

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

5,600

Open access books available

137,000

International authors and editors

170M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



A Dynamic Graph-Based Systems Framework for Modeling, and Control of Cyber-Physical Systems Typified by Buildings

Fadel M. Lashhab

Abstract

In this chapter, we present a framework for modeling certain classes of cyber-physical systems using graph-theoretic thinking. The cyber-physical systems we consider are typified by buildings. We show that the thermal processes associated with a building can be represented as a graph in which (1) the node variables (temperature and heat flows) are governed by a dynamic system and (2) interconnections between these nodes (walls, doors, windows) are also described by a dynamic system. In general, we call a collection of such nodes and interconnections a dynamic graph (dynamic consensus network). Driven to explore this by developing thermal examples, this study outlines a practical framework for dynamic consensus networks and dynamic graphs. In a manner that seamlessly extends these concepts from the static cases, we will explore the combination of dynamic degrees, adjacency, Laplacian matrices, and incident matrices. With these conceptual tools, one can quickly identify equivalent concepts of dynamic consensus networks.

Keywords: Dynamic graphs, dynamic consensus networks, dynamic Laplacian matrix, dynamic graph theory

1. Introduction

In this chapter, we present the analysis and design of cyber-physical systems using graph-theoretic ideas. We are motivated by the energy-efficient control of buildings.

Our fundamental view of a building is of overlapping, interacting networks, as shown in **Figure 1**. This diagram depicts the dominant phenomenon that contributes to building as a network (or graphs). These networks consist of nodes that constitute distinct sub-systems. For example, in heat-sensing or networks of humans, the nodes may refer to specific rooms in a house or office building. In contrast, nodes might represent a particular sensor, actuator, or perhaps a computational unit in a control network. The links between nodes denote communication of fluctuating or sharing variables in a system, such as the passage of people in a human network between rooms through hallways, or the heat flow between rooms in the thermal network through walls and doors. Smaller circles in **Figure 1** indicate links between networks. Note that typical graph-based networks assume links that

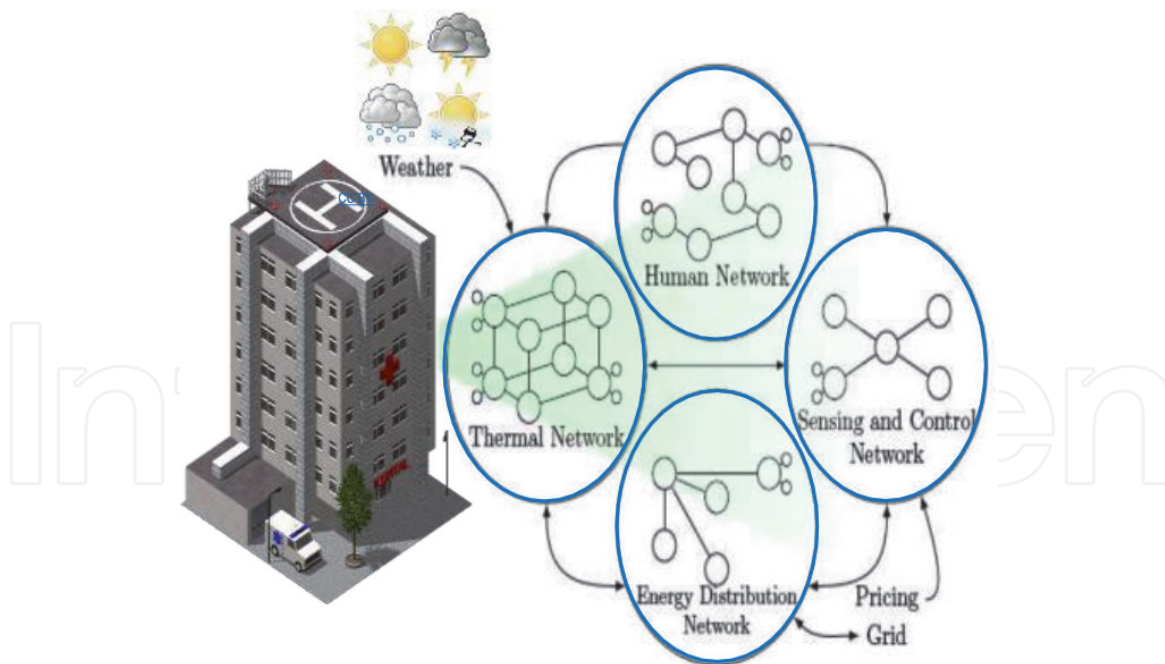


Figure 1.
A building as a collection of interacting networks.

are in some way constant, but as we will see, in some networks, such as the building thermal network, links between nodes may be dynamic.

Control of distributed systems, such as shown in **Figure 1** is a currently active area of research within the field of control systems. By a distributed system, we mean one with many inputs and outputs, possibly spatially distributed dynamics, and a decentralized decision and control architecture, with restrictions on communication between computational nodes. Many researchers have focused basically on homogeneous systems. However, a building may be viewed as a hybrid system where a physical process (the structure itself) has been augmented with a hardware infrastructure (sensors and actuators) and a cyber-infrastructure (communication and decision nodes). Such overlaid heterogeneous systems with constrained connectivity and interaction between the different layers present challenges and system optimization and control opportunities. What is needed are ways to reason about discrete, multi-attribute heterogeneous entities (such as cyber-systems) and continuous, heterogeneous processes (such as physical phenomena) operating on a hierarchy of layered graphs related to each other through a set of mappings or transformations.

In this chapter, we consider methods for studying distributed systems that are heterogeneous and possibly spatially varying. Though a building can be seen as a set of interconnected networks, we consider only the thermal network. We begin by showing how a building's thermal processes can be modeled as a graph whose node variables are temperature and heat flows and whose interconnections are walls, doors, windows, etc. In our graphical representation of a building, both the nodes and the interconnections can be (heterogeneous) dynamic systems. We call this a *dynamic graph* (or network). For such systems, we show that the relationships between the node variables reduce to the traditional graph Laplacian in the steady-state, so that consensus variable convergence can be obtained by discussing the steady-state properties of the system. We then show how a behavioral systems approach can develop kernel relationships between all system variables in dynamic graphs typified by building thermal models. Using these kernel relationships, we consider the controllability analysis of such systems.

The consensus protocols in networking in engineering have received significant academic and corporate attention because of their vast array of potential applications in various fields. Robotics, transportation, sensor networking, communication networking, biology, and physics are only some potential fields that networking consensus could benefit. This section aims to analyze a generalization of consensus problems whereby the weights of network edges are no longer static gains. Instead, they are dynamic systems, which lead towards *dynamic consensus networks*.

