Model Reduction Methods for Complex Network Systems

X. Cheng¹, and J.M.A. Scherpen²

¹Control Systems Group, Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, the Netherlands; email: x.cheng@tue.nl
²Jan C. Willems Center for Systems and Control, ENTEG, University of Groningen, the Netherlands; email: j.m.a.scherpen@rug.nl

Keywords
reduced-order modeling, network systems, interconnected systems, multi-agent systems, graph clustering, synchronization, semistability, structure-preserving

Abstract
Network systems consist of subsystems and their interconnections, and provide a powerful framework for analysis, modeling and control of complex systems. However, subsystems may have high-dimensional dynamics, and the amount and nature of interconnections may also be of high complexity. Therefore, it is relevant to study reduction methods for network systems. An overview on reduction methods for both the topological (interconnection) structure of the network and the dynamics of the nodes, while preserving structural properties of the network, and taking a control systems perspective, is provided. First topological complexity reduction methods based on graph clustering and aggregation are reviewed, producing a reduced-order network model. Second, reduction of the nodal dynamics is considered by using extensions of classical methods, while preserving the stability and synchronization properties. Finally, a structure-preserving generalized balancing method for simplifying simultaneously the topological structure and the order of the nodal dynamics is treated.
1. INTRODUCTION

The backbone of many modern technological systems is a network system (system of systems), which bonds diverse multi-physics components together. Many large-scale systems can be modeled as network systems which are composed of multiple subsystems interacting with each other via certain coupling protocols. Such systems are becoming ever more prevalent in various domains. Chemical reaction chains, cellular and metabolic networks, social networks, multi-robot coordination, and large-scale power grids are only a few examples.

However, with the increasing complexity of network scales and subsystem dynamics, the models describing the behavior of network systems can be of extremely high dimension. This will lead to serious scalability issues in simulation, optimization, transient analysis, and control synthesis due to limited computational capability and storage capacity. These issues spur the development of methodologies on complexity reduction for large-scale network systems, aiming to acquire pertinent information of the systems in a computationally manageable fashion.

In the past few decades, a variety of theories and techniques for model reduction have been developed for generic dynamical systems, including Krylov-subspace methods (also known as moment matching), balanced truncation, and Hankel norm approximation, see (9, 10, 11, 12, 13, 14) and the references therein. These conventional methodologies can generate reduced-order models that well approximate the input-output mapping of a high-dimensional system. However, when addressing the model reduction problem of large-scale network system, we have to rethink about how to implement those methods in a structure-preserving manner. This is because analysis, control and monitoring of complex networks rely heavily on their interconnection structure (15, 16, 17, 18). Actually, preserving essential network configurations in the approximation of network systems presents the most challenging problem. Early work on controller reduction can be viewed as a predecessor of structure-preserving reduction of interconnected systems, which takes into account the coupling structure between plants and controllers (19, 20). However, recent developments in large-scale networked systems have gone far beyond the simple closed-loop structure.

In this paper, we provide an overview of recent advances in dimension reduction of complex network systems. The complexity we consider consists of two aspects, namely, large-scale topology and high-dimensional subsystems (nodal dynamics), which lead to two types of model reduction problems in the context of network systems. The first one is focused on how to simplify a complicated network structure by reducing the number of nodes. Inspired by the classification and pattern recognition in data science and computer graphics (21, 22), reduction methods based on clustering and aggregation are mainstream for reducing the topological complexity. Most of the relevant work (23, 24, 25, 26, 27, 28, 29, 30, 31) is treated in this paper. Besides, we also briefly review other topological methods, including the well-known singular perturbation approximation. The second problem considers how to reduce the dimension of individual subsystems in a network. The relevant approximation approaches for interconnected systems or coupled systems based on subsystem structuring have been of interest already for a long time (32, 33, 34). Recent developments in (35, 36) further discuss diffusively coupled linear/nonlinear systems, where the reduction is performed on each subsystem in a way that certain properties of the entire network, such as synchronization and stability are retained. Furthermore, techniques in (37, 38) combine the complexity reduction of network structures and subsystem dynamics, also providing an attractive way to simplify the complexity of entire network systems.
2. FROM GRAPHS TO NETWORK SYSTEMS

In this section, we recapitulate some preliminaries on algebraic graph theory, and provide key concepts to model, analyze and design network systems. The graph-based modeling of network systems is then introduced. We refer to e.g., [39, 40] for more details.

2.1. Algebraic Graph Theory

The language of graphs is essential in modeling and control of networked systems, and it provides a natural tool for characterizing the interconnection structure of a network. Any finite graph \( G \) can be featured by a finite and nonempty node set \( \mathcal{V} := \{1, 2, ..., n\} \) and an edge set \( \mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} \). Depending on whether the edges have specific orientations, we have two basic categories of graphs:

2.1.1. Directed Graphs. Directed graph (in short, a digraph) structures are found in various applications, including biochemical reactions and social networks, see e.g., [4, 5], where the transmission of information or energy among network nodes is directional. This directionality can be encoded in the edges, which are ordered pairs of elements of \( \mathcal{V} \), and we say that there is an edge directed from node \( i \) to node \( j \) if \((i, j) \in \mathcal{E}\). A digraph \( G \) is called simple, if it does not contain self-loops (i.e., \( \mathcal{E} \) does not contain edges of the form \((i, i)\), \( \forall i \)), and there exists exactly one edge directed from \( i \) to \( j \) if \((i, j) \in \mathcal{E}\). In a simple digraph \( G \), a node \( i_0 \) is reachable from another node \( i_0 \) if there is a directed path from \( i_0 \) to \( i_n \). Here, this path is defined as a sequence of edges of the form \((i_k, i_{k+1})\), \( k = 1, ..., n \), which joins a sequence of distinct nodes \( i_0, i_1, ..., i_n \).

Next, the connectivity notions for a digraph \( G \) are presented. (i) \( G \) is strongly connected if any two nodes are reachable from each other; (ii) \( G \) is quasi strongly connected if all the nodes are reachable from a common node; (iii) \( G \) is weakly connected if its undirected version \( \bar{G} = (\mathcal{V}, \mathcal{E}_u) \) is strongly connected, where the set \( \mathcal{E}_u \supseteq \mathcal{E} \) include both \((i, j)\) and \((j, i)\), if there is an edge \((i, j) \in \mathcal{E}\). Note that any simple digraph \( G \) can be decomposed into a unique set of maximal strongly connected components (SCCs), which are the largest strongly connected subgraphs of \( G \). If an SCC has only outflows to other SCCs, it is called a root SCC (RSCC). A weakly connected digraph may contain multiple RSCCs, while a quasi strongly connected digraph has only one RSCC.

There are three matrices commonly used to characterize the topology of a digraph. The \textit{incidence matrix} \( \mathbf{B} \in \mathbb{R}^{n \times |\mathcal{E}|} \) of \( G \) is defined such that \( \mathbf{B}[i, j] = 1 \), if edge \((i, j) \in \mathcal{E}\); \( \mathbf{B}[i, j] = -1 \), if edge \((j, i) \in \mathcal{E}\); and \( \mathbf{B}[i, j] = 0 \) otherwise, where each column indicates a directed edge. While this edge is assigned a positive value (weight), i.e., \( G \) is weighted, we define a \textit{weighted adjacency matrix} \( \mathbf{A} \), where \( \mathbf{A}[i, j] \) is equal to the weight of the edge \((j, i) \) if \((j, i) \in \mathcal{E} \), and \( \mathbf{A}[i, j] = 0 \) otherwise. Moreover, the weighted out-degree and in-degree matrices of \( G \) are the diagonal matrices defined by \( D_{\text{out}} := \text{Diag}(\mathbf{A}) \) and \( D_{\text{in}} := \text{Diag}(\mathbf{A}^\top) \), respectively. A strongly connected digraph is called \textit{balanced}, if \( D_{\text{out}} = D_{\text{in}} \). The \textit{Laplacian matrix} of a digraph \( G \) is defined as \( \mathbf{L} := D_{\text{out}} - \mathbf{A} \), and the elements of \( \mathbf{L} \) are given by

\[
[L]_{ij} = \begin{cases} 
\sum_{j=1, j \neq i}^{n} \mathbf{A}_{ij}, & i = j \\
\mathbf{A}_{ij}, & \text{otherwise.}
\end{cases}
\]

Laplacian matrices are instrumental in modeling various diffusion processes, e.g., [11, 12, 13]. A Laplacian matrix enjoys two fundamental properties: (i) \( \mathbf{L} \mathbf{1} = 0 \); (ii) \([L]_{ii} \geq 0, \forall i \in \mathcal{V}\), and \([L]_{ij} \leq 0, \forall i \neq j\). Conversely, a real square matrix satisfying the two properties can
also be interpreted as a Laplacian matrix that represents a weighted simple digraph. Note that Laplacian matrices are singular. If a weakly connected digraph has \( m \) LSCCs, then its Laplacian matrix also has semisimple zero eigenvalues with multiplicity \( m \), while all the other nonzero eigenvalues have positive real parts.

2.1.2. Undirected Graphs. Undirected graphs are commonly used to characterize interconnection structure of physical systems, e.g., power grids, RC circuits, and mass-damper systems. An undirected graph can be viewed as a special digraph, whose weighted adjacency matrix \( A \) (or Laplacian matrix \( L \)) is symmetric. In this case, we can define a Laplacian matrix using an alternative formula:

\[
L = BWB^T,
\]

where \( B \) is the incidence matrix obtained by assigning an arbitrary orientation to each edge of \( G \), and \( W := \text{Diag}(w_1, w_2, \cdots, w_{|E|}) \) with \( w_k \) the weight associated to the edge \( k \), for each \( k = 1, 2, \ldots, |E| \). If \( G \) is an undirected connected graph, the Laplacian matrix \( L \) has the properties: (i) \( L^\top = L \) and \( \ker L = \text{span}\{1\} \); (ii) \( [L]_{ij} \leq 0 \) if \( i \neq j \), and \( [L]_{ii} > 0 \).

2.2. Modeling of Network Systems

In the field of network science, the evolution of network topology over time is often specified as dynamics of networks (45). Differently, for control systems, dynamics over networks is of interest, where nodes represent individual dynamical systems that are coupled through edges (46). We use the latter notion when referring to a network system. Network system examples are chemical reaction networks, power grids, robotic networks, i.e., they have a clear interconnection structure, physically or virtually. Additionally, network modeling is also applicable to spatially discretized systems that are originally described by PDE’s, such as simple beam models or fluid dynamical systems.

2.2.1. Networks of Single-Integrators. The simplest network systems consider all the nodes being just single-integrators, namely, \( \dot{x}_i(t) = v_i(t) \), where \( x_i(t), v_i(t) \in \mathbb{R} \) are the state and input of node \( i \). A digraph \( G \) then captures the interconnection topology of \( n \) single-integrators, where the coupling rule is

\[
v_i(t) = -d_i x_i(t) + \sum_{j=1, j \neq i}^{n} [A]_{ij} x_j(t).
\]

In Equation 3, \( d_i \in \mathbb{R} \) represents the state feedback gain, and \( A \) is the weighted adjacency matrix of \( G \), whose entry \( [A]_{ij} \) indicates the strength of the coupling between nodes \( i \) and \( j \). Taking into account the external control signals \( u(t) \in \mathbb{R}^p \) and measurements \( y(t) \in \mathbb{R}^q \) of the network, we then derive a compact form for the network system as

\[
\dot{x}(t) = \Gamma x(t) + Fu(t), \quad y(t) = Hx(t),
\]

where \( \Gamma := A - D \) with \( D = \text{Diag}(d_1, \ldots, d_n) \), and \( F \in \mathbb{R}^{n \times p} \), \( H \in \mathbb{R}^{q \times n} \) are the input and output matrices, respectively.

