Network Graph Cutsets

In the previous section, we showed that observability can be blocked at a set of measurement nodes, by applying state feedback at two more nodes than there are measurement nodes. By exploiting the topological structure of the Laplacian matrix, it turns out that we can often block observability using state feedback at a smaller set of network nodes. The genesis of this sparser design is that blocking observability on the nodes associated to a vertex-cutset of the network graph using actuation in one partition can serve to block observability at all nodes associated with the other partition. We formalize this notion first, and then discuss how the idea can be used to obtain sparser observability-blocking designs.

![Fig. 1. Network graph \(G\) with the vertex-sets considered in the sparser design.](image)

To formalize the notion, it is helpful to explicitly define two synchronization network models which have different measurement paradigms. Specifically, we refer to the network synchronization model defined in Section II as the base synchronization network model. We also consider a second model which has the same state dynamics and actuation nodes as the base model (hence the state matrix is \(L\) and the input matrix is \(B\)), however the measurement model is different. Specifically, to define the measurement model, we consider a vertex-cutset of the network graph that separates actuation vertices and measurement vertices in the base model, as illustrated in Fig. 1. In the figure, the cutset \(V_{cut}\) partitions the graph into two vertex-sets \(V_1\) and \(V_2\) such that
\( V_1 \) does not include any measurement vertex, \( V_2 \) does not include any actuation vertex and there is no edge between any vertex in \( V_1 \) and any vertex in \( V_2 \). Note that \( V_1, V_{\text{cut}} \) and \( V_2 \) are mutually exclusive and \( V_1 \cup V_{\text{cut}} \cup V_2 = V \). We stress that the cutset may include the actuation vertices or measurement vertices of the base model as shown in the figure, and hence a cutset always exists such that \( |V_{\text{cut}}| \leq m \) where \( |V_{\text{cut}}| \) represents the cardinality of \( V_{\text{cut}} \). (In the extreme case, the vertex-cutset may include all the base model’s measurement vertices.) In the second model, new measurement vertices are included in the cutset-measurement synchronization network model, all the observability-blocking in the cutset-measurement synchronization network model implies observability-blocking in the base synchronization network model using measurements from \( V_2 \) through state feedback. From the bottom block of (13) we obtain \( \bar{L}_{V_2,V_2} \bar{v}_{p_{V_2}} = \lambda_p \bar{v}_{p_{V_2}} \). This implies \( \bar{v}_{p_{V_2}} = 0 \) since \( \lambda_p \) is not an eigenvalue of the block \( \bar{L}_{V_2,V_2} \) according to our assumption. Now note that \( V_2 \cup V_{\text{cut}} \) contains all the measurement vertices and the entries of \( \bar{v}_p \) corresponding to the vertices in \( V_2 \cup V_{\text{cut}} \) are zero. Hence the proof is complete.

Lemma 1 serves as a basis for an algorithm for sparser observability-blocking control. The main idea is to use the algorithms from Section IV.A to design an observability-blocking controller for the cutset-measurement synchronization network model (which may have fewer measurement nodes than the base model), whereupon Lemma 1 can be leveraged to guarantee that observability is blocked in the original model. Specifically, this can be done as follows. First, a small-cardinality vertex-cutset \( V_{\text{cut}} \) separating the measurement vertices from the actuation vertices in the network graph is chosen, and the associated cutset-measurement synchronization network model is formed. Then, in accordance with the algorithms in Section IV.A, an eigenvalue \( \lambda_p \) of \( \bar{L} \) is chosen, whose eigenvector will be modified to block observability; however, an additional technical criterion is imposed on the selection, that \( \lambda_p \) is not an eigenvalue of the block \( \bar{L}_{V_2,V_2} \). Thereupon, the algorithms presented in Section IV.A can be applied to find the controller that makes the state at \((-\lambda_p)\) unobservable in the cutset-measurement synchronization network model using \(|V_{\text{cut}}|+2\) actuation nodes. From Lemma 1, it is immediate that the same controller blocks observability in the base synchronization network model. In this way the desired sparser observability-blocking controller is found that requires only \(|V_{\text{cut}}|+2\) actuation nodes. Further, this controller maintains all of the open-loop eigenvalues in the closed-loop, according to Theorem 3. The above design requires \( \bar{L} \) to have at least one eigenvalue which is not an eigenvalue of the block \( \bar{L}_{V_2,V_2} \). We claim that there is such an eigenvalue, when the network graph is strongly connected. In particular, \( \bar{L} \) has an eigenvalue at zero, but under the strong connectivity condition, \( \bar{L}_{V_2,V_2} \) cannot have an eigenvalue at zero. This is because \( \bar{L}_{V_2,V_2} \) is a grounded Laplacian where every vertex in \( V_2 \) can be reached through a directed path from some vertex in \( V_1 \cup V_{\text{cut}} \), and hence all the eigenvalues of \( \bar{L}_{V_2,V_2} \) are in the open right half plane (see Theorem 1 of [35]). We summarize this result in the following theorem. The proof is omitted as it is immediate from the above discussion.

**Theorem 4:** Consider the synchronization network model, and say that a separating vertex-cutset \( V_{\text{cut}} \) has been found.
Assume that: 1) the number of actuation nodes is two more than the cardinality of the vertex-cutset \( V_{\text{cut}} \) \( (q = |V_{\text{cut}}| + 2) \), and 2) the pair \((-L, B)\) is controllable. Then the gain matrix \( F \) of state feedback controller can be designed, according to the process described above, to block observability (i.e. to make the pair \((C, -(L+BF))\) unobservable). Furthermore, for this design, all the open-loop eigenvalues are maintained in the closed-loop system.