Network topology is static for the consensus networks, meaning that there are no dynamics in the interconnections between the nodes ($\lambda_{ij} = \text{constant} \geq 0$), and the nodes are assumed to be integrators [1]. Thus, such problems can be written in the time domain for each node $i = 1, 2, \dots, nn$ as

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} \lambda_{ij} (x_j(t) - x_i(t)). \quad (1)$$

The consensus protocol (Eq. (1)) can be written in matrix form as:

$$\dot{x}(t) = -Lx(t), \quad (2)$$

where $x(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T$ and L , and the Laplacian matrix of the graph $L = [l_{ij}]$, is defined by

$$l_{ij} = \begin{cases} \sum_{j \in \mathcal{N}_i} \lambda_{ij} & i = j \\ -\lambda_{ij} & i \neq j \text{ and } (i, j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

For the multi-agent consensus problem, suppose that N agents develop their personal beliefs $x_i \in \mathbb{R}^1$ regarding a so-called global consensus variable x by communicating with their next-door neighbors by the referred consensus protocol (Eq. (1)). A key outcome is that the solution of the problem $\dot{x}(t) = -Lx(t)$ gives $x_i \rightarrow x^*$ if only all edges are connected to the static graph [1]. This specific fact has been the basis of much of the literature related to consensus problems.

There have been many engineering scientists in the past years involved in the controllability of dynamic consensus networks. The focus was on controlling dynamic consensus networks under the leader-follower approach, where some nodes are considered leaders and other nodes are followers. This approach aims to transfer followers' trajectories from an initial position to the desired position (set-point) by adequately selecting the leaders' trajectory. Many authors [2–5] have considered this framework by using some algebraic methods and the eigenvalues and eigenvectors of the dynamic Laplacian matrix. Other researchers also investigated the controllability using graphic tools such as the graph's equitable partition [4] and symmetry properties [3]. These graphical tools are built based on the graph's configuration and topology associated with the consensus network. The controllability investigation using the minimum energy for static consensus networks using the first-order system formulated and proposed in [6, 7]. This Chapter will investigate the controllability for dynamic consensus networks with edges (links) of rational dynamical systems.

Several researchers have already studied controllability analysis for consensus networks with static topology. The vast majority of these published academic studies have investigated the impact of the static topology on the controllability of their respective consensus networks. Those studies have [8] introduced a graph-theoretic characterization of static networks' structural controllability with only a single

leader. In their research, many illustrate that a static network with variable topology is structurally controllable if the union graph of the underlying static topologies is connected. For example, in [9], the controllability of the network was investigated using the size of the graph as well as its connectivity. Meanwhile, controllability for leader-based, multi-agent systems has been analyzed by [7, 10] based on connectivity and the null space of the leader and followers' incidence matrices. Controllability using graph symmetry and equitable partition properties has been addressed in [11, 12]. The paper [13] formulated an equivalent data-driven Hautus-type test for a general input/output system that assumes no knowledge of the system's state. The authors' work proposed in this Chapter also provided an algorithm for data-driven verification of controllability of the system. They used the singular value decomposition of the Hankel matrix. A multi-vehicle system's consensus problem was proposed and analyzed by [14, 15] with a time-varying reference state. Under the condition, only a portion of the vehicles can access the reference state in this problem. Those vehicles might not have the ability to share the information with the other cars in the team. Although their paper focused on developing an algorithm for investigating the consensus conditions for a directed fixed information exchange topology, so it is helpful to extend this algorithm to directed switching information-exchange typologies. In our article, the topology (edges) that describes the interconnections between nodes is considered time-varying rational transfer functions. Investigating the consensus conditions and the controllability for a multi-vehicle system might be one of the motivating applications of this work.

The consensus protocol, such as outlined in (Eq. (1)) and its variants, have been studied extensively in previously published academic studies. The findings from these studies have been applied in many domains, such as for formation motion control during time agents are mobile. However, the common intellectual idea of the consensus paradigm can be restrictive for researchers and engineers alike in several ways. For example, notice that we have interpreted the consensus problem as having integrating nodes and static weights. In the next section, will we extend this idea to networks with weights, which were transfer functions or nodes are more than just integrators. We will explore this dilemma by modeling heat transfer in buildings [16–19]. By the notation “dynamic systems,” we mean that linear ordinary differential equations (LODEs) are described as relationships between the system variables. We call such networks *dynamic consensus networks* because all the node variables converge to a common value called a consensus value under some conditions.

The Chapter is organized as follows: In Section 2, we present a general framework for a dynamic consensus network. We present a detailed study of modeling thermal processes in buildings as directed, dynamic graphs, beginning with a simple two-room model and transitioning to multiple interconnected rooms. Section 3 outlines a theoretical framework dedicated to these dynamic graphs and dynamic consensus networks. This framework will introduce the notion of a degree of dynamics, adjacency, incident, and Laplacian matrices in a way that naturally extends these concepts from a static case. By modeling this, one can easily define equivalent concepts of dynamic interconnection matrices and dynamic consensus networks.

2. Modeling a thermal process in a building as a directed dynamic graph

This Section first presents examples showing how a dynamic graph can arise in applications and then give a general framework for a dynamic consensus network. We present a detailed study of modeling thermal processes in buildings as directed,

dynamic graphs, beginning with a simple two-room model and transitioning to a model with multiple interconnected rooms. Motivated by this example, we then propose a mathematical framework in service of dynamic graphs and dynamic consensus networks.

Historically, there has always been a recognized need to model the energy processes within buildings. Typical examples of this modeling application are sizing HVAC equipment, determining energy usage performance, and optimizing energy management in a building through persistent control. Current state-of-the-art methods include modeling packages, such as Energy Plus [20], that allows users to specify a building's geometry, equipment, orientation, materials, and usage patterns, simulated using first principles models and simulated weather data. Though undoubtedly useful for design, these computationally complex systems may suffer from certain limitations once a building has been constructed due to significant deviations in construction, occupant use, and other specifications that cause the actual building's behavior to be quite different the model.

At the opposite extreme, so-called black box models have been developed from observational data. Though we can utilize these models to predict future values of particular variables, they do not incorporate any structural information about the system when gathering data, resulting in the need for large amounts of data to train and suffering from the difficulty of extracting relevant information about internal physical parameters that may be of interest.