Equation 4 is regarded as a rather general representation for single-integrator networks, whose stability depends on the values in \( D \) and \( A \). If \( D \geq 0 \), \( \Gamma \) becomes a Metzler matrix, leading to the concept of monotone systems or positive systems (47, 48, 49). Particularly,
if we choose $D > D_{\text{out}}$ or $D > D_{\text{in}}$, then the Metzler matrix $\Gamma$ is strictly row (column) diagonally dominant. Following the Gershgorin circle theorem (50), $\Gamma$ is Hurwitz, leading to the asymptotic stability of the network system. If $D = D_{\text{in}}$, or equivalently $\Gamma = -L^T$ with $L$ the Laplacian matrix of $G$, we have a network flow model (5, 42). Furthermore, if $D = D_{\text{out}}$, i.e., $\Gamma = -L$, Equation 4 becomes a consensus network, or continuous-time averaging systems (39). The coupling rule in Equation 3 becomes

$$v_i(t) = -\sum_{j=1,j \neq i}^{n} [A]_{ij} [x_i(t) - x_j(t)],$$

which is known as the diffusive coupling rule. For both $\Gamma = -L^T$ and $\Gamma = -L$, the system in Equation 4 is semistable (or semi-convergent), i.e., $\lim_{t \to \infty} e^{t\Gamma}$ exists for any initial condition $x(0)$. Particularly, when $G$ is strongly connected, then $\lim_{t \to \infty} e^{-Lt} = \mathbb{1}\omega^T$, with $\omega$, satisfying $\mathbb{1}^T \omega = 1$, the left eigenvector of $L$ for eigenvalue 0.

**Definition 1.** A network system $\dot{x}(t) = \Gamma x(t)$ achieves synchronization if

$$\lim_{t \to \infty} [x_i(t) - x_j(t)] = 0, \quad \forall \ i, j \in \mathcal{V},$$

holds for all initial condition $x(0)$.

The approximation of the network system [4] aims for a reduced network consisting of a fewer number of nodes that captures essential properties of the original network. Specifically, a model reduction problem (Figure 1) is formulated to find a reduced-order model

$$\dot{\hat{x}}(t) = \hat{\Gamma} \hat{x}(t) + \hat{F}u(t), \quad \hat{y}(t) = \hat{H} \hat{x}(t),$$

where $\hat{x} \in \mathbb{R}^r (r < n)$, $\hat{y} \in \mathbb{R}^q$ such that (i) $\hat{\Gamma} \in \mathbb{R}^{r \times r}$ is interpretable as a reduced graph, and (ii) the approximation error is minimized between the original and the reduced-order models. The approximation error is usually evaluated by the $\mathcal{H}_\infty$ or $\mathcal{H}_2$ norms of $\eta(s) - \hat{\eta}(s)$,

$$\eta(s) := H(sI_n - \Gamma)^{-1}F, \quad \hat{\eta}(s) := \hat{H}(sI_r - \hat{\Gamma})^{-1}\hat{F}.$$

### 2.2.2. Networked Linear Systems.

The network model in Equation 4 can be extended beyond single-integrators to consider each node as a high-order linear subsystem as

$$\dot{x}_i(t) = A_i x_i(t) + B_i v_i(t), \quad y_i(t) = C_i x_i(t),$$
where \( x_i \in \mathbb{R}^k \), \( v_i \in \mathbb{R}^{n_i} \), and \( y_i \in \mathbb{R}^{\mu_i} \) are internal states, inputs and outputs, respectively. Suppose that \( n \) subsystems are interconnected through the relations: 

\[
v_i(t) = \sum_{j=1}^n K_{ij} y_j(t) + F_i u(t), \quad y_j(t) = \sum_{i=1}^n H_i y_i(t),
\]

with \( K_{ij} \in \mathbb{R}^{n_i \times \mu_j} \) the coupling coefficient between nodes \( i \) and \( j \), where \( K_{ij} = 0 \) if and only if there are no signals passing from \( j \) to \( i \). The vectors \( u(t) \) and \( y(t) \) are denoted as external inputs and outputs. Combining this with Equation 9, we obtain a compact representation of the overall network system (32, 34):

\[
\dot{x}(t) = (A_n + B_n \Gamma C_n) x(t) + B_n F u(t), \quad y(t) = H C_n x(t),
\]

where \( A_n := \text{blkdiag}(A_1, \ldots, A_n) \), \( B_n := \text{blkdiag}(B_1, \ldots, B_n) \), \( C_n := \text{blkdiag}(C_1, \ldots, C_n) \), and

\[
\Gamma = \begin{bmatrix} K_{11} & \cdots & K_{1n} \\ \vdots & \ddots & \vdots \\ K_{n1} & \cdots & K_{nn} \end{bmatrix}, \quad F = \begin{bmatrix} F_1 \\ \vdots \\ F_n \end{bmatrix}, \quad H = \begin{bmatrix} H_1 & \cdots & H_n \end{bmatrix}.
\]

An example of a networked linear system containing six subsystems is shown in Figure 2. Networked linear systems of the form Equation 10 are also known as interconnected or coupled systems (32, 34). The subsystems in Equation 9 could have different dynamics, in which sense the network is called heterogeneous.

Homogeneous networks are defined when the dynamics of each node is identical:

\[
\dot{x}_i(t) = A x_i(t) + B v_i(t), \quad y_i(t) = C x_i(t),
\]

where \( x_i \in \mathbb{R}^k \), \( v_i \in \mathbb{R}^m \) and \( y_i \in \mathbb{R}^m \) are the internal state, input and output of node \( i \), respectively. Under a simple static output feedback interconnection similar to Equation 3, the dynamics of networked homogeneous linear systems is presented in compact form as

\[
\Sigma : \begin{cases} 
\dot{x}(t) = (I \otimes A + \Gamma \otimes BC) x(t) + (F \otimes B) u(t), \\
y(t) = (H \otimes C) x(t),
\end{cases}
\]

with joint state vector \( x(t) \in \mathbb{R}^{k \times n} \), external control inputs \( u(t) \in \mathbb{R}^{m \times n} \) and external outputs \( y(t) \in \mathbb{R}^{n \times m} \). The matrix \( \Gamma \in \mathbb{R}^{n \times n} \) indicates how the subsystems are interconnected.

A commonly studied network system in the form of Equation 12 is diffusively-coupled linear systems, where \( \Gamma = -L \) is the Laplacian matrix of the underlying graph. Thus, the coupling rule among the nodes becomes \( v_i(t) = -\sum_{j=1,j \neq i}^n A_{ij} [y_i(t) - y_j(t)] \). In this setting, the synchronization problem of the system \( \Sigma \) has been intensively studied in the literature, see e.g., (33, 34, 32, 31, 30).
Theorem 1. Consider a network of diffusively-coupled homogeneous linear systems described by Equation 12 with the symmetric Laplacian \( L \). Let \( 0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n \) denote the eigenvalues of \( L \). Then, the network system 12 achieves synchronization if and only if \( A - \lambda_i BC \) is Hurwitz for all \( i \in \{2, 3, \ldots, n\} \).

A sufficient condition for synchronization is provided by assuming that the subsystem \( A, B, C \) is passive, i.e., there exists a symmetric positive definite matrix \( K > 0 \) that verifies \( A^T K + KA \leq 0 \), and \( C^T = BK \).

Passivity is a natural property of physical systems, including mechanical systems, electrical networks, and thermodynamic systems [53]. With passivity we obtain a synchronization condition that is independent from the spectrum of the graph Laplacian [54, 55].

Theorem 2. Let \( \Gamma = -L \) represent any connected undirected graph or any strongly connected digraph. If the subsystem \((A, B, C)\) in Equation 11 is passive and observable, then the network system 12 achieves synchronization.

The model complexity of networked linear systems comes from two aspects: the dimension of subsystems and the topological scale of the network. The first reduction problem is thus to reduce each subsystem (or a subset of them) by taking into account the coupling structure in order to approximate the entire network system. For the reduction of heterogeneous network system 10 the objective is to construct a network model composed of reduced-order subsystems \((\hat{A}_i, \hat{B}_i, \hat{C}_i)\), yielding an approximation of the entire system with the same form as 10 where \( \hat{A}_n := \text{blkdiag}(\hat{A}_1, \ldots, \hat{A}_n) \), \( \hat{B}_n := \text{blkdiag}(\hat{B}_1, \ldots, \hat{B}_n) \), \( \hat{C}_n := \text{blkdiag}(\hat{C}_1, \ldots, \hat{C}_n) \). The matrices \( \Gamma, H, \) and \( F \) remain the same as the original one. Homogeneous network systems can be reduced in a similar manner such that each original subsystem \((A, B, C)\) is replaced by a lower-order approximation \((\hat{A}, \hat{B}, \hat{C})\):

\[
\begin{aligned}
\dot{\hat{x}}(t) &= \left( I_n \otimes \hat{A} - \Gamma \otimes \hat{B}\hat{C} \right) \hat{x}(t) + (F \otimes B)u(t), \\
\hat{y}(t) &= (H \otimes \hat{C})\hat{x}(t).
\end{aligned}
\]

The second reduction problem is focused on a simplification of the graph structure, as illustrated in Figure 1. A resulting reduced-order model for networked homogeneous linear systems is in the form

\[
\begin{aligned}
\dot{\hat{z}}(t) &= \left( I_r \otimes A - \hat{\Gamma} \otimes BC \right) z(t) + (\hat{F} \otimes B)u(t), \\
\hat{y}(t) &= (\hat{H} \otimes C)z(t),
\end{aligned}
\]

where \( \hat{\Gamma} \in \mathbb{R}^{r \times r} \) represents a reduced graph consisting of \( r \) nodes. Let \( G(s) \) and \( \hat{G}(s) \) be the transfer matrices of the models 12 and 15 respectively. The objective now is to minimize the reduction error \( G(s) - \hat{G}(s) \) with respect to certain norms.

3. REDUCTION OF TOPOLOGICAL STRUCTURES

A powerful paradigm for simplifying a large-scale network is graph clustering. Graph clustering is a process of dividing a set of nodes into nonempty and disjoint subsets, where nodes in each subset are considered related by some similarity measure. Depending on the field, different names are used, including community detection in social networks, and
classification in data science. Furthermore, it is closely related to unsupervised learning in pattern recognition systems \(21\) \(22\). Generally, well-established clustering algorithms (such as hierarchical clustering, spectral clustering or K-means clustering) were developed for static graphs or measured data. In this section, we present a series of clustering-based model reduction techniques for dynamic networks.

### 3.1. Clustering-Based Projection

Consider an LTI system with triplet \((A, B, C)\). The Petrov-Galerkin framework, \(9\) projects the state-space onto a lower dimensional subspace, resulting in a reduced-order model \((V^\top AV, V^\top B, CV)\), where \(V\) is full column rank representing the basis of the subspace, and \(V^\dagger\) is a left inverse of \(V\), i.e., \(V^\dagger V = I\). Clearly, the choice of \(V\) is essential for obtaining the reduced-order model. For structure-preserving model reduction of network systems, \(V\) can be constructed by considering an aggregation of node states.

**Definition 2.** Consider a graph \(G\) with node set \(|V| = n\). Graph clustering of \(G\) is a process that divides \(V\) into \(r\) nonempty and disjoint subsets, denoted by \(C_1, C_2, ..., C_r\), where \(C_i\) is called a cluster (or a cell of \(G\)). The characteristic matrix of the clustering \(\{C_1, C_2, ..., C_r\}\) is a binary matrix \(\Pi \in \mathbb{R}^{n \times r}\) with

\[
[\Pi]_{ij} := \begin{cases} 
1, & \text{if node } i \in C_j, \\
0, & \text{otherwise.}
\end{cases}
\]

Note that each row of \(\Pi\) has exactly one nonzero element, indicating that each node is assigned to a unique cluster. The number of nonzero elements in each column is the cardinality of the corresponding cluster. Specifically, \(\Pi_i \Pi = I_n\) and \(\Pi_i \Pi = \sum_{k} |C_k|\) for any given undirected graph Laplacian \(L\), the matrix \(\Pi^\top L \Pi\) is a Laplacian matrix representing an undirected graph of smaller size. This important property allows for structure-preserving model reduction of network systems using \(\Pi\) for the Petrov-Galerkin projection. To construct a reduced-order network system with \(\Pi\) for the Petrov-Galerkin projection. We first have to find a clustering that partitions the nodes of a network into \(r\) clusters.

Consider the network system \(4\) which is assumed to be semistable, see the sidebar on linear semistable systems. Then, the projection matrix is defined as \(VV^\top \in \mathbb{R}^{n \times n}\)

\[
V = N \Pi \in \mathbb{R}^{n \times r}, \quad V^\top := (\Pi^\top M N \Pi)^{-1} \Pi^\top M \in \mathbb{R}^{r \times n},
\]

where \(M\) and \(N\) are nonsingular diagonal weighting matrices \(55\). A reduced-order model is thereby obtained in the form of Equation \(7\) where \(\hat{\Gamma} = V^\top TV, \hat{F} = V^\top F, \text{ and } \hat{H} = HV\).