Remark: When all the eigenvalues of \( L \) are real, we can use one fewer actuation node \( (q = |V_{\text{cut}}| + 1 \) actuation nodes) in our design.

The sparser observability-blocking controller design formalized in Theorem 4 is especially useful when the cardinality of the cutset is much less than the number of measurement nodes \( m \) (i.e. \( |V_{\text{cut}}| << m \)). Graphs for many real-world networks are sparse, and have small-cardinality vertex-cutsets. For instance, the power grid is known to have a sparse structure with regions in the grid connected by few long transmission lines, while backbones for traffic and communication infrastructures are planar and often quite sparse. Networks in nature with scale-free structures are also essentially sparse [36]. In such networks, Theorem 4 indicates that observability can be blocked, and hence security against adversaries can be enhanced, using only a few actuators compared to the number of measurement nodes. As a limiting case, if the network graph is tree, we can block observability in a connected subnetwork using only two actuators, assuming that the model is controllable.

C. Observability-Blocking Using Regional State feedback Controllers

Thus far, we have considered the design of state feedback controllers to block observability. However, because observability-blocking control problems tend to arise in networks with independent or adversarial control authorities, these authorities sometimes may not have access to the full network state. Here, we consider the alternative that control authorities only can access states in a region of the network. For this case, we demonstrate that observability can be blocked at measurement nodes even outside the region, by blocking observability at nodes associated with a vertex-cutset at the boundary of the accessible region. A general design of this sort, which however does not necessarily maintain stability, is demonstrated in Theorem 5. Then stability-preserving design is obtained by applying a time-scale separating control (Theorem 6).

Formally, we assume that there is a subset of nodes in the network whose states can be accessed and used in feedback by the control authorities; this subset is assumed to include all actuation nodes. We refer to these nodes as accessible nodes (respectively accessible vertices in the graph), and the remaining as inaccessible nodes (respectively vertices).

To design observability-blocking controllers, it is again helpful to consider two synchronization network models. The first model is the base synchronization network model defined in Section IV.B. The second model resembles the measurement-cutset synchronization network model, but only encompasses the subnetwork of accessible nodes, and also defines a separating cutset based on accessibility.

To develop the second model, we consider a vertex-cutset \( V_{\text{cut}} \) of the network graph, which creates the following partitions (see Fig. 2): 1) a partition \( V_1 \) with only accessible vertices which does not contain any measurement vertices; and 2) a partition \( V_2 \) with other vertices which includes all the inaccessible vertices (and may include accessible vertices) and does not contain any actuation vertices. Here, the cutset can contain actuation and measurement vertices, but all vertices in the cutset must be accessible. It is easy to check such a cutset can always be found. Next, we distinguish the accessible and inaccessible vertices in \( V_2 \) as \( V_3 \) and \( V_4 \) respectively. The model is defined on the accessible vertices i.e. vertex-set \( V_1 \cup V_{\text{cut}} \cup V_3 \). Formally, the network graph of the second model is the subgraph of \( G \) induced by the vertex-set \( V_1 \cup V_{\text{cut}} \cup V_3 \) (see [37] for background on induced subgraphs). We denote this subgraph and associated Laplacian matrix as \( \tilde{G} \) and \( \tilde{L} \) respectively. This model is defined as having the same actuation vertices as the base model. However, the measurement vertices are defined differently, as the vertices in \( V_{\text{cut}} \). We denote the input and output matrix as \( \tilde{B} \) and \( \tilde{C} \) respectively, and formally refer to this model as accessible-region synchronization network model. For convenience, we call the network defined on the subgraph of \( \tilde{G} \) induced by the remaining inaccessible vertex-set \( V_4 \) as the inaccessible-region synchronization network model and denote the associated Laplacian matrix as \( \tilde{L}_{\text{rem}} \).

Without loss of generality, we renumber the vertices/nodes as follows. We assign indices to the vertices in \( V_1 \) first, then vertices in \( V_{\text{cut}} \), then vertices in \( V_3 \), and lastly vertices in \( V_4 \). Based on this indexing, the Laplacian \( L \) can be partitioned into blocks as

\[
L = \begin{bmatrix}
L_{V_1V_1} & L_{V_1V_{\text{cut}}} & 0 & 0 \\
L_{V_{\text{cut}}V_1} & L_{V_{\text{cut}}V_{\text{cut}}} & L_{V_{\text{cut}}V_3} & L_{V_{\text{cut}}V_4} \\
0 & L_{V_3V_{\text{cut}}} & L_{V_3V_3} & L_{V_3V_4} \\
0 & L_{V_4V_{\text{cut}}} & L_{V_4V_3} & L_{V_4V_4}
\end{bmatrix}
\]