Semi-physical models resulting in an intermediate level of modeling are known as gray-box modeling. Simple modeling elements containing parameters identified using observational data are chosen and connected based upon physical insight to represent the system's actual configuration. This is commonly the modeling technique used for thermal networks, which have been used to study load-shifting and peak-reducing control in buildings [21, 22]. A typical thermal network model for a single room is shown in **Figure 2**, which was adapted from [22]. These networks of (analogous) thermal resistors and capacitors model different building elements. To date, this has typically been performed at a very coarse level, sometimes by

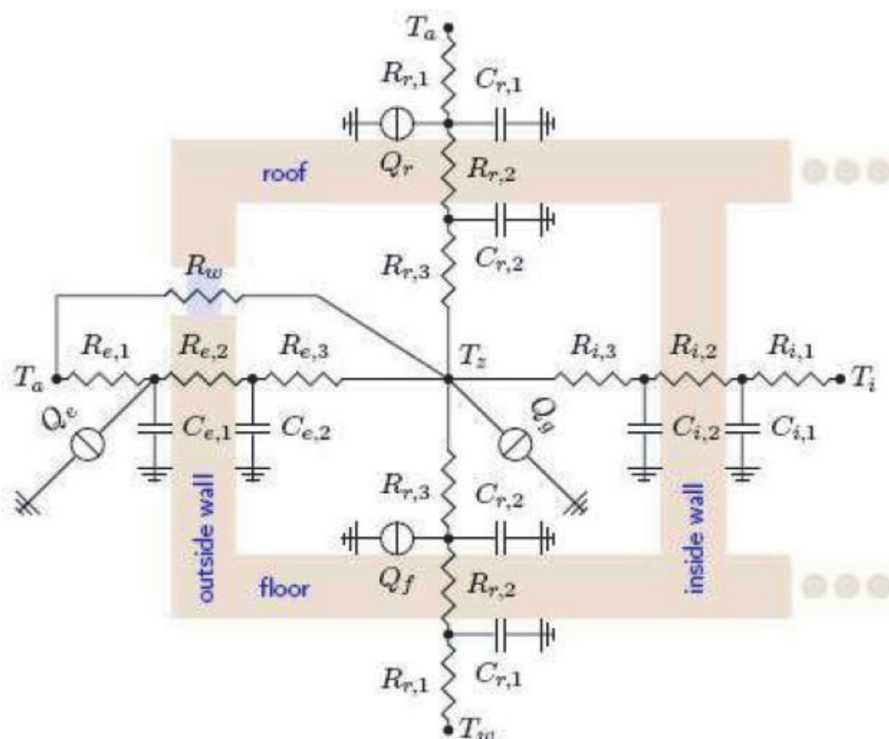


Figure 2.
Thermal model of a room.

combining multiple rooms into one practical room per zone. In [22], a gray-box model for an experimental building was created by utilizing measurements of weather, room temperature, and room air supply and flow. This model was used to predict the effects of a demand-limiting control strategy that we later validated experimentally.

This section uses a single-room model as shown in **Figure 2** from [22] as the basis for a node and its interconnections to other nodes to build up a dynamic graph representation of a building's thermal processes. First, we consider two rooms connected by a wall. We then illustrate how several such nodes may be interconnected, using the example of a hypothetical four-room building, with analysis provided of the resulting model that motivates the generalization in the next section.

Before proceeding, we note that the initial interest in modeling thermal processes in a building comes from viewing a building as a group of overlapping, interacting networks. In thermal networks, these nodes may denote rooms in an office or school, while in the control network, they may represent a sensor, actuator, or a unit of computation. The connections in the middle of nodes will represent variables that share information, such as the flow of air-conditioned air between rooms through walls, windows, and doors in a thermal network. While some typical graph-based networks consist of links that are in some way constant gains; some networks, such as a building's thermal network, may have dynamic links between nodes, as we will explore in the next section.

2.1 Two rooms connected by a wall

Figure 3 depicts what is called a 3R2C model in the literature [23]. We identify a room i as a node with node variable T_i , the lumped room temperature, Q_i^{in} , the input heat flow (a manipulated variable, not shown), and q_{ij} , the heat flow out of the room through walls or doors or windows (of course with this convention, if $q_{ij} < 0$ then this is heat flow into the room). The parameter C_i^r is the thermal capacity (mass) of the room i .

The interconnection between the two rooms is a wall represented analogously by an electrical circuit with three resistors and two capacitors, simplifying the model in **Figure 3**. The capacitors C_2 and C_4 can be thought of as the heat storage capacity of the wall's materials, which could be different on each side of the wall. A complete

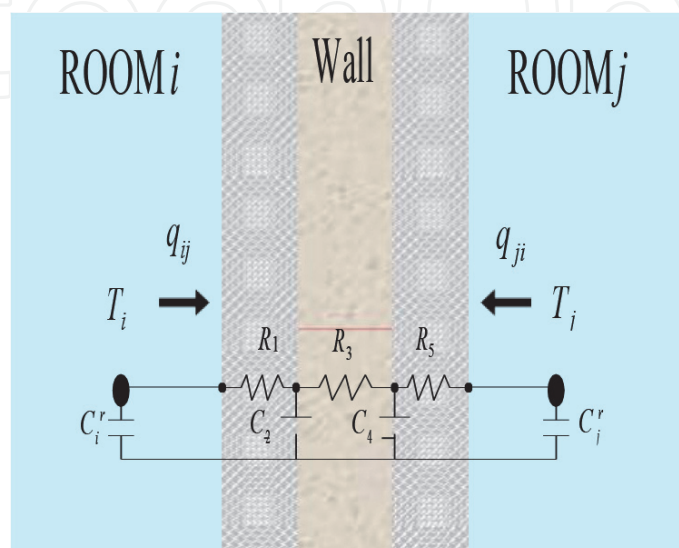


Figure 3.
Two rooms connected by a wall using the 3R2C model.

model adds a capacitor to represent the insulation properties in addition to that of the wall's board materials. For the resistors, R_3 represents the heat dissipation inside the wall, while R_1 and R_5 represent the heat dissipation from each room to the inside of the wall.