**Theorem 3.** \(56\) Let \(\eta(s)\) and \(\hat{\eta}(s)\) be the transfer matrices of the original and reduced-order models in Equations \(4\) and \(7\). Then, a bounded \(H_2\) reduction error is guaranteed, i.e., \(\eta(s) - \hat{\eta}(s) \in H_2\), if there exist diagonal and positive definite matrices \(M\) and \(N\) such that

\[
(e_i - e_j)^\top N^{-1} J = 0, \quad \text{and } J M^{-1}(e_i - e_j) = 0,
\]

holds for each pair \(i, j \in C_k\), with any \(k \in \{1, 2, \ldots, r\}\).

If \(\Gamma\) is Hurwitz, or \(\Gamma = -L\) with \(L\) the Laplacian of a connected undirected graph, we simply choose \(M = N = I_n\), which always guarantees \(\eta(s) - \hat{\eta}(s) \in H_2\). A Hurwitz \(\Gamma\) implies \(J = 0\), while a connected undirected graph yields \(J = \frac{1}{2} \mathbb{1}_n \mathbb{1}_n^\top\). In \(19\) \(59\) \(60\) dynamic networks having a strongly connected topology are treated, i.e., \(\Gamma\) is irreducible.
Linear Semistable Systems and Pseudo Gramians

Semistability is a more general concept than asymptotic stability as it allows for multiple poles that are zero. The systems’ trajectories thus may converge to a nonzero Lyapunov stable equilibrium (57, 58). Specifically, a linear system $\dot{x}(t) = Ax(t)$ is semistable if $\lim_{t \to \infty} e^{At}$ is non-zero and exists for all initial states $x(0)$, or equivalently, the zero eigenvalues of $A$ are semisimple, and all the other eigenvalues have negative real parts.

It is well-known that the standard controllability and observability Gramians, (9), are not well-defined for a semistable system. Therefore, in (56) the definition of pseudo Gramians is presented. Consider a linear semistable system $(A,B,C)$. The pseudo controllability and observability Gramians are defined as

$$P = \int_0^\infty (e^{At} - \mathcal{J})BB^\top(e^{At} - \mathcal{J})^\top dt, \quad Q = \int_0^\infty (e^{At}T - \mathcal{J}^\top)C^\top C(e^{At} - \mathcal{J})dt,$$

respectively, where $\mathcal{J} := \lim_{t \to \infty} e^{At}$ is a constant matrix. The pseudo Gramians $P$ and $Q$ in Equation (19) are well-defined for semistable systems. The pseudo Gramians can be computed as $P = \hat{P} - \mathcal{J}\hat{P}\mathcal{J}^\top$ and $Q = \hat{Q} - \mathcal{J}^\top\hat{Q}\mathcal{J}$, where $\hat{P}$ and $\hat{Q}$ are arbitrary symmetric solution of the Lyapunov equations

$$A\hat{P} + \hat{P}A^\top + (I - \mathcal{J})BB^\top(I - \mathcal{J}^\top) = 0, \quad A^\top\hat{Q} + \hat{Q}A + (I - \mathcal{J}^\top)C^\top C(I - \mathcal{J}) = 0,$$

respectively. The pseudo Gramians are useful for computing the $\mathcal{H}_2$ norm of a semistable system. The transfer matrix $G(s) \in \mathcal{H}_2$ if and only if $CJB = 0$. Furthermore, $\|G(s)\|_{\mathcal{H}_2}^2 = \text{Tr}(CPC^\top) = \text{Tr}(B^\top QB)$.

and has only one zero eigenvalue with corresponding left and right eigenvectors $\mu_l$ and $\mu_r$ the so-called Frobenius eigenvectors with all real and positive entries (61). In this case $M = \text{Diag}(\mu_l)$ and $N = \text{Diag}(\mu_r)$ satisfy Equation (18) for any clustering. Furthermore, we have $\eta(s) - \hat{\eta}(s) \in \mathcal{H}_2$ and $\mathcal{N}M + M\mathcal{N}N \leq 0$. Following (56), we can obtain a posteriori bound on the reduction error as $\|\eta(s) - \hat{\eta}(s)\|_{\mathcal{H}_2} \leq \gamma_s \sqrt{\text{Tr}(I - \mathcal{V}\mathcal{V}^\top)\mathcal{P}(I - \mathcal{V}\mathcal{V}^\top)^\top}$, where $\mathcal{P}$ is the pseudostability Gramian of system (4) and $\gamma_s \in \mathbb{R}_{>0}$ satisfies

$$\begin{bmatrix}
\mathcal{N}\mathcal{G}M + M\mathcal{G}N & \mathcal{N}\mathcal{G}(I - \mathcal{J}^\top)\mathcal{H}^\top \\
\mathcal{G}^\top M & -\gamma_s I \\
\mathcal{H}(I - \mathcal{J}) & \mathcal{H}^\top & -\gamma_s I
\end{bmatrix} \leq 0.$$  

There is a balanced graph representation of the digraph system (4) as follows:

$$MN\hat{\xi}(t) = L_\phi \hat{\xi}(t) + MFu(t), \quad y(t) = HN\hat{\xi}(t),$$

where $L_\phi := M\mathcal{G}N$ is the Laplacian matrix of the balanced digraph, and the resulting reduced-order model in Equation (7) becomes

$$\Pi^\top M\Pi\hat{\xi}(t) = \Pi^\top L_\phi \Pi\hat{\xi}(t) + \Pi^\top MFu(t), \quad \hat{y}(t) = H\Pi\hat{\xi}(t),$$

with $\Pi^\top L_\phi \Pi$ representing a reduced balanced digraph. In (62), a generalized balanced digraph is defined as a weakly connected digraph in which each RSOC is balanced while removing all the non-RSOC nodes resulting in a generalized balanced graph representation.
similar to Equation 22. For networks with a weakly connected topology the error system generally $\eta(s) - \hat{\eta}(s) \notin \mathcal{H}_2$. Then, clusterability is defined between two nodes $i, j$ if they satisfy Equation 18. Clusterability of all nodes in each cluster then guarantees the stability of the error $\eta(s) - \hat{\eta}(s)$ [62].

**Example 1.** Consider the mass-damper system in Figure 3a, where the masses are interconnected via linear dampers. $u_1(t)$, $u_2(t)$ represent external forces, and $y_1(t)$ and $y_2(t)$ are measured velocities. Suppose that all the masses are identical, the network system in the form of Equation 4 is obtained as

$$\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\dot{x}_3 \\
\dot{x}_4 \\
\dot{x}_5
\end{bmatrix} = -\begin{bmatrix}
6 & -3 & 0 & -2 & -1 \\
-3 & 4 & -1 & 0 & 0 \\
0 & -1 & 6 & -2 & -3 \\
-2 & 0 & -2 & 5 & -1 \\
-1 & 0 & -3 & -1 & 5
\end{bmatrix} \begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5
\end{bmatrix} + \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix} \begin{bmatrix}
u_1 \\
u_2 \\
y_1 \\
y_2
\end{bmatrix},$$

where $-\Gamma$ is an undirected graph Laplacian, and the off-diagonal entry $[\Gamma]_{ij}$ represents the damping coefficient of the edge $(i, j)$. Consider $\{C_1, C_2, C_3\} = \{\{1, 2\}, \{3, 5\}, \{4\}\}$ to be the clustering of the graph, which leads to the following characteristic matrix

$$\Pi = \begin{bmatrix}
1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix}^T.$$

Therefore, a reduced-order network model is obtained as

$$\begin{bmatrix}
\dot{\hat{x}}_1 \\
\dot{\hat{x}}_2 \\
\dot{\hat{x}}_3
\end{bmatrix} = -\begin{bmatrix}
4 & -2 & -2 \\
-2 & 5 & -3 \\
-2 & -3 & 5
\end{bmatrix} \begin{bmatrix}
\hat{x}_1 \\
\hat{x}_2 \\
\hat{x}_3
\end{bmatrix} + \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix} \begin{bmatrix}
u_1 \\
u_2 \\
y_1 \\
y_2
\end{bmatrix},$$

where $-\Pi^T \Gamma \Pi$ is again an undirected graph Laplacian. To bring it in the form of Equation 4 we can define $\hat{\Gamma} := (\Pi^T \Pi)^{-1} \Pi^T \Gamma \Pi$, $\hat{F} := (\Pi^T \Pi)^{-1} \Pi^T F$, and $\hat{H} := H \Pi$. However,
from $-\Pi^\top \Gamma \Pi$ it follows that this model allows for a physical interpretation, as shown in Figure 3b: the nodes in each cluster are aggregated into a single node in the reduced network, while all edges connecting nodes from two distinct clusters are merged to a single edge linking the corresponding nodes in the reduced network.

Analogously, and beyond the single integrator case, a reduced-order model of networked homogeneous linear systems in Equation 12 can be formed using the Petrov-Galerkin projection framework of Equation 17, which gives a reduced-order model in the form of Equation 15, where $\hat{\Gamma} := \Pi^\top L \Pi$, $\hat{F} = \Pi^\top F$ and $\hat{H} = \Pi \Pi$. The new state vector $z^\top(t) := [z_1^\top(t) z_2^\top(t) ... z_r^\top(t)] \in \mathbb{R}^{r\ell}$, $z_i(t) \in \mathbb{R}^\ell$, $i = 1, ..., r$ represents an estimate of the state vector of the dynamics of all the nodes in the $i$-th cluster. Note that the extension of clustering-based approaches towards networks of heterogeneous subsystems in Equation 10 remains an open problem. A major challenge lies in the representation of a cluster of nonidentical subsystems.

Denote $G(s)$ and $\hat{G}(s)$ as the transfer matrices of the models 12 and 15, respectively. The analysis of the reduction error $G(s) - \hat{G}(s)$ is more complicated than in the single integrator case, and for general subsystems, the reduction error $G(s) - \hat{G}(s)$ may not be stable. However, there is a theoretical guarantee if the subsystem $(A, B, C)$ in Equation 11 are observable and passive. With Theorem 2 it can be verified that if the original network is undirected, or strongly connected, the reduced-order network system in Equation 15 achieves synchronization, and $G(s) - \hat{G}(s) \in \mathcal{H}_2$, for any clustering $\Pi$ (27, 29).

In the framework of clustering-based projection, the approximation error $\|G(s) - \hat{G}(s)\|_{\mathcal{H}_2}$ only depends on the choice of graph clustering. Thus, the most crucial problem in this framework is how to determine clusters of nodes to minimize the approximation error. In the following, we review several specific cluster selection approaches.

**3.1.1. Almost Equitable Partitions.** Almost equitable partitions provide a graph clustering where nodes in the same cluster are connected to other clusters in a “similar” fashion.

**Definition 3.** Consider a weighted undirected graph $\mathcal{G}$ with adjacency matrix $A$. A clustering $\{C_1, C_2, ..., C_r\}$ is called an almost equitable partition if for any indexes $\mu, \nu \in \{1, 2, ..., r\}$ with $\mu \neq \nu$, it holds that $\sum_{k \in C_\nu} w_{ik} = \sum_{k \in C_\mu} w_{jk}, \forall i, j \in C_\mu$, where $w_{ij} := [A]_{ij}$.

In Figure 4 an example of the almost equitable partition of an undirected graph is shown. (25). The nodes in a cluster have the same total edge weight to other clusters. An almost equitable partition of an undirected graph has the key property that $\text{Im}(\Pi)$ is $L$-
invariant, i.e., \( L \text{ Im}(\Pi) \subseteq \text{ Im}(\Pi) \), where \( L = L^\top \) is the Laplacian of an undirected graph, see e.g., (25) [31]. Further, we have \( L \Pi = \Pi \hat{L} \) with \( \hat{L} := (\Pi^\top \Pi)^{-1} \Pi^\top L \Pi \). A generalization of almost equitable partitions to digraphs is considered [64], where nodes in the same cluster should have identical weighted out-degrees. Then, \( \text{ Im}(\Pi) \) is still \( L \)-invariant. Now consider the following error system

\[
\Delta(s) = \eta(s) - \hat{\eta}(s) = \left[ \begin{array}{cc} H & -L \Pi \\ -H \Pi & sI_n + L \\ 0 & sL_r + \Pi^\top L \Pi \end{array} \right]^{-1} \left[ \begin{array}{c} F \\ \Pi^\top F \end{array} \right],
\]

where \( \Pi^\top = (\Pi^\top \Pi)^{-1} \Pi^\top \) and \( \eta(s), \hat{\eta}(s) \) are the transfer matrices of Equation 8. From the \( L \)-invariance it is verified that \( \hat{\eta}(-s)^\top \Delta(s) = 0 \), and thus \( \|\Delta(s)\|_\infty = \|\eta(s)\|_\infty - \|\hat{\eta}(s)\|_\infty \).