(14)

where e.g. \( L_{V_1V_1} \) refers to the block of \( L \) whose rows and columns correspond to the vertices of \( V_1 \) and \( V_{\text{cut}} \) respectively (and other blocks are referenced similarly). Similarly,
the Laplacian $\tilde{L}$ of the accessible-region synchronization network model can also be written as

$$
\begin{bmatrix}
\tilde{L}_{v_1,v_1} & \tilde{L}_{v_1,v_{cut}} & 0 \\
\tilde{L}_{v_{cut},v_1} & \tilde{L}_{v_{cut},v_{cut}} & \tilde{L}_{v_{cut},V_3} \\
0 & \tilde{L}_{v_{cut},v_{cut}} & \tilde{L}_{v_{cut},V_3} \\
\end{bmatrix}
= 
\begin{bmatrix}
L_{v_1,v_1} & L_{v_1,v_{cut}} & 0 \\
L_{v_{cut},v_1} & L_{v_{cut},v_{cut}} & L_{v_{cut},V_3} \\
0 & L_{v_{cut},v_{cut}} & L_{v_{cut},V_3} \\
\end{bmatrix}
$$

(15)

where e.g. $\tilde{L}_{v_1,v_{cut}}$ refers to the block of $\tilde{L}$ whose rows and columns correspond to the vertices of $V_1$ and $V_{cut}$ respectively (and the other blocks are referenced similarly).

The following lemma gives a condition under which an observability-blocking controller designed for the accessible-region synchronization network model yields an observability-blocking controller for the base synchronization network model. This result is then used in tandem with the observability-blocking controller design presented in Section IV.A to obtain a general design for the regional state feedback case.

**Lemma 2:** Consider a base synchronization network model and associated accessible-region synchronization network model. Suppose the gain matrix $\tilde{F}$ of the state feedback controller is designed to block observability in the accessible-region synchronization network model, in such a way that the pair $(\tilde{C}, -(\tilde{L} + \tilde{B}\tilde{F}))$ has an unobservable mode at $(-\lambda_p)$ where $\lambda_p$ is not an eigenvalue of the block $\tilde{L}_{v_1,v_{cut}}$. Then the gain matrix $F$ obtained from $\tilde{F}$ via appropriate zero padding (i.e. $F = [\tilde{F} \ 0]$) blocks observability in the base synchronization network model, specifically the pair $(C, -(L + BF))$ has an unobservable mode at $(-\lambda_p)$.

**Proof:**

Consider a synchronization network defined on graph $G = (V, E : W)$ and also consider the vertex-sets $V_1$, $V_2$, $V_{cut}$, $V_3$ and $V_4$ (see Fig. 2) as defined above. When $V_4 = \emptyset$, the proof is immediate from Lemma 1.

Now consider the case that $V_4 \neq \emptyset$ (i.e. the network has inaccessible nodes). Assume that $\tilde{v}_p$ is the eigenvector of $\tilde{L} + \tilde{B}\tilde{F}$ for the eigenvalue at $\lambda_p$. Now we will apply Lemma 1 to characterize the zero pattern of $\tilde{v}_p$. For that, first consider an intertim model on the accessible region whose state dynamics and actuation vertices are same as the accessible-region synchronization network model but measurement vertices are different, which are $V_{cut} \cup V_3$. Note, $V_{cut}$ serves as a cutset between actuation and measurement vertices of this intertim model. By assumption, $\lambda_p$ is not an eigenvalue of the block $\tilde{L}_{v_1,v_{cut}}$ and the gain matrix $\tilde{F}$ is designed to block observability in the accessible-region synchronization network model so that the mode $(-\lambda_p)$ is unobservable. Hence, Lemma 1 can be applied here by comparing the accessible-region synchronization network model and the intertim model respectively to the cutset-measurement synchronization network model and the base model of Lemma 1. Thereby per Lemma 1, $\tilde{F}$ also serves to block observability in the intertim model so that the mode $(-\lambda_p)$ is unobservable. As a consequence, all the entries of $\tilde{v}_p$ corresponding to $V_{cut} \cup V_3$, the measurement vertices of the intertim model, are zero. This implies $\tilde{v}_p = [\tilde{v}_{p,V_1} \ 0^T \ 0^T]^T$ where $\tilde{v}_{p,V_1} \neq 0$. Now we can write the eigenvalue equation for the eigenvalue $\lambda_p$ of the accessible-region synchronization network model as

$$
\begin{bmatrix}
L_{v_1,v_1} + \tilde{F}_{v_1,v_1} & L_{v_1,v_{cut}} + \tilde{F}_{v_1,v_{cut}} & L_{v_1,V_3} + \tilde{F}_{v_1,V_3} \\
L_{v_{cut},v_1} + \tilde{F}_{v_{cut},v_1} & L_{v_{cut},v_{cut}} + \tilde{F}_{v_{cut},v_{cut}} & L_{v_{cut},V_3} + \tilde{F}_{v_{cut},V_3} \\
0 & L_{v_{cut},v_{cut}} + \tilde{F}_{v_{cut},v_{cut}} & L_{v_{cut},V_3} + \tilde{F}_{v_{cut},V_3} \\
\end{bmatrix}
[\tilde{v}_{p,V_1} \ 0 \ 0]^T
= 
\begin{bmatrix}
\tilde{v}_{p,V_1} \ 0 \ 0 \\
0 \ 0 \ 0 \\
\end{bmatrix}
(-\lambda_p)
$$

(16)