As shown in [23], the heat flows in **Figure 3** can be written as:

$$\begin{bmatrix} q_{ij} \\ q_{ji} \end{bmatrix} = \frac{1}{B_{ij}(s)} \begin{bmatrix} A_{ij}(s) & -D_{ij}(s) \\ -D_{ij}(s) & A_{ji}(s) \end{bmatrix} \begin{bmatrix} T_i \\ T_j \end{bmatrix} \quad (4)$$

where

$$A_{ij}(s) = 1 + a_1^{ij}s + a_2^{ij}s^2$$

$$A_{ji}(s) = 1 + a_1^{ji}s + a_2^{ji}s^2$$

$$B_{ij}(s) = b_0^{ij} + b_1^{ij}s + b_2^{ij}s^2$$

$$D_{ij}(s) = 1 + d_1^{ij}s + d_2^{ij}s^2$$

and

$$a_1^{ij} = C_4R_5 + C_2R_3 + C_2R_5$$

$$a_2^{ij} = C_4C_2R_5R_3$$

$$a_1^{ji} = C_4R_1 + C_4R_3 + C_2R_1$$

$$a_2^{ji} = C_4C_2R_3R_1$$

$$b_0^{ij} = R_5 + R_3 + R_1$$

$$b_1^{ij} = C_4R_5R_1 + C_2R_3R_1 + C_2R_5R_1 + C_4R_5R_3$$

$$b_2^{ij} = C_4C_2R_5R_3R_1$$

$$d_1^{ij} = d_2^{ij} = 0$$

Here, s is the independent variable of the Laplace transform, which can be interpreted as $s \doteq \frac{d}{dt}(\cdot)$. As noted in [23], we can interpret $G_x(s) = A_{ij}(s)/B_{ij}(s)$ as the external conduction of the wall, $G_y(s) = D_{ij}(s)/B_{ij}(s)$ as the cross-conduction of the wall, and $G_z(s) = A_{ji}(s)/B_{ij}(s)$ as the internal conduction of the wall.

From (3), the nodal equation can be written as:

$$C_i^r \frac{dT_i}{dt} = Q_i^{in} - q_{ij} \quad (5)$$

Combining (Eqs. (4) and (5)) gives:

$$\begin{bmatrix} C_i^r s & 0 \\ 0 & C_j^r s \end{bmatrix} \begin{bmatrix} T_i \\ T_j \end{bmatrix} = \begin{bmatrix} Q_1^{in} \\ Q_2^{in} \end{bmatrix} - \frac{1}{B_{ij}(s)} \begin{bmatrix} A_{ij}(s) & -D_{ij}(s) \\ -D_{ij}(s) & A_{ji}(s) \end{bmatrix} \begin{bmatrix} T_i \\ T_j \end{bmatrix} \quad (6)$$

To motivate later analysis, notice that in the absence of any external heat inputs (i.e., $Q_i^{in} = 0$), we can rewrite the previous equation as:

$$\begin{bmatrix} sC_i^r T_i(s) \\ sC_i^r T_j(s) \end{bmatrix} = -\frac{1}{B_{ij}(s)} \begin{bmatrix} A_{ij}(s) & -D_{ij}(s) \\ -D_{ij}(s) & A_{ij}(s) \end{bmatrix} \begin{bmatrix} T_i \\ T_j \end{bmatrix}, \quad (7)$$

which defines the relationship between the temperatures in two rooms using the 3R2C model.

It is useful to separate (Eq. (7)) as

$$\begin{aligned} T_i(s) &= -\frac{1}{C_i^r s} \left[\frac{A_{ij}(s)}{B_{ij}(s)} T_i(s) - \frac{D_{ij}(s)}{B_{ij}(s)} T_j(s) \right] \\ &= -\frac{1}{s} \left[\lambda_{ij}^S(s) T_i(s) - \lambda_{ij}^C(s) T_j(s) \right], \end{aligned} \quad (8)$$

where $\lambda_{ij}^S(s) = \frac{A_{ij}(s)}{B_{ij}(s)}$ is a self-correction weight and $\lambda_{ij}^C(s) = \frac{D_{ij}(s)}{B_{ij}(s)}$ is a cross-correction weight. Note that we assume C_i^r in (Eq. (8)) is equal to the unity for simplicity. Comparing this to static-weights consensus networks, we see that this appears similar to the equation of a two-node consensus network, in which the nodes are integrators. The difference is that there is a separate weighting on the terms $T_i(s)$ and $T_j(s)$, and these weights are dynamic.

2.2 Several interconnected rooms

This subsection uses the previous subsection's expressions to develop a building model with several interconnected rooms with different possible pathways between each room and the outside environment. Ideas are developed for a specific theoretical four-room building shown in **Figure 4** with each room having neighboring rooms with which heat can travel through-and-from. One such neighbor is always the external environment (which does not include the rooms), whose variable is denoted T_a with 'a' referring to the ambient outdoor temperature variable. Walls, doors, and windows are all pathways that thermal energy can potentially flow in such a network. Therefore, the corresponding graph for this example is shown in **Figure 5**. There are several heat flows that are not shown, including $q_{14} = q_{14}^{door}$, $q_{41} = q_{41}^{door}$, $q_{24} = q_{24}^{wall} + q_{24}^{door}$ and $q_{42} = q_{42}^{wall} + q_{42}^{door}$.

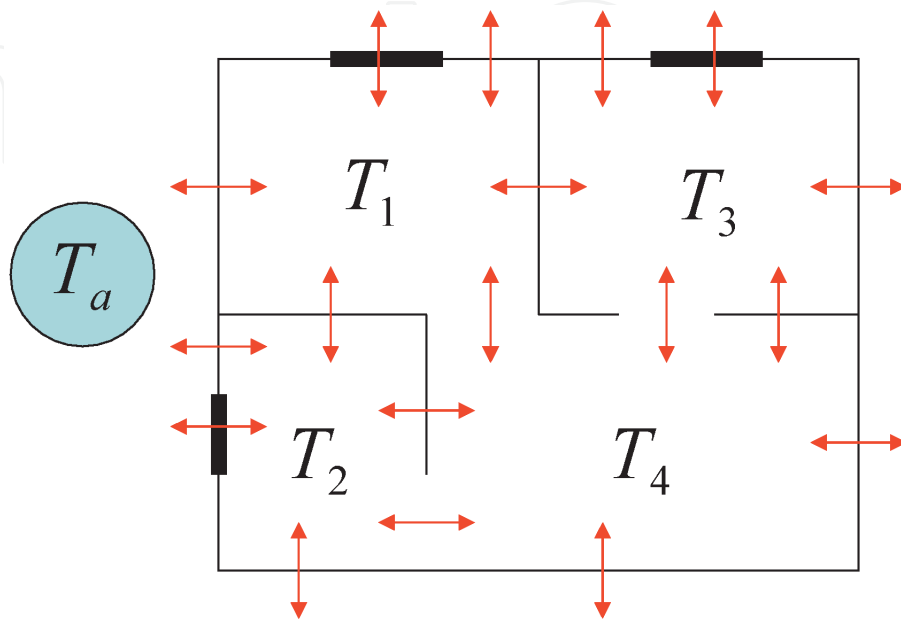


Figure 4.
A hypothetical four-room example.