Furthermore, for a special output of the system [4] explicit expressions for the reduction error \( \|\Delta(s)\|_\infty \) and \( \|\Delta(s)\|_\infty \) are provided in (25) [30]. Further discussion on model reduction of networked symmetric linear system based on almost equitable partitions can be found in (23). Although an almost equitable partition as a particular clustering offers us analytical expression for the reduction error, it does not necessarily lead to a small error. In fact, the methods in (65, 66) provide significantly lower errors via alternative choices of clustering for some examples. Moreover, how to find all almost equitable partitions for a large-scale graph is generally a rather difficult and computationally expensive problem (24).

3.1.2. Tree Networks. Next, we focus on a particular class of undirected networks with tree topology. In graph theory, a tree is a connected undirected graph in which there is only one path between any two nodes. An example of an undirected tree is shown in Figure 5.

![Figure 5](image_url)

A tree with 6 nodes.

Clearly, a tree \( T \) with \( n \) nodes has exactly \( n - 1 \) edges. Let \( B \in \mathbb{R}^{n \times (n-1)} \) be the incidence matrix of \( T \). Relevant to the expression of the graph Laplacian \( L \) in Equation 2 we define an edge Laplacian as \( L_e = B^\top BW \in \mathbb{R}^{(n-1) \times (n-1)} \), where \( W \) is the diagonal edge weight matrix. Observe that \( L_e \) is full rank and has all eigenvalues real and positive. The eigenvalues of \( L_e \) coincide with the nonzero eigenvalues of \( L \), the Laplacian matrix of \( T \).

Consider the Laplacian dynamics in Equation 4 where \( \Gamma = -L \) is an undirected Laplacian in Equation 2. Applying the transformation \( x_e = B^\top x \) then leads to the so-called edge agreement protocol [11, 67]: \( \dot{x}_e(t) = -L_e x_e(t) + B^\top F u(t) \), which is asymptotically stable and minimal. Network reduction approaches can be developed based on edge operations. For example, (68) provides a greedy algorithm for edge-based contraction to simplify the graph topology. A more general form of the edge agreement protocol is derived when subsystems are taken into account. In (24) the edge system of a network system with \( \Gamma = -L \) representing an undirected tree graph is defined as

\[
\Sigma_e : \left\{ \begin{array}{l}
\dot{x}_e(t) = (I_{n-1} \otimes A - L_e \otimes BC)x_e(t) + (B^\top F \otimes B)u(t), \\
\eta_e(t) = (HBW \otimes C)x_e(t),
\end{array} \right.
\]

12 Cheng • Scherpen
where \( x = (B^\top \otimes I)x \in \mathbb{R}^{(n-1)} \). Assuming that the subsystem \((A, B, C)\) in Equation 11 is passive and minimal, we have the synchronization property of the network system \( \Sigma \) from Theorem 2. It then follows that the edge system \( \Sigma_e \) is asymptotically stable. Furthermore, we can define a pair of generalized controllability and observability Gramians of the edge system \( \Sigma_e \) as follows, (27), \( P_e := X \otimes K^{-1} \), \( Q_e := Y \otimes K \), where \( K > 0 \) satisfies 13 for the passive subsystem, and \( X > 0 \) and \( Y > 0 \) are solutions of the following inequalities:

\[
-L_e X - XL_e^\top + B^\top FF^\top B \leq 0, \quad -L_e^\top Y - YL_e + WB^\top H^\top HBW \leq 0.
\]

The matrices \( X \) and \( Y \) admit a diagonal structure: \( X = \text{Diag}(\xi_1, \xi_2, ..., \xi_{n-1}) \), \( Y = \text{Diag}(\eta_1, \eta_2, ..., \eta_{n-1}) \), where the ordering \( \xi_i \eta_i \geq \xi_{i+1} \eta_{i+1} \) is imposed. The value of \( \xi \eta \) can be roughly viewed as an indication for the importance of the \( i \)-th edge, since similar to balanced realization theory \( \xi \) and \( \eta \) are related to controllability and observability properties of the edges. Reduction by truncation methods then are equivalent to aggregating nodes connected by the truncated edges, and moreover (27) provides an \textit{a priori} upper bound for the approximation error in terms of the \( \mathcal{H}_\infty \) norm \( \| G(s) - \hat{G}(s) \|_{\mathcal{H}_\infty} \leq 2 \left( \sum_{i=0}^{n-1} |L_e^{-1}|_{\mathcal{H}_2} \right)^{\frac{1}{2}} \right) \), where \( r \) is the number of nodes in the reduced network.

For an extension beyond tree graphs a major challenge lies in the characterization of the edge system, (27).

### 3.1.3. Dissimilarity-Based Clustering

For generic network systems, we may resort to a dissimilarity-based clustering approach presented in e.g., (69, 29, 28, 62). In line with data classification or pattern recognition in the other domains, dissimilarity-based clustering for dynamic networks starts with a proper metric that quantifies the difference between any pair of nodes (subsystems) in a network. For static graphs, dissimilarity (or distance) between two nodes or data points can be computed in a vector space using some sort of metric, (21, 22). Considering dynamic networks with external inputs, the dissimilarity metric is then featured in a function space (29, 28, 62).

**Definition 4.** Consider the network system of Equation 12, the dissimilarity between nodes \( i \) and \( j \) is defined as

\[
D_{ij} := \| sI_n - \Gamma \|_{\mathcal{H}_2},
\]

where \( \eta_i(s) := (e_i^\top \otimes C)(sL_{ij} - I_n \otimes A + \Gamma \otimes BC)^{-1}(F \otimes B) \). Consider a single-integrator network in Equation 4, a node dissimilarity can be simply defined as

\[
D_{ij} = \| (e_i - e_j)^\top (sI_n - \Gamma)^{-1} F \|_{\mathcal{H}_2}.
\]

The transfer matrix \( \eta_i(s) \) maps the external control signal \( u \in \mathbb{R}^p \) to the measured output of the \( i \)-th node \( y_i \in \mathbb{R}^m \). Thus \( \eta_i(s) \) can be interpreted as the behavior of the \( i \)-th node with respect to the external inputs, while the dissimilarity measure in Equation 25 indicates how different two nodes behave. The location of inputs and network topology determine the value of dissimilarity. Dissimilarity can also be defined in terms of other function norms, e.g., the \( \mathcal{H}_\infty \) norm. However, to compute the dissimilarity between every pair of nodes in a large-scale network, the \( \mathcal{H}_2 \) norm of a stable LTI system can be characterized by its Gramians (29), whereas for other norms there is no such characterization, making them computationally less feasible.

Note that the dissimilarity in Equation 25 or Equation 26 is only well defined when \( \eta_i(s) - \eta_j(s) \in \mathcal{H}_2 \). This condition is guaranteed for network systems that are asymptotically stable or achieve synchronization (29). For instance, if \( \Gamma \) in Equation 4 is Hurwitz,
then Equation 26 immediately becomes $D_{ij} = \sqrt{(e_i - e_j)^\top P (e_i - e_j)}$, where $P$ is the controllability Gramian of the network system and $P$ is computed as the unique solution of the Lyapunov equation $\Gamma P + P \Gamma^\top + F F^\top = 0$. If system is semistable, we resort to pseudo Gramians defined by Equation 19 for the computation of the $H_2$ norm.

For example, a single-integrator network in Equation 4 is considered with $\Gamma = -L$ the Laplacian matrix of an undirected graph. Following (70, 59), the pseudo controllability Gramian of the system is computed as $P = \tilde{J} P \tilde{J}^\top$, where $\tilde{P}$ is a solution of 

$$ -L \tilde{P} - \tilde{P} L + (I - \tilde{J}) F F^\top (I - \tilde{J}) = 0, \quad \tilde{J} := \frac{1}{n} \mathbb{1} \mathbb{1}^\top. \tag{27} $$

The dissimilarity in Equation 26 is thereby obtained as

$$ D_{ij} = \sqrt{(e_i - e_j)^\top \tilde{P} (e_i - e_j)}. \tag{28} $$

Further consider the network system with $\Gamma = -L$ a symmetric Laplacian matrix. Assume that the network achieves synchronization. In this case, (29) provides the expression for Equation 28 as

$$ D_{ij} = \sqrt{\text{Tr}(\Psi_{ij} \tilde{P} \Psi_{ij}^\top)}, \tag{29} $$

where $\Psi_{ij}$ is defined with help of the output matrix $C$ and $\tilde{P} \in \mathbb{R}^{(n-1)\times(n-1)}$ is the unique solution of a Lyapunov equation with matrices built from the system matrices.

The dissimilarity in Definition 4 is a pairwise measure that shows how close the behavior of two subsystems is, thus taking dynamics into account. This is significantly different from conventional node dissimilarity in data science or computer graphics (21, 22). Nevertheless, we can still follow similar clustering procedures or algorithms for data sets or static graphs.

Formally, given a network, the goal of clustering is to divide the nodes into clusters such that the elements assigned to a particular cluster are similar in a predefined metric. However, clustering with respect to a distance metric is generally an NP-hard combinatorial optimization problem, which is commonly solved by approximation algorithms. Here, we review two of such algorithms and their adaptation to the clustering of network systems.

Agglomerative hierarchical clustering is a method that produces multi-level clusters. The key of this method is to define the proximity between two clusters $C_i$ and $C_j$. There are several alternatives for such definition, such as considering the minimum, maximum or average dissimilarity between any node in $C_i$ and any node in $C_j$, respectively. The proximity of two clusters allows us to identify a pair of clusters with the smallest proximity and merge them into a single cluster. This operation is executed iteratively to generate clusters in a hierarchy structure, which is visualized as a dendrogram, see an example shown in Figure 6.

This is a tree-like diagram that records the sequences of cluster merges. A graph clustering is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster. The implementation of hierarchical clustering to the model reduction of network systems can be found in e.g., (71, 28, 59, 72).

Example 2. Consider the networked mass-damper system in Example 1. The dissimilarity matrix can be computed as in Equation 28 which yields

$$ D = \begin{bmatrix}
0 & 0.2494 & 0.3154 & 0.3919 & 0.4142 \\
0.2494 & 0 & 0.2119 & 0.3688 & 0.3842 \\
0.3154 & 0.2119 & 0 & 0.2410 & 0.2394 \\
0.3919 & 0.3688 & 0.2410 & 0 & 0.0396 \\
0.4142 & 0.3842 & 0.2394 & 0.0396 & 0
\end{bmatrix}. $$

14 Cheng • Scherpen
We use the average-link to define the cluster proximity, and a dendrogram is generated as depicted in Figure 6, showing how clusters are merged hierarchically. The dashed line cuts the dendrogram at a chosen level such that three clusters are formed: \{1\}, \{2,3\}, \{4,5\}.

![Dendrogram](image)

**Figure 6**

Dendrogram illustrating the hierarchical clustering of the networked mass-damper system. The horizontal axis is labeled by node numberings, while the vertical axis represents the proximity of clusters. The level at which branches merge indicates the proximity between two clusters.

K-means clustering is a typical centroid-based partitioning method, by which a cluster is constructed such that all the nodes within the cluster is more similar to the centroid of this cluster than to the centroid of any other clusters. For a network system of Equation 12, the centroid of a cluster \(C_k\) can be defined as

\[
\mu(C_k) = \frac{1}{|C_k|} \sum_{i \in C_k} \eta_i(s),
\]

30.

with \(\eta_i(s)\) defined in Equation 25. Given a network of \(n\) nodes, K-means clustering aims to partition the nodes into \(r\) subsets so as to minimize the following objective function:

\[
\arg \min_{1 \leq k \leq r} \sum_{i \in C_k} \left \| \eta_i(s) - \mu(C_k) \right \|_{\mathcal{H}_2}^2,
\]

in which \(\eta_i(s) - \mu(C_k) \in \mathcal{H}_2\) holds for synchronized networks. This problem can be solved using a simple iterative algorithm. First, take an initial \(r\) clusters of a given network, and specify the centroid as in Equation 30 for each cluster. Then, compute the dissimilarity between every node \(i\) and the \(r\) centroids, and assign node \(i\) to the cluster whose centroid is the closest to \(i\). Finally, we update the cluster centroids accordingly and repeat the steps until convergence.