Here, the feedback $\tilde{F}$ is partitioned according to the defined vertex-sets, e.g. $\tilde{F}_{V_1,V_{cut}}$ refers to the feedback applied to $V_1$ using measurements from $V_{cut}$ through state feedback. Also note that $B = [\tilde{B}^T \ 0^T]^T$. Let us consider $F = [\tilde{F} \ 0]$ and $v_p = [\tilde{v}_p^T \ 0^T]^T$. Then using (14), (15) and (16) we can write $(L + BF)\tilde{v}_p$ as

$$
\begin{bmatrix}
L_{v_1,v_1} + F_{v_1,v_1} & L_{v_1,v_{cut}} + F_{v_1,v_{cut}} & L_{v_1,V_3} + F_{v_1,V_3} \\
L_{v_{cut},v_1} + F_{v_{cut},v_1} & L_{v_{cut},v_{cut}} + F_{v_{cut},v_{cut}} & L_{v_{cut},V_3} + F_{v_{cut},V_3} \\
0 & L_{v_{cut},v_{cut}} + F_{v_{cut},v_{cut}} & L_{v_{cut},V_3} + F_{v_{cut},V_3} \\
\end{bmatrix}
[\tilde{v}_{p,V_1} \ 0 \ 0]^T
$$

(17)

Therefore, $\lambda_p$ and $v_p$ are the eigenvalue and associated eigenvector of $(L + BF)$. Since all the entries of $v_p$ corresponding to $V_{cut} \cup V_2$ (which includes all the measurement vertices) are zero, $(-\lambda_p)$ is an unobservable mode of the closed-loop base synchronization network model. Hence $F = [\tilde{F} \ 0]$ makes the base synchronization network model unobservable.

Lemma 2 is the basis of our algorithm for designing regional state feedback observability-blocking controller. The main idea is to use the algorithm presented in Section IV.A to design $\tilde{F}$ of the observability-blocking controller for the accessible-region synchronization network model, and then employ Lemma 2 to guarantee that $F = [\tilde{F} \ 0]$ blocks observability in the original model. This controller serves as a regional state feedback controller since the entries of $F$ associated to inaccessible nodes are zero and therefore only the states of the accessible nodes are used by this controller. Systematically, this controller can be obtained as follows. First, a vertex-cutset $V_{cut}$ is chosen as defined above and the associated accessible-region synchronization network model is formed. Then, in accordance with the algorithms in Section IV.A, an eigenvalue $\lambda_p$ of $\tilde{L}$ is chosen, whose eigenvector will be modified to block observability in the accessible-region synchronization network model; however, similar to the sparser observability-blocking controller design, an additional technical criterion is imposed on the selection, that $\lambda_p$ is not an eigenvalue of the block $\tilde{L}_{v_1,v_{cut}}$. Next, the observability-blocking controller algorithm in Section IV.A is applied to find $\tilde{F}$ that makes the mode at $(-\lambda_p)$ unobservable in the accessible-region synchronization network model using $|V_{cut}| + 2$ actuation nodes. From Lemma 2, it is immediate that $F = [\tilde{F} \ 0]$ blocks observability in the base synchronization network model. In this way, an observability-blocking controller is found in the base model that uses states of the accessible nodes only.

The design algorithm requires the graph $G$ to be strongly connected, and the pair $(\tilde{L}, \tilde{B})$ to be controllable. The strong connectivity condition guarantees that $\tilde{L}$ has at least one
appropriate eigenvalue, while the controllability condition allows the design of an observability-blocking controller in the accessible-region synchronization network model per Theorem 3. We formalize this result in the following theorem. The proof is omitted as it is immediate from the discussion.

**Theorem 5:** Consider a (base) synchronization network model which has a set of accessible nodes as defined above. Say that a separating cutset has been identified, and associated accessible-region synchronization network model has been constructed. Assume that: 1) the number of actuation nodes is two more than the cardinality of the vertex-cutset $V_{cut}$ ($q = |V_{cut}| + 2$), 2) the graph $\tilde{G}$ of the accessible-region synchronization network is strongly connected, and 3) the pair $(\tilde{L}, \tilde{B})$ is controllable. Then the gain matrix $F$ of the regional state feedback controller, which uses only the states of accessible nodes, can be designed according to the process described above to block observability in the base synchronization network model (i.e. to make the pair $(C, -(L + BF))$ unobservable).

It should be noted that the second condition in Theorem 5 can be relaxed for the special case where inaccessible vertices comprise the partition $V_2$ entirely. In that case, any eigenvalue of $L$ can be chosen in the design as $V_3 = \emptyset$, and therefore the graph $\tilde{G}$ does not need to be strongly connected. This implies that we can also relax this connectivity condition if we consider the accessible-region synchronization network model to be defined on the vertex-set $V_1 \cup V_{cut}$ only. However, the feedback control will then be restricted from using the states of the nodes associated to $V_3$ even though they are accessible.

The design algorithm presented in Theorem 5 has a drawback: the designed controller does not maintain the open-loop eigenvalues of the base synchronization network model. In fact, the closed-loop base model may become unstable (for instance, see Example 2 in Section V). For some special cases, the design can be guaranteed to maintain stability in the closed-loop base model. For instance, it can be argued that the designed regional feedback controller maintains stability in the closed-loop base model when the links between accessible-region synchronization network and inaccessible-region synchronization network are sufficiently weak (or equivalently, weights of the edges between accessible and inaccessible vertices are sufficiently small). However, stability cannot be guaranteed in general for this design.