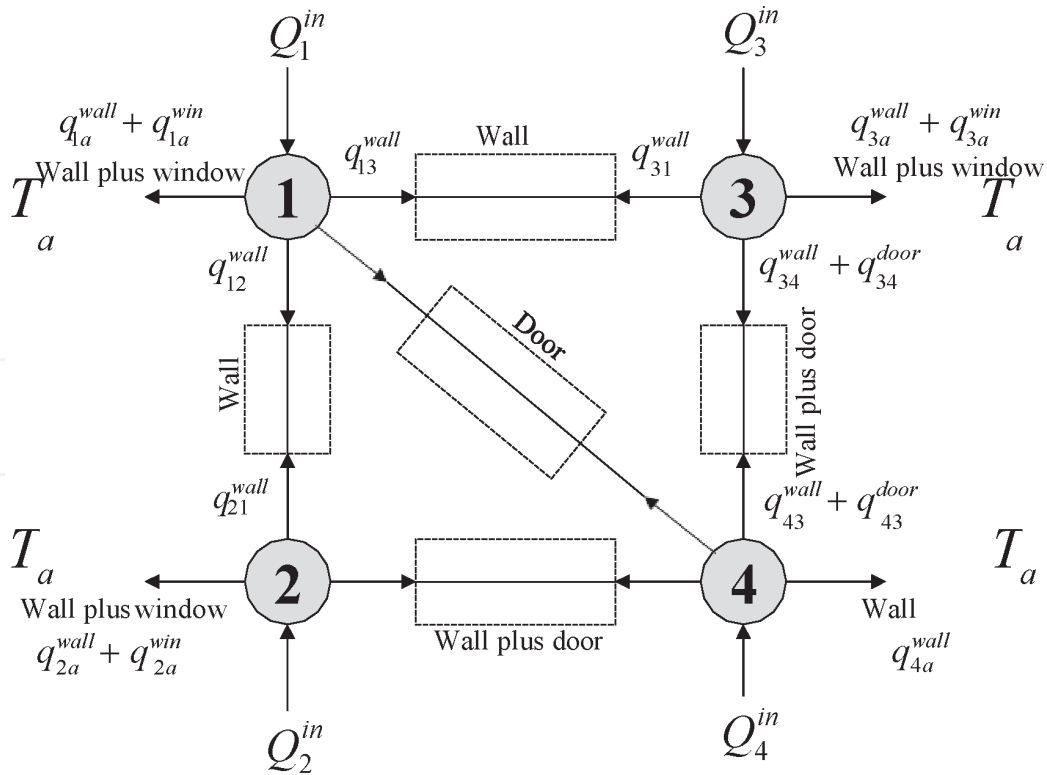


Figure 5.
 Heat flow network corresponding to the four-room example.

In developing a model for this system, we modify (Eq. (5)) to sum the energy losses through all pathways connected to a node, resulting in:

$$C_i^r \frac{dT_i}{dt} = Q_i^{\text{in}} - \sum_{j \in \mathcal{N}_i} \sum_{k_j \in \mathcal{P}_j} q_{ij}^{k_j} \quad (9)$$

where \mathcal{N}_i is the set of neighbors to which a node i is connected and \mathcal{P}_j is the set of pathways k_j associated with any neighbor j of node i . We note that $q_{ij} = \sum_{k_j \in \mathcal{P}_j} q_{ij}^{k_j}$.

For building thermal analysis, there may be several different types of interconnection elements, though they will all have the basic format of (Eq. (4)). Because there is negligible energy storage in doorways and windows, when these are the sole interconnection elements between rooms, we use a single R model, so that (Eq. (4)) is expressed with $b_0^{ij} = R$ with all other variables being set to zero. To make the notation a bit more uniform, for the common case when a door or window is in parallel with a wall, the interconnection transfer function matrix is the sum of the single R model and the 3R2C model. That is, the model would be given by A_{ij}, A_{ji}, B_{ij} , and D_{ij} that satisfies:

$$\begin{bmatrix} \frac{A_{ij}(s)}{B_{ij}(s)} & -\frac{D_{ij}(s)}{B_{ij}(s)} \\ -\frac{D_{ij}(s)}{B_{ij}(s)} & \frac{A_{ji}(s)}{B_{ij}(s)} \end{bmatrix} = \begin{bmatrix} \frac{A'_{ij}(s)}{B'_{ij}(s)} & -\frac{1}{B'_{ij}(s)} \\ -\frac{1}{B'_{ij}(s)} & \frac{A'_{ij}(s)}{B'_{ij}(s)} \end{bmatrix} + \begin{bmatrix} \frac{1}{R} & -\frac{1}{R} \\ -\frac{1}{R} & \frac{1}{R} \end{bmatrix}, \quad (10)$$

where the primed variables represent the 3R2C model, and the unprimed variables represent the resulting parallel connection. In the expressions below, we

assume that this computation has been done and the resulting unprimed coefficients can be easily calculated and are thus omitted here. Note that in the case of a door or window that is parallel to a wall, the coefficients d_1^{ij} and d_2^{ij} associated with the unprimed variables are non-zero.

Table 1 summarizes the neighbors for each node and the pathways between each node and each of its neighbors for this example. The table also identifies the coefficients used in the transfer matrix describing the interconnection between each pair of neighbors where the various polynomials A_{ij} , B_{ij} , D_{ij} (when there is no parallel door or wall) or A'_{ij} , B'_{ij} , D'_{ij} (before being combined in parallel with any doors or windows) are defined as above in (Eq. (4)).

Combining (Eqs. (4) and (9)) for the configuration shown in **Figure 5** with the parameters shown in **Table 1** and defining the vectors

$$T(s) = [T_1(s) \ T_2(s) \ T_3(s) \ T_4(s)]^T,$$

$$Q^{in}(s) = [Q_1^{in}(s) \ Q_2^{in}(s) \ Q_3^{in}(s) \ Q_4^{in}(s)]^T,$$

we can easily show that:

$$sT(s) = Q^{in}(s) - L(s)T(s), \quad (11)$$

Node	Neighbors	Paths	Coefficients
1	2	1-wall	A_{12}, B_{12}
		2-wall	A'_{1a}, B'_{1a}
	1-window	R_{w1a}	
	3	1-wall	A_{13}, B_{13}
		4	1-door
2	1	1-wall	D_{12}, B_{12}
		2-wall	A'_{2a}, B'_{2a}
	1-window	$R_{w2a} = B_{2a}$	
	4	2-wall	A'_{24}, B'_{24}
		1-door	$R_{d24} = B_{24}$
3	1	1-wall	D_{13}, B_{13}
		2-wall	A'_{3a}, B'_{3a}
	1-window	$R_{w3a} = B_{3a}$	
	4	2-wall	A'_{34}, B'_{34}
		1-door	$R_{d34} = B_{34}$
4	1	1-door	$R_{d41} = B_{41}$
	a	1-wall	A'_{4a}, B'_{4a}
		2-wall	D'_{24}, B'_{24}
	2	1-door	$R_{d42} = B_{42}$
		3	2-wall
1-door	$R_{d43} = B_{43}$		