Note that the formation of clusters in hierarchical clustering or K-means clustering solely relies on the dissimilarity measures and thus does not take into account the connectedness of nodes within a same cluster. It is worth noting that both methods can be modified to produce clusters of a graph, where each cluster forms a connected subgraph. (20, 52, 71).

Generally, an upper bound on the reduction error \(G(s) - \hat{G}(s)\), with \(G(s)\) and \(\hat{G}(s)\) the transfer matrices of systems 12 and 15, is not easy to obtain. We thus impose extra assumptions on the network system 12: \(\Gamma = -L\) represents a connected undirected graph, and \(A\) in Equation 11 satisfies \(A + A^\top < 0\). Then, a posteriori error bound is given as

\[
\|G(s) - \hat{G}(s)\|_{\mathcal{H}_2} < \gamma \sum_{k=1}^r |C_k| \cdot \max_{i,j \in C_k} D_{ij},
\]

where \(\gamma \in \mathbb{R}_{>0}\) only depends on the original system 12 and satisfies an LMI, (37, 29).

The most crucial part in dissimilarity-based clustering is to properly define the dissimilarity of nodes and clusters. For LTI network systems, dissimilarity can be defined using transfer matrices, which is applicable to different types of dynamical networks, see, e.g., (28, 62, 62, 71) for more generalizations to second-order networks, directed networks and controlled power networks. However, how to extend the dissimilarity-based clustering to network systems containing nonlinearities still needs further exploration. One potential solution resorts to the dc gain of monotone systems, that is introduced (72), where the dc-gain can be regarded as indicator of the node importance.

www.annualreviews.org • Model Reduction Methods for Network Systems 15
3.2. Clustering Meets Optimization

In the previous section, we have reviewed how to select clusters and construct a reduced-order network model using the clustering-based projection. In this section, we formulate the model reduction problem from the perspective of optimization, that is to construct a lower-order network model which minimizes a certain reduction error.

3.2.1. Reducibility and an $H_2$ Error Bound. The pioneering work on clustering-based model reduction of dynamic networks in [24, 39] introduces a notion of cluster reducibility, which is relevant to the classic notions exact aggregation and approximate aggregation from the control and model reduction literature [75, 76].

Consider the network system in Equation 4 with $\Gamma$ a Hurwitz, Metzler and symmetric matrix and $F \in \mathbb{R}^n$. A cluster $C_k$ is said to be reducible if there exist a scalar rational function $g(s)$ and a vector $p_k \in \mathbb{R}^{\lvert C_k \rvert}$ such that $I(C_k)g(s) = p_kg(s)$, where $g(s) := (sI - \Gamma)^{-1}F$, and $I(C_k)$ denotes the matrix composed of the column vectors of $I_n$ compatible with the set $C_k$. Reducibility reflects the uncontrollability of node states in a cluster and can be further characterized in an algebraic manner. It is shown in [24] that the pair $(\Gamma, F)$ can be converted into a positive tridiagonal realization by a unitary matrix $T$. Define $\Phi := -TT^{-1}F \in \mathbb{R}^{n \times n}$. Then, a cluster $C_k$ is reducible if and only if there exists a vector $\phi_k \in \mathbb{R}^{\lvert C_k \rvert}$ such that $I(C_k)\phi = p_k\phi_k$, where $p_k := -I(C_k)\Gamma^{-1}F$. With the vector $p_k$, an aggregation matrix can be defined as

$$\bar{\Pi} := \left[I(C_1) \cdots I(C_r)\right] \text{blkdiag} \left( \frac{p_1}{\|p_1\|}, \ldots, \frac{p_r}{\|p_r\|} \right).$$  \hspace{1cm} (31)

which can be viewed as a weighted characteristic matrix $\Pi$ in Definition 2. Note that $\bar{\Pi}^\top \bar{\Pi} = I_r$, and the reduced-order network model then becomes $\dot{\tilde{g}}(s) = \bar{\Pi}(sI - \bar{\Pi}^\top \Gamma \bar{\Pi})^{-1}\bar{\Pi}^\top F$.

If all the clusters are reducible, then the obtained reduced-order network model in Equation 7 has exactly the same input-output behavior as that of the original network, thus $\|g(s) - \tilde{g}(s)\|_{H_2} = 0$. To further reduce the network model, the so-called $\theta$-reducibility is defined for a cluster $C_k$ as $\|I(C_k)\Phi - p_k\phi_k\|_{F} \leq \theta$, for vectors $p_k, \phi_k \in \mathbb{R}^{\lvert C_k \rvert}$. If all the clusters are $\theta$-reducible, a posteriori upper bound on the reduction error can be formed:

$$\|g(s) - \tilde{g}(s)\|_{H_2} \leq \gamma \left( \sum_{k=1}^{r} |C_k| (|C_k| - 1) \theta \right).$$  \hspace{1cm} (32)

where $\gamma$ is an upper bound of $\|\bar{\Pi}(sI_r - \bar{\Pi}^\top \Gamma \bar{\Pi})^{-1}\bar{\Pi}^\top \Gamma + I_n\|_{H_\infty}$ and characterized by a Riccati inequality.

A generalization of the above method is provided in [49] that considers semistable directed networks. In this case, the Frobenius eigenvector of $\Gamma$ is used for constructing the aggregation matrix in Equation 31 to preserve both semistability and positivity in the reduced-order network model. A so-called projected controllability Gramian, which can be viewed as a special pseudo controllability Gramian, is used for the characterization. The posteriori error bound in Equation 32 is also extended to directed networks. However, this extension turns out to be questionable, since the relevant Riccati inequality in general does not have a solution for semistable systems. A correct formulation in terms of the $H_\infty$ norm of a semistable system can be found in [50], which has a form of Equation 31.

The notion of reducibility and the error bound are essential for the clustering procedure in [24, 39]. The core step in the clustering algorithm is to produce a set of $\theta$-reducible clusters.
clusters, where the value of \( \theta \) is adjusted in relation to the approximation error bound. This approach is extended in (77) to reduce stable second-order network systems, and to reduce networked dissipative systems in the form of Equation 12.

### 3.2.2. \( \mathcal{H}_2 \)-Suboptimal Methods

Model reduction of a network system can be formulated as a nonconvex optimization problem of which the objective function is the \( \mathcal{H}_2 \) reduction error. The characteristic matrix \( \Pi \) is the optimization variable and is subject to the constraint:

\[
\Pi \in \mathcal{C} := \{ \Pi \in \mathbb{R}^n, \text{ and } [\Pi]_{ij} = \{0, 1\}, \forall i = 1, \ldots, n, j = 1, \ldots, r \}. \quad 33
\]

The optimization problem itself is nonconvex due to the nonlinear objective function in terms of the \( \mathcal{H}_2 \) norm and the binary variable \( \Pi \). In order to solve such a nonconvex problem, a relaxation of the binary constraints can be taken, leading to suboptimal approaches.

When we would drop the constraint \( \Pi \in \mathcal{C} \) we obtain an \( \mathcal{H}_2 \) optimal model reduction problem for a generic LTI system which can be solved with the so-called Iterative Rational Krylov Algorithm (IRKA) to seek for a (locally) optimal solution (65). The algorithm gives a subspace of dimension \( r \) with the basis \( V_r \in \mathbb{R}^{n \times r} \). Different from \( V \) defined in Equation 17, this \( V_r \) is not a feasible solution since it does not preserve the network structure. Therefore, the idea is to find a \( \Pi \) matrix in the feasible set \( \mathcal{C} \) such that the image of \( \Pi \) is approximately equal to the image of \( V_r \), i.e., \( \text{Im}(\Pi) \approx \text{Im}(V_r) \). To this end, (65) adopts a approach based on a QR decomposition with column pivoting, which is originated from (73) for solving K-means clustering problems.

An alternative method in (78) further studies this nonconvex optimization problem for a network system in Equation 4 with \( G \) being a given graph clustering of \( \mathcal{G} \). Then, a \( \Pi \) matrix in the feasible set \( \mathcal{C} \) with the basis \( V_r \in \mathbb{R}^{n \times r} \) is given by:

\[
\Pi = \mathcal{C} \Pi \mathcal{C}^\top \approx \mathcal{C} \Pi \mathcal{C}^\top, \quad \forall \mathcal{C} \in \mathcal{C}, \mathcal{C}^\top \mathcal{C} = I_n.
\]

### 3.2.3. Edge Weighting

Instead of seeking for a way to do the graph clustering, the \( \mathcal{H}_2 \) optimization scheme can also be applied to construct a “good” reduced-order model from a given clustering. To achieve this, we have to go beyond the framework of Petrov-Galerkin projection. Given a certain clustering, the topology of a reduced-order network is known, while the coupling strengths (edge weights) are considered as free parameters to be determined. (66) further studies this nonconvex optimization problem of which the objective function is the \( \mathcal{H}_2 \) norm and the binary variable \( \Pi \). In order to solve such a nonconvex problem, a relaxation of the binary constraints can be taken, leading to suboptimal approaches.

When we would drop the constraint \( \Pi \in \mathcal{C} \) we obtain an \( \mathcal{H}_2 \) optimal model reduction problem for a generic LTI system which can be solved with the so-called Iterative Rational Krylov Algorithm (IRKA) to seek for a (locally) optimal solution (65). The algorithm gives a subspace of dimension \( r \) with the basis \( V_r \in \mathbb{R}^{n \times r} \). Different from \( V \) defined in Equation 17, this \( V_r \) is not a feasible solution since it does not preserve the network structure. Therefore, the idea is to find a \( \Pi \) matrix in the feasible set \( \mathcal{C} \) such that the image of \( \Pi \) is approximately equal to the image of \( V_r \), i.e., \( \text{Im}(\Pi) \approx \text{Im}(V_r) \). To this end, (65) adopts a approach based on a QR decomposition with column pivoting, which is originated from (73) for solving K-means clustering problems.

An alternative method in (78) further studies this nonconvex optimization problem for a network system in Equation 4 with \( G \) being a given graph clustering of \( \mathcal{G} \). Then, a \( \Pi \) matrix in the feasible set \( \mathcal{C} \) with the basis \( V_r \in \mathbb{R}^{n \times r} \) is given by:

\[
\Pi = \mathcal{C} \Pi \mathcal{C}^\top \approx \mathcal{C} \Pi \mathcal{C}^\top, \quad \forall \mathcal{C} \in \mathcal{C}, \mathcal{C}^\top \mathcal{C} = I_n.
\]
Example 3. Consider an undirected graph composed of 6 nodes in Figure 7a. Given a clustering with $C_1 = \{1, 2\}$, $C_2 = \{3\}$, $C_3 = \{4\}$, $C_4 = \{5, 6\}$, the quotient graph is obtained in Figure 7b with edge weight matrix $W = \text{Diag}(\hat{w}_1, \hat{w}_2, \hat{w}_3, \hat{w}_4)$. Then, the parameterized model of the reduced-order network is obtained as

$$
\begin{align*}
\begin{bmatrix}
\hat{z}_1 \\
\hat{z}_2 \\
\hat{z}_3 \\
\hat{z}_4 \\
\end{bmatrix} &= \begin{bmatrix}
2 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 2 \\
\end{bmatrix} \begin{bmatrix}
\hat{z}_1 \\
\hat{z}_2 \\
\hat{z}_3 \\
\hat{z}_4 \\
\end{bmatrix} - \begin{bmatrix}
\hat{w}_1 + \hat{w}_2 & -\hat{w}_1 & -\hat{w}_2 & 0 \\
-\hat{w}_1 & \hat{w}_1 + \hat{w}_4 & 0 & -\hat{w}_4 \\
-\hat{w}_2 & 0 & \hat{w}_2 + \hat{w}_3 & -\hat{w}_3 \\
0 & -\hat{w}_4 & -\hat{w}_3 & \hat{w}_3 + \hat{w}_4 \\
\end{bmatrix} \begin{bmatrix}
\hat{z}_1 \\
\hat{z}_2 \\
\hat{z}_3 \\
\hat{z}_4 \\
\end{bmatrix} \\
&= \begin{bmatrix}
\hat{w}_1 + \hat{w}_2 & -\hat{w}_1 & -\hat{w}_2 & 0 \\
-\hat{w}_1 & \hat{w}_1 + \hat{w}_4 & 0 & -\hat{w}_4 \\
-\hat{w}_2 & 0 & \hat{w}_2 + \hat{w}_3 & -\hat{w}_3 \\
0 & -\hat{w}_4 & -\hat{w}_3 & \hat{w}_3 + \hat{w}_4 \\
\end{bmatrix} \begin{bmatrix}
\hat{z}_1 \\
\hat{z}_2 \\
\hat{z}_3 \\
\hat{z}_4 \\
\end{bmatrix} \\
&+ \begin{bmatrix}
0 \\
1 \\
0 \\
0 \\
\end{bmatrix} \begin{bmatrix}
u_1 \\
u_2 \\
u_3 \\
u_4 \\
\end{bmatrix} \\
&= \begin{bmatrix}
\hat{w}_1 + \hat{w}_2 & -\hat{w}_1 & -\hat{w}_2 & 0 \\
-\hat{w}_1 & \hat{w}_1 + \hat{w}_4 & 0 & -\hat{w}_4 \\
-\hat{w}_2 & 0 & \hat{w}_2 + \hat{w}_3 & -\hat{w}_3 \\
0 & -\hat{w}_4 & -\hat{w}_3 & \hat{w}_3 + \hat{w}_4 \\
\end{bmatrix} \begin{bmatrix}
\hat{z}_1 \\
\hat{z}_2 \\
\hat{z}_3 \\
\hat{z}_4 \\
\end{bmatrix} \\
&+ \begin{bmatrix}
0 \\
1 \\
0 \\
0 \\
\end{bmatrix} \begin{bmatrix}
u_1 \\
u_2 \\
u_3 \\
u_4 \\
\end{bmatrix}
\end{align*}
$$

and $y = z_3$. In this model, the diagonal elements of $\hat{W}$ are the parameters to be determined.