Now we suggest a modification to the above design algorithm, such that unobservability can be enforced in the base model with a general guarantee on stability. The modified design uses time-scale separation principles. The idea is to design an observability-blocking controller in the accessible-region synchronization network model, which achieves a time-scale separation between the states of accessible and inaccessible nodes. This is done by shifting the eigenvalues of $L$ sufficiently far to the right first, and then using the algorithm presented in Section IV.A. For this design, the same argument as for Theorem 5 can be used to verify unobservability, while the time-scale separation guarantees stability. Specifically, the controller is designed as follows. First, as before vertex-cutset $V_{cut}$ is chosen and the associated accessible-region synchronization network model is formed. Then, using any standard eigenvalue assignment technique (e.g. [38], [39], [40]) the gain matrix $\tilde{F}_1$ of a state feedback controller is designed so that the closed loop $L + BF$ has all eigenvalues with real part greater than a pre-selected nonnegative constant $d$, and further the technical constraint that the assigned eigenvalues do not match with the eigenvalues of the block $\tilde{L}_{v_2,v_2}$ is met. Next, an eigenvalue $\lambda_p$ of $(L + BF_1)$ is chosen so that it is not an eigenvalue of block $\tilde{L}_{v_3,v_3}$. Thereupon, the observability-blocking controller design algorithm presented in Section IV.A is applied on $(L + BF_1)$ to obtain the gain matrix $\tilde{F}_2$, for which the pair $(\tilde{C}, -(L + BF_1 + BF_2))$ has an unobservable mode at $(-\lambda_p)$. The combination of these gains, i.e. $F = F_1 + F_2$, serves as the observability-blocking controller in the accessible-region synchronization network model. Now according to the Lemma 2, $F = [\tilde{F} \ 0]$ blocks observability in the base synchronization network model. Further, this regional state feedback controller also maintains stability when $d$ is chosen appropriately (e.g. large enough). The following theorem formalizes this.

**Theorem 6:** Consider a (base) synchronization network model which has a set of accessible nodes as defined above. Say that a separating cutset has been identified, and associated accessible-region synchronization network model has been constructed. Assume that: 1) the number of actuation nodes is two more than the cardinality of the vertex-cutset $V_{cut}$ ($q = |V_{cut}| + 2$), 2) the pair $(\tilde{L}, \tilde{B})$ is controllable. Then the gain matrix $F$ of regional state feedback controller, which uses only the states of accessible nodes, can be designed according to the process described above to block observability in the base synchronization network model (i.e. to make the pair $(C, -(L + BF))$ is unobservable) while maintaining stability of the closed-loop system.

**Proof:**

For a controllable linear time-invariant system, the eigenvalues can be placed anywhere in complex plane using a state feedback controller [29]. Since the pair $(\tilde{L}, \tilde{B})$ is assumed to be controllable, therefore $\tilde{F}_1$, as described in the above algorithm can always be found. Subsequently, a second control loop can be applied to block observability, using an argument identical to that for Theorem 5. Thus, we can conclude that a combined and zero-padded gain $F = [\tilde{F} \ 0]$ can always be found, which first moves the eigenvalues and then blocks observability in the base synchronization network model.

Now using time-scale separation based argument we will show that there are values of $d$ for which the closed-loop base synchronization network model is stable. Using (16) and (17), we can write the closed-loop base model as:

$$L + BF = \begin{bmatrix} (L + BF_1) + P_1 & L_{reg,rem} \\ L_{rem,reg} & L_{v_3,v_2} \end{bmatrix}$$  (18)
where \( P_1 \) and \( P_2 \) are positive semidefinite and non-zero diagonal matrices, \( L_{reg,rem} = \begin{bmatrix} 0^T & L_{V_{cut}V_\lambda}^T \end{bmatrix} \) and \( L_{rem,reg} = \begin{bmatrix} 0 & L_{V_{cut}V_\lambda} \end{bmatrix} \). Recall that \( \tilde{F}_2 \) maintains the eigenvalues of \( (\tilde{L} + \tilde{B}\tilde{F}_1) \). Now if we choose \( d \) to be very large, then it is obvious that the states of the accessible nodes will have very fast decaying dynamics in the close-loop system. Thereupon by choosing sufficiently large \( d \), the states of the accessible nodes can be made separated in time scale from those of inaccessible nodes. In that case the stability of the closed-loop model can be addressed via singular perturbation theory. Specifically in the case of such time-scale separation, according to the singular perturbation theory [41] the closed-loop system of the base synchronization network model will be stable, if \( \Re\{\lambda(L + B\tilde{F} + P_1)\} > 0 \) and \( \Re\{\lambda(L_{V_\lambda V_\lambda} - L_{reg,rem} (L + B\tilde{F} + P_1)^{-1} L_{rem,reg})\} > 0 \). (Here, \( \Re\{\lambda(A)\} \) denotes the real parts of all the eigenvalues of \( A \) following the notation in [41].) Now for arbitrarily large \( d \) these conditions reduce to \( \Re\{\lambda(L_{V_\lambda V_\lambda})\} > 0 \) as \( (L + B\tilde{F} + P_1)^{-1} \rightarrow 0 \). The proof is completed by noting that \( L_{V_\lambda V_\lambda} \) is a grounded Laplacian in a strongly connected graph and thus all of its eigenvalues are in the open right half plane [35].

**Remark:** When all the eigenvalues of \( L \) are real, we can use one fewer actuation node \( (q = |V_{cut}| + 1 \) actuation nodes) in our design.