Table 1.
Hypothetical four room example.

where the matrix $L(s) = [L_{ij}(s)]$ is given as:

$$L_{ij}(s) = \begin{cases} \sum_{j \in \mathcal{N}_i} \lambda_{ij}^S(s) & i = j \\ -\lambda_{ij}^C(s) & i \neq j \text{ and } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise,} \end{cases} \quad (12)$$

or $L(s)$ is given as (Eq. (13)). We will refer to $L(s)$ defined in this way as

$$L(s) = \begin{bmatrix} \frac{1}{R_{14}^d} + \sum_{j=2,3} \frac{A_{1j}}{B_{1j}} & -\frac{1}{B_{12}} & -\frac{1}{B_{13}} & -\frac{1}{R_{14}^d} \\ -\frac{1}{B_{21}} & \frac{1}{R_{24}^d} + \sum_{j=1,4} \frac{A_{2j}}{B_{2j}} & 0 & -\frac{1}{B_{24}} - \frac{1}{R_{24}^d} \\ -\frac{1}{B_{31}} & 0 & \frac{1}{R_{34}^d} + \sum_{j=1,4} \frac{A_{3j}}{B_{3j}} & -\frac{1}{B_{34}} - \frac{1}{R_{34}^d} \\ -\frac{1}{R_{41}^d} & \frac{-1}{B_{42}} - \frac{1}{R_{42}^d} & -\frac{1}{B_{43}} - \frac{1}{R_{43}^d} & \frac{1}{R_{41}^d} + \frac{1}{R_{42}^d} + \frac{1}{R_{43}^d} + \sum_{j=2,3} \frac{A_{4j}}{B_{4j}} \end{bmatrix} \quad (13)$$

a *dynamic Laplacian matrix*. For the graph topology shown in **Figure 5**, the dynamic Laplacian matrix has the form shown in (Eq. (13)). In the previous work [16], We have shown that when the weight matrices $\lambda_{ij}(s)$ satisfy certain assumptions, $-L(s)$ can be viewed as a *dynamic interconnection matrix*, allowing the demonstration of consensus.

Notice that we can redraw **Figure 5** as shown in **Figure 6**, where

$$\lambda_{ij}(s) = \begin{bmatrix} \lambda_{ij}^S(s) & -\lambda_{ij}^C(s) \end{bmatrix} \quad (14)$$

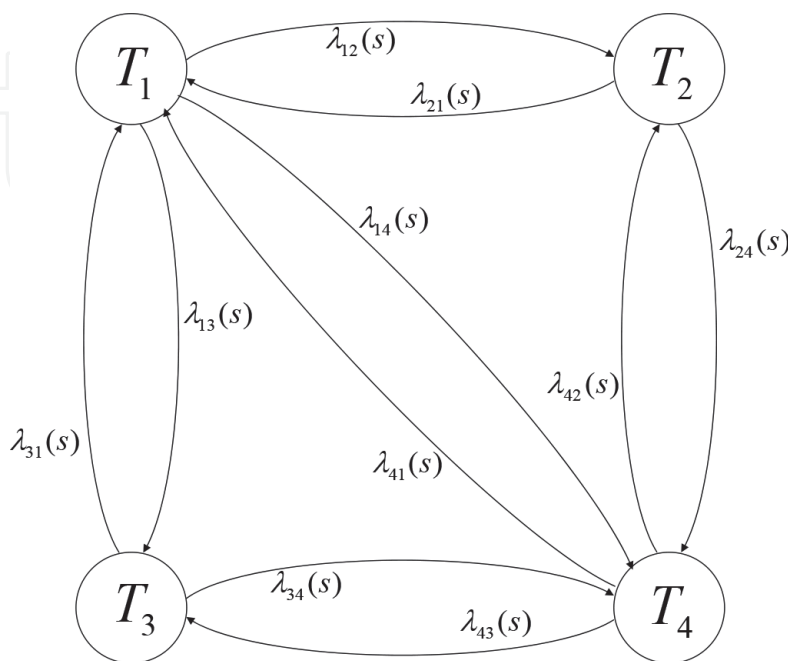


Figure 6.
 A hypothetical four-room example as a dynamic consensus network.

$$= \left[\begin{array}{cc} \left(\frac{A_{ij}}{B_{ij}}\right)_w + \frac{1}{R_{ij}^d} & -\left(\frac{1}{B_{ij}}\right)_w - \frac{1}{R_{ij}^d} \end{array} \right], \quad (15)$$

The graph shown in **Figure 6** will be referred to as a *dynamic graph* or a *dynamic consensus network*. Then applying the definition of the dynamic.

Laplacian (12) for the dynamic graph **Figure 6** we get

$$L(s) = \begin{bmatrix} \sum_{j=2,3,4} \lambda_{1j}^S(s) & -\lambda_{12}^C(s) & -\lambda_{13}^C(s) & -\lambda_{14}^C(s) \\ -\lambda_{21}^C(s) & \sum_{j=1,4} \lambda_{2j}^S(s) & 0 & -\lambda_{24}^C(s) \\ -\lambda_{31}^C(s) & 0 & \sum_{j=1,4} \lambda_{3j}^S(s) & -\lambda_{34}^C(s) \\ -\lambda_{41}^C(s) & -\lambda_{24}^C(s) & -\lambda_{43}^C(s) & \sum_{j=1,2,3} \lambda_{4j}^S(s) \end{bmatrix}, \quad (16)$$

which reduces to (Eq. (13)) if we insert the full expressions for $\lambda_{ij}^C(s)$ and $\lambda_{ij}^S(s)$ defined in (Eq. (14)). This leads us to consider the idea of dynamic consensus networks.

Figure 7 shows a simple simulation of (Eq. (9)) for the case when $Q_i^{in} = 0$ and the scalar $T_a = 80$ for a nominal set of parameters available from the authors upon request (omitted here in the interest of space). As intuitively expected, all temperatures converge to the ambient temperature, as the external environment has infinite capacity and there is no energy input or removal. This can also be seen by examining (Eq. (9)) at steady-state with $Q_{in} = 0$. Further noting that $L(0)$ has the form of a classic graph Laplacian matrix, with row sum equal to zero (and in this case column sum equal zero as well), we can argue that $T^{ss} = T_a$ is a unique solution.