Similar to the process in Section 3.3.2, an optimization problem can be formulated to minimize the reduction error between the original and reduced-order network systems by tuning the edge weights. Specifically, the objective function is $\|\eta(s) - \hat{\eta}_p(s, \hat{W})\|_{\mathcal{H}_2}$, in which a diagonal and positive definite $\hat{W}$ is the optimization variable.

There are multiple algorithms for solving such a problem. In (66), the upper bound of the reduction error, i.e., the expression $\|\eta(s) - \hat{\eta}_p(s, \hat{W})\|_{\mathcal{H}_2} < \gamma$, is characterized by a set of LMIs, and a cross iteration algorithm is applied such that the upper bound $\gamma$ decreases.
via iterations. An alternative approach is presented in [60] by means of a convex-concave decomposition. This approach is inspired by the work in [80], and is based on linearization of the optimization problem at a given point, so that the problem becomes convex and can be solved efficiently. Then, the overall problem can be solved in an iterative fashion, and in each iteration a convex optimization problem needs to be solved.

It is worth noting that the edge weighting approach can be implemented as a subsequent procedure for the clustering-based methods described earlier, i.e., we can first apply an aforementioned algorithm to find a graph clustering, whose edge weights can be used to initialize $\hat{W}$ in the edge weighting approach. Then, through iterations, a more accurate reduced-order network model can be generated.

### 3.3. Other Topological Reduction Methods

There exists a vast amount of literature about the problem of topological reduction. In this section, we briefly summarize several other representative methods.

#### 3.3.1. Singular Perturbation Approximation and Kron reduction

Along with graph clustering, the other mainstream methodology for simplifying the topological complexity of a network is based on time-scale separation analysis, and in particular, singular perturbation approximation [61]. This method has been extensively investigated in the applications of biochemical reaction systems [4, 82, 83] and electric power networks [84, 85, 86, 87, 88]. A key feature of those systems is that there usually is an explicit or non-explicit separation of time scales in the states of networks. For example, slow coherency theory implies that power networks are naturally decomposed into areas, where power generators within each area synchronize on a faster time scale, while network-wide interactions between the areas are captured by slower motions. Singular perturbation methods then help to separate these dynamics to analyze them separately. This produces a reduced-order network model that retains for example the low frequency behavior of a large-scale network. How to identify and separate fast/slow states is a crucial step in this method, and its application is highly dependent on the specific application.

A relevant concept is Kron reduction of graphs, which is a terminology commonly used in classic circuit theory and in related fields such as electrical impedance tomography, and transient stability assessment in power networks, see e.g., [8, 87, 88]. This may also be used for exact reduction, e.g., to go from a differential algebraic description of the network system to a differential description with structure preservation.

#### 3.3.2. Kullback-Leibler Aggregation

Networked dynamical systems derived from the discretization of thermodynamics and fluid dynamics are usually modeled as regular discrete-time Markov chains without control inputs. For this type of systems, the notion of Kullback-Leibler (K-L) divergence rate can be adopted to measure the difference between two Markov chains, defined on the same state space. This notion is further extended in [89] to measure the K–L divergence rate between the original and reduced-order models defined on different state spaces. With the new K–L divergence rate, an optimization problem is formulated which aims to find an optimal partition of the states, which are aggregated to form a reduced-order model. An application of this method is explored in reducing complex building thermal systems [26].
3.3.3. Network Reduction towards Scale-Free Structure. Graph clustering and aggregation have also been studied to retain more relevant properties such as connectivity and scale-freeness in (31, 90). The scale-free networks typically contain a few nodes with a large degree (the so-called hubs) and a large number of nodes with a small degree, and the distribution of the node degree follows a certain power law (31).

The model reduction problem of networks preserving the scale-free property can be formulated as an optimization problem that finds a clustering of a given large-scale network such that the aggregated network has a degree distribution closest to a desired scale-free distribution. The preservation of the scale-free structure is particularly important for applications of flow networks, such as traffic networks, power networks or packet flow networks.

3.3.4. Indirect Network Reduction Methods. Different from the mechanisms of clustering and aggregation that directly produce a reduced network, indirected methods in (91, 38, 92, 93) seek a structure-preserving reduced-order model using a two-step procedure: reduction and transformation. In the reduction step, a lower-dimensional model of a given large-scale network system is constructed by using conventional model reduction methods, e.g. generalized balanced truncation (91, 38) or moment matching (92). Generally, the reduced-order model generated in this step does not allow for a network interpretation. Then, the transformation step is implemented subsequently, which converts the reduced-order model obtained in the previous step into a network model. This method is also relevant for the combined nodal and topological reduction procedure in Section 4.2.

A key for the transformation is presented in (91, 38): A matrix is similar to a Laplacian matrix of a connected undirected graph if and only if it is diagonalizable and has exactly one zero eigenvalue while all the other eigenvalues are real positive. This result guides the reduction step, in which certain spectral constraint has to be imposed. Then, the second step turns out to be an eigenvalue matching problem.

Similarly, an eigenvalue assignment approach, (93), directly selects a subset of the Laplacian spectrum of the original network to be the eigenvalues of the Laplacian matrix for the reduced network. By doing so, certain properties of original network such as stability and synchronization can be preserved through the reduction process.

4. REDUCTION OF THE FULL SYSTEM DYNAMICS

In the previous section the reduction of the topological structure based on clustering methods is considered. The input-output structure is considered when looking at for example the $H_2$ norm, but for control systems it is undoubtedly very important that the model preserves certain input-output or control structures beyond only considering the $H_2$ norm. For general linear and nonlinear systems various methods are developed, e.g., (9), (94), (95). We refer to the Sidebar on Linear Systems for a very brief introduction. Here we first consider reduction of the nodal dynamics while preserving certain graph properties. Secondly, we treat a combined nodal and topological method.

4.1. Reduction of the Nodal Dynamics

For the nodal dynamics it is useful to consider reduction methods for interconnected systems. In the corresponding literature the network perspective is not the primary focus, but the methods are nevertheless relevant for network systems and treated in this subsection.
In addition, we treat reduction methods for nodal dynamics that do explicitly take the network perspective and preserve properties like synchronization.

4.1.1. Reduction Methods for Interconnected Linear Systems. Perhaps one of the first relevant papers that considers reduction methods for interconnected systems is (34), where linear fractional transformations are considered, see Figure 8. The latter paper considers two methods for reduction, both based on balanced truncation principles. The first method only considers the diagonal blocks of the observability and controllability Gramians, with each block corresponding to a subsystem, hence neglecting the off diagonal blocks. An expression for the posteriori error bound is provided, but unfortunately, an a priori error bound is not obtained. The second method uses generalized Gramians, i.e., instead of considering the observability and controllability Gramians that are the unique solutions to corresponding Lyapunov equations, the (non-unique) solutions to Lyapunov inequalities are considered. The freedom in choosing solutions to these Lyapunov inequalities provides a possibility to pick block diagonal solutions, and consequently results in an a priori error bound. An extension towards a singular perturbation perspective is provided in (96), and balanced truncation based on generalized Gramians in the discrete time setting for interconnected systems is provided in (97). More generally, reduction methods for coupled systems, and an overview of the various methods until 2008 are provided in (33) and (32).

A more recent result deals with interconnected systems in a graph setting, e.g., (98). In particular, the subsystems are of the form $\dot{x}_i = \sum_{j \in N_i} A_{ij}x_j + B_iu$, where $N_i$ is the index set of the connections of the $i^{th}$ subsystem with other subsystems. With help of generalized Gramians, optimal $H_2$ moment matching based model reduction is performed while preserving the network structure. This is done in a convex optimization setting, and additionally a projected gradient method is applied for the non-convex case.

4.1.2. Nodal Reduction while Preserving Synchronization Properties. The above methods consider interconnected systems, but do not take into account yet the preservation of properties that are typically relevant for (controlled) network systems, such as consensus or synchronization properties. For that, additional steps have to be taken. To the best of our knowledge the first work that considers preservation of synchronization properties in network systems is (35). Consider a diffusively-coupled linear network system as in Equa-
tion with a symmetric Laplacian $\Gamma = -L$. Then a bounded real balancing method can be employed in order to preserve the stability of the network. In addition, synchronization can be preserved by considering the positive definite solution $K$ of the following Riccati equation:

$$(A - \lambda BC)^T K + K (A - \lambda BC) + C^T C + \delta^2 K B B^T K = 0$$

where $\delta$ is a scalar that has to fulfill some additional conditions, and $\lambda$ is an eigenvalue of the Laplacian $L$ for which $A - \lambda BC$ is Hurwitz. The maximum and minimum solutions of the equation can now be balanced, i.e., simultaneously diagonalized, similar as is done for standard balancing. Truncating the system based on these diagonal values results in reduced order dynamics of the agents and a network system which is still synchronized. Furthermore, an a priori error bound is provided based on the truncated diagonal values, $\delta$, and the largest eigenvalue of the Laplacian.

The above balancing method based on the minimum and maximum solution of a Riccati equation can be generalized to finding a solution to the inequality, and as such this can be related to solutions of Linear Matrix Inequalities (LMIs). Also, so far, only linear dynamics in the nodes is considered, where in practice nonlinearities play an important role. Because error dynamics are more difficult to handle in the case of nonlinear systems, it useful to consider nodal dynamics represented by Lur’e systems, i.e., systems with a static nonlinearity in the feedback loop. In (99, 36) robust synchronization preserving reduction methods for nodal Lur’e systems are considered. For this, consider the following nodal dynamics:

$$\Sigma_i : \begin{cases} 
\dot{x}_i(t) = Ax_i(t) + Bu_i(t) + Ez_i(t) \\
y_i(t) = Cx_i(t) \\
z_i(t) = -\phi(y_i(t)) 
\end{cases}$$

where $x_i \in \mathbb{R}^n$, $u_i, y_i, z_i \in \mathbb{R}^m$, and $\phi : \mathbb{R}^m \to \mathbb{R}^m$ fulfilling some sector bound condition. Without loss of generality, we take $m = 1$, (99). For a diffusively coupled network which is robustly synchronized and which has nodal dynamics as in Equation (36) we now take the minimum and maximum solutions $K_{\text{min}}$ and $K_{\text{max}}$ of the following LMI

$$\begin{bmatrix} 
A^T K + KA + C^T C \lambda \ K B K E - C^T \\
\lambda \ K B B^T K \\
\lambda \ B^T K - C \\
E^T K - C 
\end{bmatrix} \preceq 0,$$

with $\lambda$ the largest eigenvalue of the Laplacian. $K_{\text{min}}$ and $K_{\text{max}}$ can be balanced and reduction based on them results in a robustly synchronized network of Lur’e systems with a priori determined error bounds. Variations on this can be done to obtain even better error bounds, and an extension to the Multi-Input Multi-Output case is provided in (36).