Theorem 6 suggests that we can design observability-blocking controller which uses states of the accessible nodes only and preserves stability as long as the pair \( (\tilde{L}, \tilde{B}) \) is controllable and there are sufficient number of actuation nodes. Theoretically, an arbitrarily large value of \( d \) in the design leads to arbitrarily large gains in the controllers. However, in practice arbitrarily large \( d \) is not necessary in the design to achieve stability, as the edge-weights in the real-world networks are finite and in many cases small. Therefore for a real-world network satisfying the conditions given in Theorem 6, it should be possible to design regional state feedback observability-blocking controller with a relatively limited control gain that maintains stability (by choosing a moderate value of \( d \)).

Many real-world networks e.g. power, communication, and traffic networks, are sparse. The regional feedback controller design algorithm may be useful for such networks, in that it can permit authorities to prevent learning of network dynamics even without full measurements of the network state. As a limiting case, if the network graph is tree, observability can be blocked for a connected subnetwork using only two actuators, and without any access to the subnetwork states, assuming that the accessible-region model is controllable.

V. **Numerical Examples**

Two numerical examples are presented to illustrate the sparser controller design and the regional state feedback controller design, respectively.

**Example 1:** The sparser observability-blocking controller, as formalized in Theorem 4, is illustrated. A network with 11 nodes is considered. The network’s graph is shown in Fig. 3: all edges shown in the figure are bi-directional, and the edge-weights are set to 1. Nodes \{6, 8, 9, 11\} are measurement nodes, while nodes \{1, 2, 3, 4, 5, 10\} are actuation nodes.

Our goal is to design a sparse state feedback controller that blocks observability at these measurement nodes. In this example, Vertex 5 is a single vertex-cutset that separates the measurement and possible actuation vertices in the network graph. Thus, according to Theorem 4, we should be able to block observability at the measurement nodes by blocking observability at Node 5, using two actuation nodes. (Note that the graph is undirected, therefore \( q = |V_{cut}| + 1 \) actuation nodes suffices.) We select Nodes 1 and 10 as the actuation nodes. The model is controllable for this selection.

We apply the controller design process described in Section IV.B, which uses Algorithm 1, to design the feedback controller. Specifically, we choose \( \lambda_p = 0.1853 \) from the eigenvalues of \( L \). (It is easy to check that \( \lambda_p \) is not an eigenvalue of block \( L_{V_\lambda V_\lambda} \).) Then following the algorithm, the control gains for controllers at Nodes 1 and 10 are obtained as \([0.7311; 0.7281; 0.5956; 0.5601; 0.2527; -0.4445; -0.7016; -0.8286; -0.8021; 0.8938; -0.9846]\) and \([-0.5980; -0.5956; -0.4872; -0.4581; -0.2067; 0.3636; 0.5739; 0.6778; 0.6561; -0.7311; 0.8054]\), respectively.

In the closed-loop system, the eigenvector associated with \( \lambda_p \) is modified to \([-0.6008; 0.2656; -0.0696; 0.0696; 0; 0; 0; 0; 0.7475; 0]\), while other eigenvectors and all the eigenvalues remain unchanged. The zero entries of this modified eigenvector indicate that that the base model has now unobservable no matter which nodes in the partition \{5, 6, 7, 8, 9, 11\} are measured.

**Example 2:** A regional state feedback control design which blocks observability while maintaining stability via a time-scale separation is illustrated. A network with 10 nodes is considered. The network’s graph is shown in Fig. 4: all edges shown in the figure are bi-directional, and the number shown next to each edge is the weight of that edge. The measurement nodes are Nodes \{6, 7, 10\}, while the actuation nodes are Nodes \{1, 2, 3, 4, 5\}. In this example we further assume that only the states of the Nodes \{1, 2, 3, 4, 5, 6, 7\} are accessible to the control authorities. Our goal is to
design an observability-blocking controller that only uses the states of these accessible nodes. Here, we have deliberately constructed the network so that a naive design using Theorem 5 leads to instability, and hence a time-scale separation-based design is needed.

In this example, Vertex 5 is a single vertex-cutset which partitions the network graph as required for the regional controller design: specifically, it forms a partition with only accessible vertices, which contains the possible actuation vertices and no measurement vertices. The design depends on the construction of accessible-region synchronization network model on Nodes \{1, 2, 3, 4, 5, 6, 7\} with the measurement node as Node 5 and two actuation nodes, as described in Section IV.C. (The graph is undirected, hence \(q = |V_{cyl}| + 1\) actuation nodes suffices.) We select Nodes 1 and 2 as the actuator nodes; the accessible-region synchronization model can be checked to be controllable. The state matrix for the synchronization dynamics of this model is given by the following Laplacian matrix \(\tilde{L}\).

\[
\tilde{L} = \begin{bmatrix}
-22 & -10 & -12 & 0 & 0 & 0 & 0 \\
-10 & 21 & 0 & -11 & 0 & 0 & 0 \\
-12 & 0 & 24 & -12 & 0 & 0 & 0 \\
0 & -11 & -12 & 53 & -30 & 0 & 0 \\
0 & 0 & 0 & -30 & 59 & -15 & -14 \\
0 & 0 & 0 & 0 & -15 & 15 & 0 \\
0 & 0 & 0 & 0 & -14 & 0 & 14 \\
\end{bmatrix}
\]

We first design the regional state feedback observability-blocking controller based on Theorem 5. For the design, we choose the eigenvalue \(\lambda_p = 14.4812\) of \(\tilde{L}\), and then apply Algorithm 1 to obtain the gain matrix \(\tilde{F}\) that blocks observability in the accessible-region synchronization network model. Per Theorem 5, the gain \(F = [\tilde{F} \ 0]\) then blocks observability in the base model. However, for this specially constructed example, the closed-loop system has a pair of unstable modes \((0.0548 \pm 2.8411i)\).