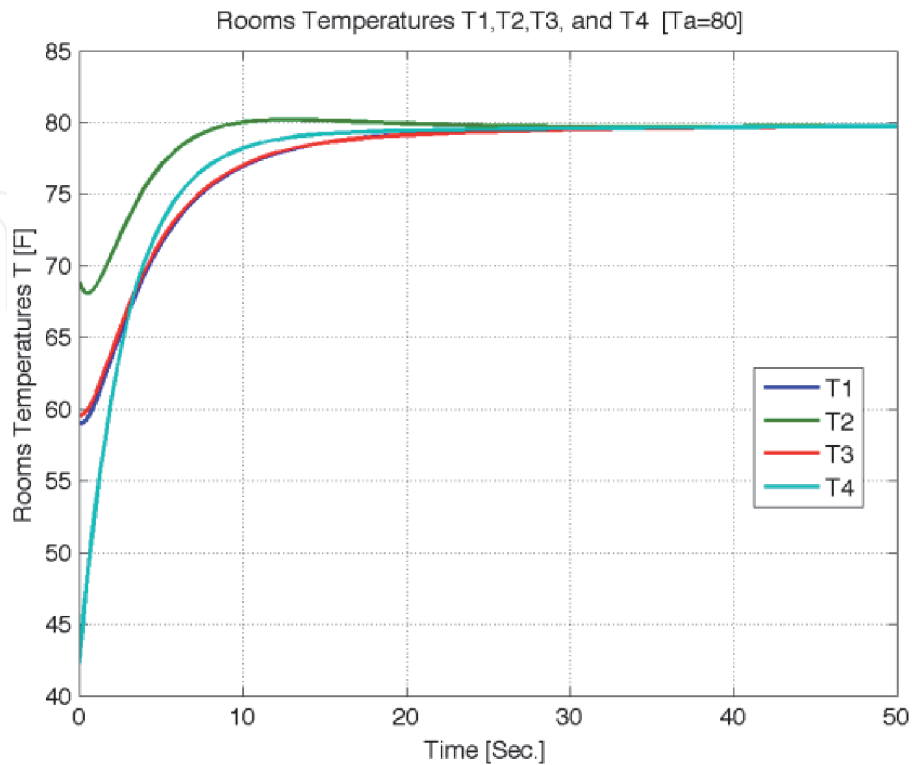


Figure 7.
Example simulation.

We also [24] consider another example that motivated a generalization of the static consensus problem (Eq. (1)), modeling the load frequency control (LFC) network of an electrical power grid as a dynamic consensus network. We consider the following network:

$$Y_i(s) = \frac{1}{s} \sum_{j \in \mathcal{N}_i} G_i(s) a_{ij} (Y_i(s) - Y_j(s)), \quad (17)$$

$i = 1, \dots, N$, which can be viewed as a single-integrator consensus network with dynamic interconnection coefficients $G_i(s) a_{ij}$. In the grid's LFC network, each system's output is the phase of its voltage, which is the integration of the angular velocity. The interconnection is power exchanges among the individual systems through transmission lines dependent on phase differences.

Based on the dynamics of a network's nodes and their topology, several consensus problems can be specified. This Chapter focuses on two types of dynamic consensus networks: directed and undirected. The dynamic consensus networks studied are:

- **Dynamic Network 1:** Directed dynamic networks with integrator nodes and dynamic edges:

$$\dot{x}_i(t) = - \sum_{j \in \mathcal{N}_i} \left[\lambda_{ij}^S(t) * x_i(t) - \lambda_{ij}^C(t) * x_j(t) \right], \quad (18)$$

or,

$$x_i(s) = - \frac{1}{s} \sum_{j \in \mathcal{N}_i} \left[\lambda_{ij}^S(s) x_i(s) - \lambda_{ij}^C(s) x_j(s) \right] \quad (19)$$

- **Dynamic Network 2:** Undirected dynamic networks with integrator nodes and strictly-positive-real (SPR) transfer function edges:

$$\dot{x}_i(t) = - \sum_{j \in \mathcal{N}_i} \left[\lambda_{ij}(t) * (x_i(t) - x_j(t)) \right], \quad (20)$$

or,

$$x_i(s) = - \frac{1}{s} \sum_{j \in \mathcal{N}_i} \left[\lambda_{ij}(s) (x_i(s) - x_j(s)) \right] \quad (21)$$

- **Dynamic Network 3:** Undirected dynamic networks with identical nodes and dynamic edges:

$$x_i(t) = -p(t) * \sum_{j \in \mathcal{N}_i} \left[\lambda_{ij}(t) * (x_i(t) - x_j(t)) \right], \quad (22)$$

or,

$$x_i(s) = -p(s) \sum_{j \in \mathcal{N}_i} \left[\lambda_{ij}(s) (x_i(s) - x_j(s)) \right] \quad (23)$$

- **Dynamic Network 4:** Undirected dynamic networks with heterogeneous nodes and dynamic edges:

$$x_i(t) = -p_i(t) * \sum_{j \in \mathcal{N}_i} [\lambda_{ij}(t) * (x_i(t) - x_j(t))], \quad (24)$$

$$x_i(s) = -p_i(s) \sum_{j \in \mathcal{N}_i} [\lambda_{ij}(s) (x_i(s) - x_j(s))] \quad (25)$$

Assumptions: For the dynamic networks (Eqs. (19)–(25)), we make the following assumptions:

1. The node and edge processing in the proposed dynamic networks (Eqs. (19)–(25)) are linear, time-invariant LTI.
2. The dynamic topology consists of dynamic edges $\lambda_{ij}(s)$ modeled as transfer functions. For the second proposed dynamic network (Eq. (21)), we assume the edges' dynamics are strictly positive real (SPR) transfer functions.
3. The topology of a network can be directed or undirected. The first dynamic network (Eq. (19)) uses a directed topology, whereas the second dynamic network (Eq. (21)) uses an undirected topology.
4. Depending on the application, the flow is modeled differently. For instance, $[\lambda_{ij}^S(s)x_i(s) - \lambda_{ij}^C(s)x_j(s)]$ and $[\lambda_{ij}(s)(x_i(s) - x_j(s))]$ are two different ways of modeling flow, as is indicated by the previous Section. The difference between these two cases is illustrated in (Eqs. (19) and (21)). These flow models are rooted in the types of dynamic networks to be modeled per the motivation for each network. The first dynamic network (Eq. (19)) is based upon modeling buildings' thermal processes as directed dynamic graphs. In contrast, the second dynamic network (Eq. (21)) is based upon the motivation of modeling micro-grids of power systems as undirected dynamic graphs.
5. The nodes' dynamics can be integrators (Eqs. (19) and (21)) or more general dynamics Eqs. (23) and (25).
6. The nodes' dynamics and the edges can be identical Eqs. (23) or heterogeneous Eqs. (25).
7. These models are often autonomous, meaning no input flows into the dynamic consensus networks. However, we add inputs and disturbances to the proposed dynamic consensus networks' general forms in some problems.