4.2. Combined Topological and Nodal Reduction

From the previous sections, it has been observed that the techniques for topological simplification and subsystem reduction in network systems are derived from rather different perspective. The methods for reducing subsystem dynamics are commonly adapted from classic model reduction techniques e.g., balanced truncation, or Krylov subsystem methods, while in structure-preserving simplification of network structures, clustering-based approaches have demonstrated a superior performance. In this section, we discuss the combination of the two approximation problems in a unified framework. This is particularly needed when dealing with a network with both complex topology and high-order subsystems.
Model Reduction for Linear Systems

It is generally accepted that model reduction approaches for linear control systems can be roughly divided into two types of approaches, i.e., singular value based and moment matching based methods, e.g., (9). A very brief review follows. Let \((A,B,C)\) be a state space realization of \(\Sigma\) with dimension \(n\), with input \(u \in \mathbb{R}^m\), state \(x \in \mathbb{R}^n\) and output \(y \in \mathbb{R}^p\). We assume that the system is asymptotically stable and minimal, i.e., controllable and observable. The corresponding transfer matrix is given by \(G(s) = C(sI - A)^{-1}B\).

Balanced Realizations

Theorem 4. \((12)\) Consider the system \(\Sigma\). Take \(P\) the controllability Gramian and \(Q\) the observability Gramian. The eigenvalues of \(QP\) are similarity invariants, i.e., they do not depend on the choice of the state space coordinates. Furthermore, there exists a state space representation where \(\Sigma : = Q = P = \text{Diag}\{\sigma_1, \ldots, \sigma_n\}\), with \(\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0\) the square roots of the eigenvalues of \(QP\). Such representations are called balanced, and the system is in balanced form. Furthermore, the \(\sigma_i\)’s, \(i=1,\ldots,n\), equal the Hankel singular values, i.e., the singular values of the Hankel operator.

The Hankel singular values form a measure for the contribution to minimality of a state component. This provides the basis for model reduction methods based on balanced realizations. Model reduction based on balancing is possible with a-priori error bounds in various norms, e.g., the \(H_\infty\), and Hankel norm. In particular, it can be shown that balanced truncation results in an error bound corresponding to the sum of the truncated Hankel singular values, i.e., \(\|G(s) - G_r(s)\|_\infty \leq 2 \sum_{k=r}^{n} \sigma_k\), where \(G_r(s)\) represents the transfer matrix of the reduced order system. \((13)\). We refer to \((9)\) and \((95)\) for a more elaborate overview.

Moment Matching

The principle of moment matching for a linear system is based on the series representation of the transfer matrix of the system. For more detailed expositions, we refer to e.g., \((9)\) and \((100)\). Without loss of generality it is assumed that \(m = p = 1\).

Definition 5. The 0-moment at \(s_1 \in \mathbb{C}\) of \(\Sigma\) is the complex number \(\eta_0(s_1) = C(s_1I - A)^{-1}B\). The \(k\)-moment at \(s_1\) is given by the complex number

\[
\eta_k(s_1) = \frac{(-1)^k}{k!} \left. \frac{d^k \{C(sI - A)^{-1}B\}}{ds^k} \right|_{s = s_1}, \quad k = 1, 2, 3, \ldots
\]

The point \(s_1 \in \mathbb{C}\) is called an interpolation point. The approximation problem for system \((A,B,C)\) at \(s_1 \in \mathbb{C}\) is to find a system \((\hat{A},\hat{B},\hat{C})\) of order \(\nu < n\), with transfer function \(\hat{G}(s) = \hat{C}(sI - \hat{A})^{-1}\hat{B}\), such that \(\eta_k(s_1) = \hat{\eta}_k(s_1)\), with \(\hat{\eta}_k(s_1)\) the moments of \(\hat{G}(s)\), \(k = 1,\ldots,\nu\). Various types of moment matching methods are developed. Generally it is not possible to provide an a-priori error bound. These methods are however computationally very interesting if one handles systems with millions of states.

However, how to perform a simultaneous reduction of topological complexity and subsystem dynamics is not straightforward. Naively, we may apply the methods of Section 2 and Section 4.1 separately to achieve the two approximation goals one by one. Nevertheless, which reduction sequence gives a better approximation is still unclear, and moreover
there is hardly a guarantee on the approximation error. Relevant results in the literature on combined topological and nodal reduction are rare. The existing ones are developed generally for networked homogeneous linear systems as in Equation 12 under specific assumptions [37, 38]. For example, regularity and dissipativity of the entire system matrix $I \otimes A + \Gamma \otimes BC$ is imposed in [37], which admits a block-diagonal Lyapunov function. It is essential for preserving the stability of the reduced-order model and to derive an upper bound on the approximation error caused by the reduction of both network structure and subsystems, where the topological reduction is done by graph clustering.

In [38], a network system is considered which is synchronized but not necessarily stable. The synchronization property is based on the assumption of each subsystem $(A, B, C)$ being minimal and passive. Extending the results for networked single integrators of Section 3.3.4, the topological complexity is reduced in an indirect manner. A generalized balanced truncation method then results in a unified framework to simplify the network structure and subsystem dynamics simultaneously. The reduction scheme is illustrated in Figure 9, which contains two core steps.

The first step is to decompose the considered network system into two parts which correspond to the average of all subsystems and the discrepancy among the subsystem states, respectively. Due to the synchronization property, the latter part that captures the main dynamics of the entire network is represented by an asymptotically stable system $\Sigma_s$ with a Hurwitz system matrix $(I_{n-1} \otimes A - \bar{\Lambda} \otimes BC)$, where $\bar{\Lambda} \in \mathbb{R}^{(n-1) \times (n-1)}$ shares all the nonzero eigenvalues with the Laplacian matrix $L \in \mathbb{R}^{n \times n}$ of the original network. Then, a pair of generalized Gramians with a Kronecker product structure are selected, which is key to decouple the balanced truncation procedures of the network structure and subsystem dynamics. In the second step, the simplified stable system and average system are integrated, resulting in a reduced-order model of the original $\Sigma$. However, such reduced-order model only provides an approximation of the input-output mapping of $\Sigma$, rather than...
the network structure. To restore the network interpretation in that reduced-order model, a coordinate transformation is required to recover a reduced Laplacian matrix. A theoretical foundation for such transformation is firstly presented in (91):

**Theorem 5.** A real square matrix \( \mathbf{N} \) is similar to the Laplacian matrix \( \mathbf{L} \) associated with a weighted undirected connected graph, if and only if \( \mathbf{N} \) is diagonalizable and has an eigenvalue at 0 with multiplicity 1 while all the other eigenvalues are real and positive.

In (38), a detailed proof of the above theorem is provided, and meanwhile a method is given for reconstructing an undirected network from given eigenvalues \( 0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n \). Recently, an alternative graph reconstruction approach is presented in (101). The reconstructed graphs are usually complete, i.e., there is an edge between any pair of nodes. A subsequent procedure can be taken to sparsify the interconnection structure, see e.g., (101, 102).

### SUMMARY POINTS

1. Graph-theoretical analysis plays a paramount role in reduced-order modeling of complex network systems.
2. Clustering methods provide a structure preserving manner to reduce the topology of network systems. Dissimilarity-based clustering provides a rather general framework for simplifying the topological complexity of a network system, where the key is to properly define a metric to characterize the dissimilarity between nodes/clusters.
3. Network systems that achieve synchronization are naturally semistable. Novel pseudo Gramian notions are introduced for semistable systems as an extension of Gramian matrices for asymptotically stable systems, providing a useful tool to characterize dissimilarity and compute the reduction error.
4. Generalized balanced truncation allows more freedom in constructing a reduced-order model with some desired structures/properties. Thus, it is widely used in the approximation of network systems, particularly in dealing with subsystem reduction.
5. Graph reconstruction can realize a network representation from a reduced-order model satisfying certain spectral constraints. This makes it possible to apply more classical model order reduction methods from the control systems literature for structure preserving reduction of network systems.

### FUTURE ISSUES

1. The approximation of complex network systems with nonlinear couplings and nonlinear subsystems is still a challenge, requiring further investigation.
2. How to reduce the topological complexity of dynamic networks composed of heterogeneous subsystems is not clear yet.
3. The application of reduced-order network models for designing controllers and observers for large-scale networks is appealing, while obtaining provable guarantees on the functionality of the controllers/observers based on reduced-order models should be further explored.
ACKNOWLEDGMENTS

Posted with permission from the Annual Review of Control, Robotics, and Autonomous Systems, Volume 4 by Annual Reviews, [http://www.annualreviews.org](http://www.annualreviews.org)