Since the naive design does not preserve stability, we use the alternate stability preserving design. To do so, we first use feedback to shift all eigenvalues of the Laplacian \(\tilde{L}\) to have real parts greater than 10, using a standard eigenvalue placement technique. (In fact, we only place eigenvalues at 0 and 6.5855 at 11 and 12 respectively keeping the remaining eigenvalues fixed.) Then we choose the same \(\lambda_p\) as before, and use the developed algorithm to find the gain matrix \(\tilde{F}\) that blocks observability in the accessible-region synchronization network model. We notice that \(F = [\tilde{F} \ 0]\) blocks observability in the base model and also maintains closed-loop stability. Here, the control gains for the controllers at Node 1 and 2 are given by \([0.8659; -5.0798; -11.1017; 79.5732; -91.3091; 13.5638; 12.1537; 0; 0; 0]\) and \([22.6071; 15.6908; 14.2283; 51.1514; 1.6000; 45.4956; 45.3810; 0; 0; 0]\). It should be noted that the last 3 entries of the controller gains of the both controllers are zero, and thus the states of the inaccessible nodes, i.e. Nodes \{8, 9, 10\}, are not used by the feedback controller.

VI. Conclusions and Future work

Designs for control systems in network synchronization processes are presented, which prevent observers at remote nodes from having full visibility of the state dynamics. These designs exploit eigenstructure assignment techniques to block observability, while also preserving much of the system’s eigenstructure. Sparser control schemes that exploit the network’s graph topology, as well as feedback schemes which only require regional data, have also been developed. These sorts of design algorithms may be valuable for network-control applications where security and privacy needs are paramount, since they can allow operators to keep adversaries and other stakeholders from having full visibility of the dynamics. From a methodological standpoint, our study begins to address design in networks with multiple orthogonal control authorities, by showing how one controller can be designed to shape remote channels or input-output processes.

A number of directions of further work are of interest including: 1) generalizing the algorithm to more complex network models, 2) achieving sparser designs by allowing some further flexibility in the eigenvalue/eigenvector placements, and 3) developing algorithms that block particular critical statistics from being estimated.

Acknowledgements

The authors gratefully acknowledge the support of the United States National Science Foundation under grants CNS-1545104 and CMMI-1635184.

REFERENCES

[1] Liu YY, Slotine JJ, Barahosi AL, "Controllability of complex networks", Nature, vol. 473, pp. 167-173, 2011.
[2] Rahmani A, Ji M, Mesarah M, Egerstedt M, "Controllability of multi-agent systems from a graph-theoretic perspective", SIAM Journal on Control and Optimization, Vol. 48, no. 1, pp. 162-186, 2009.
[3] Pasqualetti F, Zampieri S, Bullo F, "Controllability metrics, limitations and algorithms for complex networks", IEEE Transactions on Control of Network Systems, vol. 1, pp. 40-52, 2014.
[4] Wang L, Chen G, Wang X, Tang WK, "Controllability of networked MIMO systems", Automatica, vol. 69, pp. 405-409, 2016.
[5] Xue M, Roy S, "Input-output properties of linearly-coupled dynamical systems: Interplay between local dynamics and network interactions", IEEE 56th Annual Conference on Decision and Control, pp. 487-492, 2017.
[6] Dhal R, Roy S, "Vulnerability of network synchronization processes: A minimum energy perspective", IEEE Transactions on Automatic Control, vol. 61, no. 9, pp. 2525-2530, 2016.
[7] Summers TH, Cortesi FL, Lygeros J, "On Submodularity and controllability in complex dynamical networks", IEEE Transactions on Control of Network Systems, vol. 3, no. 1, pp. 91-101, 2016.
[8] Xue M, Yeung E, Rai A, Roy S, Wan Y, Warnick S, "Initial-condition estimation in network synchronization processes: algebraic and graphical characterizations of the estimator", AIAA Guidance, Navigation, and Control Conference, p. 6398, 2011.

[9] Xue, Mengran, Wei Wang, and Sandip Roy, "Security concepts for the dynamics of autonomous vehicle networks", Automatica, vol. 50, no. 3, pp. 852-857, 2014.

[10] Sundaram S, Hadjicostis CN, "Structural controllability and observability of linear systems over finite fields with applications to multi-agent systems", IEEE Transactions on Automatic Control, vol. 58, no. 1, pp. 60-73, 2013.

[11] Abad Torres J, Roy S, "Graph-theoretic characterizations of zeros for the input-output dynamics of complex network processes", International Journal of Control, vol. 87, no. 5, pp. 940-950, 2014.

[12] Roy S, Torres JA, Xue M, "Sensor and actuator placement for zero-shaping in dynamical networks", IEEE 55th Conference on Decision and Control, pp. 1745-1750, 2016.

[13] Summers TH, Lygeros J, "Optimal sensor and actuator placement in complex dynamical networks", IFAC Proceedings Volumes, vol. 47, no. 3, pp. 3784-3789, 2014.

[14] Sridhar, Siddharth, Adam Hahn, and Manimaran Govindarasu. "Cyber-Physical System Security for the Electric Power Grid", Proceedings of the IEEE, vol. 100, no. 1 pp. 210-224, 2012.

[15] Liu W, Kwon C, Aljanabi I, Hwang I, "Cyber security analysis for state estimators in air traffic control systems", AIAA Guidance, Navigation, and Control Conference, pp. 4929, 2012.

[16] Chetty V, Woodbury N, Vaziripour E, Warnick S, "Vulnerability analysis for distributed and coordinated destabilization attacks", IEEE 53rd Annual Conference on Decision and Control, pp. 511-516, 2014.

[17] Xue M, Wang W, Roy S, "Security concepts for the dynamics of autonomous vehicle networks", Automatica, vol. 50, no. 3, pp. 852-857, 2014.

[18] Ren W, Beard RW, "Consensus seeking in multiagent systems under dynamically changing interaction topologies", IEEE Transactions on automatic control, vol. 50, no. 5, pp. 655-661, 2005.

[19] Xiao L, Boyd S, Kim SJ, "Distributed average consensus with least-mean-square deviation", Journal of parallel and distributed computing, vol. 67, no. 1, pp. 33-46, 2007.

[20] Xiao L, Boyd S, "Fast linear iterations for distributed averaging", Systems and Control Letters, vol. 53, no. 1, pp. 65-78, 2014.

[21] Yang T, Roy S, Wan Y, Saberi A, "Constructing consensus controllers for networks with identical general linear agents", International Journal of Robust and Nonlinear Control, vol. 21, no. 11, pp. 1237-1250, 2011.

[22] Roy S, "Scaled consensus", Automatica 51, pp. 259-262, 2015.

[23] Moore B, "On the flexibility offered by state feedback in multivariable systems beyond closed loop eigenvalue assignment", IEEE Trans. on Automatic Control, pp. 689-692, October 1976.

[24] Klein G, Moore B, "Eigenvalue-generalized eigenvector assignment with state feedback", IEEE Trans. on Automatic Control, vol. 22,1, pp. 140-141, 1977.

[25] Manuf AA, Roy S, "A Comment on Surgical Eigenstructure Assignment via State Feedback".

[26] Trudnowski DJ, Smith JR, Short TA, Pierre DA, "An application of Prony methods in PSS design for multimachine systems", IEEE Transactions on Power Systems, vol. 6, no. 1, pp. 118-126, 1991.

[27] Wan Y, Roy S, Saberi A, "Network design problems for controlling virus spread", 46th IEEE Conference on Decision and Control 2007, pp. 3925-3932, IEEE, 2007.

[28] Yu W, Chen G, Lu J, Kurths J, "Synchronization via pinning control on general complex networks", SIAM Journal on Control and Optimization, vol. 51, no. 2, pp. 1395-416, 2013.

[29] Rugh WJ, Rugh WJ, Rugh WJ, Linear system theory, Upper Saddle River, NJ: prentice hall, 1996.

[30] Fahmy M, O’reilly J, "Eigenstructure assignment in linear multivariable systems—A parametric solution", IEEE Transactions on Automatic Control, vol. 28,10, pp. 990-994, 1983.

[31] Manuf AA, Roy S, "Observability-Blocking Controllers for Network Synchronization Processes", American Control Conference 2019, pp. 2066-2071, 2019.

[32] DeMarco CL, Sarashkhar JV, Alvarado F, "The potential for malicious control in a competitive power systems environment", IEEE International Conference on Control Applications, pp. 462-467, 1996.

[33] Youssef MG, Kimura M, "A modal control method for designing a decentralized controller to stabilize power systems", IFAC Proceedings Volumes, vol. 14, pp. 1251, 1981.

[34] Nieto-Wire C, Sobel K, "Delta operator eigenstructure assignment for reconfigurable control of a tailless aircraft", Journal of Guidance, Control, and Dynamics, vol. 37, pp. 1824, 2014.

[35] Xia W, Cao M, "Analysis and applications of spectral properties of grounded Laplacian matrices for directed networks", Automatica, vol. 80, pp. 10-16, 2017.

[36] Del Genio CI, Gross T, Bassler KE, “ All Scale-Free Networks are Sparse”, Physical review letters, vol. 107, no. 17, p. 178701, 2011.

[37] Chartrand G, Zhang P, A first course in graph theory, Courier Corporation, 2013.

[38] Kautsky, Jaroslav and Nichols, Nancy K and Van Dooren, Paul, "Robust pole assignment in linear state feedback", International Journal of control, vol. 41, no. 5, pp. 1129-1155, 1985.

[39] A Varga, "A Schur method for pole assignment", IEEE Transactions on Automatic Control, vol. 26, no. 2, pp. 517, 1981.

[40] Miminis, George S and Paige, C Chris, "An algorithm for pole assignment of time invariant linear systems", International Journal of Control, vol. 35, no. 2, pp. 341-354, 1982.

[41] Kokotovic, Petar V and O’Malley Jr, Robert E and Sannuti, Peddapullaiah, "Single perturbations and order reduction in control theory—an overview", Automatica, vol. 12, no. 2, pp. 123-132, 1976.