LITERATURE CITED

1. Mesbahi M, Egerstedt M. 2010. *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press
2. Newman ME. 2003. The structure and function of complex networks. *SIAM Review* 45:167–256
3. Ren W, Beard RW, Atkins EM. 2005. A survey of consensus problems in multi-agent coordination. In *Proceedings of the 2005 American Control Conference (ACC)*, pp. 1859–1864. IEEE
4. Rao S, van der Schaft AJ, Jayawardhana B. 2013. A graph-theoretical approach for the analysis and model reduction of complex-balanced chemical reaction networks. *Journal of Mathematical Chemistry* 51:2401–2422
5. Absendorf T, Wong F, Els R, Gunawardena J. 2014. A framework for modelling gene regulation which accommodates non-equilibrium mechanisms. *BMC Biology* 12:102
6. Proskurnikov AV, Tempo R. 2017. A tutorial on modeling and analysis of dynamic social networks, part i. *Annual Reviews in Control* 43:65–79
7. Dorfler F, Bullo F. 2012. Synchronization and transient stability in power networks and nonuniform kuramoto oscillators. *SIAM Journal on Control and Optimization* 50:1616–1642
8. Chow JH. 2013. *Power System Coherency and Model Reduction*. Springer
9. Antsaklis AC. 2005. *Approximation of Large-Scale Dynamical Systems*. Philadelphia, USA: SIAM
10. Bai Z. 2002. Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. *Applied Numerical Mathematics* 43:9–44
11. Astolfi A. 2010. Model reduction by moment matching for linear and nonlinear systems. *IEEE Transactions on Automatic Control* 55:2321–2336
12. Moore BC. 1981. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control* 26:17–32
13. Glover K. 1984. All optimal Hankel-norm approximations of linear multi-variable systems and their $L_{\infty}$-error bounds. *International Journal of Control* 39:1115–1193
14. Besselink B, Tabak U, Kutowska A, Van de Wouw N, Nijmeijer H, et al. 2013. A comparison of model reduction techniques from structural dynamics, numerical mathematics and systems and control. *Journal of Sound and Vibration* 332:4403–4422
15. Summers TH, Lygeros J. 2014. Optimal sensor and actuator placement in complex dynamical networks. *IFAC Proceedings Volumes* 47:3784–3789
16. Gates AJ, Rocha LM. 2016. Control of complex networks requires both structure and dynamics. *Scientific Reports* 6:1–11
17. Kim JZ, Soffer JM, Kahn AE, Vettel JM, Pasqualetti F, Bassett DS. 2018. Role of graph architecture in controlling dynamical networks with applications to neural systems. *Nature Physics* 14:91–98
18. Ishizaki T, Chakrabortty A, Imura JI. 2018. Graph-theoretic analysis of power systems. *Proceedings of the IEEE* 106:931–952
19. Obinata G, Anderson BDO. 2012. *Model reduction for Control System Design*. Springer Science & Business Media
20. Mustafa D, Glover K. 1991. Controller reduction by $H_{\infty}$ balanced truncation. *IEEE Transactions on Automatic Control* 36:668–682
21. Jain AK, Murty MN, Flynn PJ. 1999. Data clustering: a review. *ACM Computing Surveys (CSUR)* 31:264–323
22. Schaeffer SE. 2007. Graph clustering. *Computer Science Review* 1:27–64
23. van der Schaft AJ. 2014. On model reduction of physical network systems. In *Proceedings of 21st International Symposium on Mathematical Theory of Networks and Systems (MTNS)*, pp. 1419–1425. Groningen, The Netherlands
24. Ishizaki T, Kashima K, Imura JI, Aihara K. 2014. Model reduction and clustering of large-scale bidirectional networks. *IEEE Transactions on Automatic Control* 59:48–63
25. Monshizadeh N, Trentelman HL, Camlibel MK. 2014. Projection-based model reduction of multi-agent systems using graph partitions. *IEEE Transactions on Control of Network Systems* 1:145–154
26. Deng K, Goyal S, Barooah P, Mehta PG. 2014. Structure-preserving model reduction of nonlinear building thermal models. *Automatica* 50:1188–1195
27. Besselink B, Sandberg H, Johansson KH. 2016. Clustering-based model reduction of networked passive systems. *IEEE Transactions on Automatic Control* 61:2958–2973
28. Cheng X, Kawano Y, Scherpen JMA. 2017. Reduction of second-order network systems with structure preservation. *IEEE Transactions on Automatic Control* 62:5026–5038
29. Cheng X, Kawano Y, Scherpen JMA. 2019. Model reduction of multi-agent systems using dissimilarity-based clustering. *IEEE Transactions on Automatic Control* 64:1663–1670
30. Jongsma HJ, Milanić P, Grundel S, Benner P, Trentelman HL. 2018. Model reduction of linear multi-agent systems by clustering with $h_2$ and $H_\infty$ error bounds. *Mathematics of Control, Signals, and Systems* 30:6
31. Martin N, Frasca P, Canudas-de Wit C. 2018. Large-scale network reduction towards scale-free structure. *IEEE Transactions on Network Science and Engineering* 6:711–723
32. Reis T, Stykel T. 2008. A survey on model reduction of coupled systems. In *Model order reduction: theory, research aspects and applications*. Springer
33. Vandendorpe A, Van Dooren P. 2008. Model reduction of interconnected systems. In *Model order reduction: theory, research aspects and applications*. Springer
34. Sandberg H, Murray RM. 2009. Model reduction of interconnected linear systems. *Optimal Control Applications and Methods* 30:225–245
35. Monshizadeh N, Trentelman HL, Camlibel MK. 2013. Stability and synchronization preserving model reduction of multi-agent systems. *Systems & Control Letters* 62:1–10
36. Cheng X, Scherpen JMA, Zhang F. 2019. Model reduction of synchronized homogeneous Lur’e networks with incrementally sector-bounded nonlinearities. *European Journal of Control* 50:11–19
37. Ishizaki T, Ku R, Imura Ji. 2016. Clustered model reduction of networked dissipative systems. In *Proceedings of the 2016 American Control Conference*, pp. 3662–3667. IEEE
38. Cheng X, Scherpen JMA, Besselink B. 2019. Balanced truncation of networked linear passive systems. *Automatica* 104:17–25
39. Bullo F. 2019. *Lectures on Network Systems*. Kindle Direct Publishing
40. Wu CW. 2007. *Synchronization in Complex Networks of Nonlinear Dynamical Systems*. World Scientific
41. Fax JA, Murray RM. 2001. Graph Laplacians and stabilization of vehicle formations
42. Mirzaev I, Gunawardena J. 2013. Laplacian dynamics on general graphs. *Bulletin of Mathematical Biology* 75:2118–2149
43. Dörfler F, Simpson-Porco JW, Bullo F. 2018. Electrical networks and algebraic graph theory: Models, properties, and applications. *Proceedings of the IEEE* 106:977–1005
44. van der Schaft AJ. 2017. Modeling of physical network systems. *Systems & Control Letters* 101:21–27
45. Newman ME, Barabási ALE, Watts DJ. 2006. *The Structure and Dynamics of Networks*. Princeton University Press
46. Fagnani F, Frasca P. 2017. *Introduction to Averaging Dynamics over Networks*, vol. 472. Springer
47. Rantzer A. 2015. Scalable control of positive systems. *European Journal of Control* 24:72–80
48. Rantzer A, Valcher ME. 2018. A tutorial on positive systems and large scale control. In 2018 *IEEE Conference on Decision and Control (CDC)*, pp. 3686–3697. IEEE
49. Ishizaki T, Kashima K, Girard A, Imura Ji, Chen L, Aihara K. 2015. Clustered model reduction of positive directed networks. *Automatica* 59:238–247
50. Johnson CR. 1990. *Matrix Theory and Applications*, vol. 40. American Mathematical Society
51. Scardovi L, Sepulchre R. 2008. Synchronization in networks of identical linear systems. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 546–551. IEEE
52. Ishizaki T, Kashima K, Girard A, Imura Ji, Chen L, Aihara K. 2015. Clustered model reduction of positive directed networks. *Automatica* 59:238–247
53. Bhat SP, Bernstein DS. 1999. Lyapunov analysis of semistability. In *Proceedings of the 1999 American Control Conference*, vol. 3, pp. 1608–1612. IEEE
54. Arcak M. 2007. Passivity as a design tool for group coordination. *IEEE Transactions on Automatic Control* 52:1380–1390
55. Chopra N. 2012. Output synchronization on strongly connected graphs. *IEEE Transactions on Automatic Control* 57:2896–2901
56. Cheng X, Scherpen JMA. 2020. Novel gramians for linear semistable systems. *Automatica* 115:108911
57. Zhang S, Cao M, Cai B. 2013. Upper and lower bounds for controllable subspaces of networks of diffusively coupled agents. *IEEE Transactions on Automatic Control* 59:745–750
58. Aguilar CO, Gharesifard B. 2017. Almost equitable partitions and new necessary conditions for network controllability. *Automatica* 80:25–31
59. Milanić P, Grundel S, Benner P. 2015. Efficient model order reduction for multi-agent systems using QR decomposition-based clustering. In *Proceedings of the 54th IEEE Conference on Decision and Control*, pp. 4794–4799
60. Cheng X, Yu L, Scherpen JMA. 2019. Reduced order modeling of linear consensus networks using weight assignments. In *Proceedings of the 17th European Control Conference*, pp. 2005–2010. Napoli, Italy
61. Zelazo D, Schuler S, Allgöwer F. 2013. Performance and design of cycles in consensus networks. *Systems & Control Letters* 62:85–96
62. Leiter N, Zelazo D. 2017. Graph-based model reduction of the controlled consensus protocol. *IFAC-PapersOnLine* 50:9456–9461
63. Cheng X, Scherpen JMA. 2016. Graph structure-preserving model reduction of linear network systems. In *Proceedings of the 15th European Control Conference*, pp. 1970–1975. Aalborg, Denmark
64. Cheng X, Scherpen JMA. 2016. Introducing network Gramians to undirected network systems.
for structure-preserving model reduction. In Proceedings of the 55th IEEE Conference on Decision and Control, pp. 5756–5761. Las Vegas, the USA

71. Cheng X, Scherpen JMA. 2018. Clustering approach to model order reduction of power networks with distributed controllers. Advances in Computational Mathematics 44:1917–1939

72. Kawano Y, Besselink B, Scherpen JM, Cao M. 2020. Data-driven model reduction of monotone systems by nonlinear dc gains. IEEE Transactions on Automatic Control 65:2094 – 2106

73. Zha H, He X, Ding C, Gu M, Simon HD. 2002. Spectral relaxation for k-means clustering. In Advances in neural information processing systems, pp. 1057–1064

74. Niazi MUB, Cheng X, Canudas de Wit C, Scherpen JMA. 2019. Structure-based clustering for model reduction of large-scale networks. In Proceedings of the 58th IEEE Conference on Decision and Control, pp. 5038–5043. Nice, France

75. Aoki M. 1968. Control of large-scale dynamic systems by aggregation. IEEE Transactions on Automatic Control 13:246–253

76. Feliachi A, Bhurtun C. 1987. Model reduction of large-scale interconnected systems. International Journal of Systems Science 18:2249–2259

77. Ishizaki T, Imura JI. 2015. Clustered model reduction of interconnected second-order systems. Nonlinear Theory and Its Applications, IEICE 6:26–37

78. Cheng X, Necsoara I, Lupu D. 2020. A suboptimal $H_2$ clustering-based model reduction approach for linear network systems. In 17th European Control Conference (ECC2020), pp. 1961–1966

79. Lacoste-Julien S. 2016. Convergence rate of frank-wolfe for non-convex objectives. arXiv preprint arXiv:1607.00345

80. Dinh QT, Gumussoy S, Michiels W, Diehl M. 2011. Combining convex-concave decompositions and linearization approaches for solving BMIIs, with application to static output feedback. IEEE Transactions on Automatic Control 57:1377–1390

81. Kokotovic PV, O’Malley Jr RE, Sannuti P. 1976. Singular perturbations and order reduction in control theory—an overview. Automatica 12:123–132

82. Anderson J, Chang YC, Papachristodoulou A. 2011. Model decomposition and reduction tools for large-scale networks in systems biology. Automatica 47:1165–1174

83. Hancock EJ, Stan GB, Arpino JA, Papachristodoulou A. 2015. Simplified mechanistic models of gene regulation for analysis and design. Journal of The Royal Society Interface 12:20150312

84. Chow JH, Galzarra R, Accari P, Price WW. 1995. Inertial and slow coherency aggregation algorithms for power system dynamic model reduction. IEEE Transactions on Power Systems 10:680–685

85. Buyk E, Arcak M. 2008. Area aggregation and time-scale modeling for sparse nonlinear networks. Systems & Control Letters 57:142–149

86. Romeres D, Dörfler F, Bullo F. 2013. Novel results on slow coherency in consensus and power networks. In Proceedings of the 2013 European Control Conference, pp. 742–747. IEEE

87. Dörfler F, Bullo F. 2013. Kron reduction of graphs with applications to electrical networks. IEEE Transactions on Circuits and Systems I: Regular Papers 60:150–163

88. Monshizadeh N, De Persis C, van der Schaft AJ, Scherpen JMA. 2017. A novel reduced model for electrical networks with constant power loads. IEEE Transactions on Automatic Control

89. Deng K, Mehta PG, Meyn SP. 2011. Optimal kullback-leibler aggregation via spectral theory of markov chains. IEEE Transactions on Automatic Control 56:2793–2808

90. Martin N, Frasca P, Canudas-De-Wit C. 2018. Mergetocure: a new strategy to allocate cure in an epidemic over a grid-like network using a scale-free abstraction. IFAC-PapersOnLine 51:34–39

91. Cheng X, Scherpen JMA. 2017. Balanced truncation approach to linear network system model order reduction. IFAC-PapersOnLine 50:2451–2456

92. Yu L, Cheng X, Scherpen JMA. 2019. $H_2$ sub-optimal model reduction for second-order network systems. In Proceedings of the 58th IEEE Conference on Decision and Control, pp. 5062–5067. Nice, France
93. Yu L, Cheng X, Scherpen JMA. 2019. Synchronization preserving model reduction of multi-agent network systems by eigenvalue assignments. In Proceedings of the 58th IEEE Conference on Decision and Control, pp. 7794–7799. Nice, France

94. Fujimoto K, Scherpen JMA. 2010. Balanced realization and model order reduction for nonlinear systems based on singular value analysis. SIAM Journal on Control and Optimization 48:4591–4623

95. Scherpen JMA. 2011. Balanced realization, balanced truncation and the Hankel operator, chap. 4. The Control Handbook, Control Systems Advanced Methods, Eds. W. Levine, CRC Press, Taylor & Francis Group, 1–24

96. Ishizaki T, Sandberg H, Johansson KH, Kashima K, Imura Ji, Aihara K. 2013. Structured model reduction of interconnected linear systems based on singular perturbation. In 2013 American Control Conference, pp. 5524–5529. IEEE

97. Jaoude DA, Farhood M. 2017. Balanced truncation model reduction of nonstationary systems interconnected over arbitrary graphs. Automatica 85:405–411

98. Necoara I, Ionescu TC. 2020. H2 model reduction of linear network systems by moment matching and optimization. IEEE Transactions on Automatic Control

99. Cheng X, Scherpen JMA. 2018. Robust Synchronization Preserving Model Reduction of Lur’e Networks. In Proceedings of the 16th European Control Conference, pp. 2254–2259. Limassol, Cyprus

100. Scarciotti G, Astolfi A. 2005. Nonlinear Model Reduction by Moment Matching. NOW Publishers, Foundation and Trends in Systems and Control, Vol. 4, No. 3-4, 224-409

101. Forrow A, Woodhouse FG, Dunkel J. 2018. Functional control of network dynamics using designed Laplacian spectra. Physical Review X 8:041043

102. Jongsma HJ, Trentelman HL, Camlibel KM. 2017. Model reduction of networked multiagent systems by cycle removal. IEEE Transactions on Automatic Control 63:657–671