3. Dynamic graphs definitions

Consider the example in **Figure 8**. These graphical depictions are outlined as a set of nodes (or vertices) $\mathcal{N} = \{n_i\}$ connected by a set of edges $\mathcal{E}(s) = \{(n_i, n_j) : n_i, n_j \in \mathcal{N}\}$. We modeled each respective edge as a transfer function $\lambda_{ij}(s)$. In this example, each edge $\lambda_{ij}(s) \in \mathcal{R}(s)$, where $\mathcal{R}(s)$ denotes the set of all complex-valued functions analytic in the open right-half complex plane (real rational functions). It is important to note that this model assumes that zero self-loops are connected to any node in question. Meanwhile, if an edge between nodes exists n_i and n_j , we describe such nodes as adjacent (or neighbors). We denote the neighbors of each node n_i as $\mathcal{N}_i = \{j : (n_i, n_j) \in \mathcal{E}(s)\}$. As previous academic work has

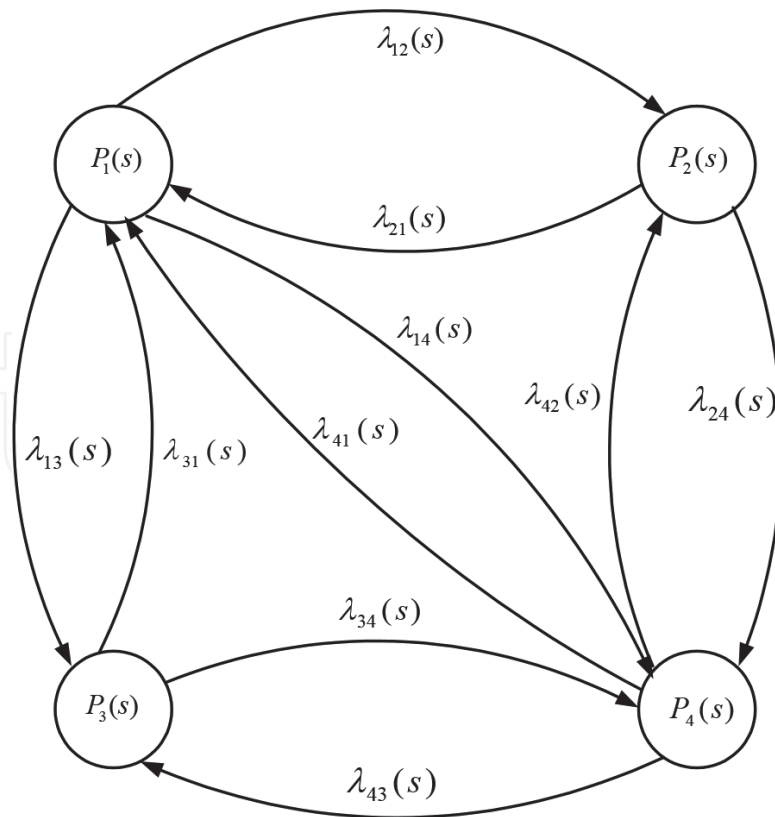


Figure 8.
 Directed-dynamic graph.

described, the path between two respective nodes consists of a sequence of edges from which it is possible to travel along the arc sequence from one of the nodes to the other. If there is at least one node with at least one path to every other node, the graph is connected.

Later, we also view a node as implementing a transfer function that produces the node variable ($P_i(s)$ for $i = 1, 2, \dots, N$, where N is the number of nodes in the dynamic graph).

As in the static case, we ordered the edges $e_{ij}(s)$ by the edge originating from node $P_i(s)$, known as the tail node, and terminating at node $P_j(s)$, which we referred as the head node, which a compass can identify. If the dynamics of the nodes are different, we will call such dynamic networks heterogeneous networks. If the nodes have the same dynamics, the network is homogeneous. However, a particular case does arise when integrator nodes with dynamic edges are present.

Each node $P_i(s)$ in a directed dynamic network, such as **Figure 8**, is associated with a dynamic degree $v_i(s)$ representing the total sum of the dynamic edge weights that are connected to the node i (entering or leaving the node). More specifically, each node has a dynamic in-degree $v_i^{in}(s)$ and a dynamic out-degree $v_i^{out}(s)$ representing the sum of the dynamic edge weights of the incoming and outgoing edges, respectively. Clearly $v_i(s) = v_i^{in}(s) + v_i^{out}(s)$. From these dynamic degree definitions, we can define three different dynamic degree matrices:

1. The dynamic in-degree matrix $D^{in}(s) = \text{diag}(v_i^{in}(s))$.
2. The dynamic out-degree matrix $D^{out}(s) = \text{diag}(v_i^{out}(s))$.
3. The dynamic degree matrix $D(s) = \text{diag}(v_i(s))$.

Notice that $D(s) = D^{in}(s) + D^{out}(s)$. To illustrate, for the example shown in **Figure 8**, these matrices are given by (26):

$$\begin{aligned}
 D^{in}(s) &= \begin{bmatrix} \lambda_{21}(s) + \lambda_{31}(s) + \lambda_{41}(s) & 0 & 0 & 0 \\ 0 & \lambda_{12}(s) + \lambda_{42}(s) & 0 & 0 \\ 0 & 0 & \lambda_{13}(s) + \lambda_{43}(s) & 0 \\ 0 & 0 & 0 & \lambda_{14}(s) + \lambda_{24}(s) + \lambda_{34}(s) \end{bmatrix}; \\
 D^{out}(s) &= \begin{bmatrix} \lambda_{12}(s) + \lambda_{13}(s) + \lambda_{14}(s) & 0 & 0 & 0 \\ 0 & \lambda_{21}(s) + \lambda_{24}(s) & 0 & 0 \\ 0 & 0 & \lambda_{31}(s) + \lambda_{34}(s) & 0 \\ 0 & 0 & 0 & \lambda_{41}(s) + \lambda_{42}(s) + \lambda_{43}(s) \end{bmatrix}; \\
 D(s) &= \begin{bmatrix} D^{in}(1,1) + D^{out}(1,1) & 0 & 0 & 0 \\ 0 & D^{in}(2,2) + D^{out}(2,2) & 0 & 0 \\ 0 & 0 & D^{in}(3,3) + D^{out}(3,3) & 0 \\ 0 & 0 & 0 & D^{in}(4,4) + D^{out}(4,4) \end{bmatrix}.
 \end{aligned} \tag{26}$$

If a dynamic edge $e_{ij}(s)$ exists between two nodes i and j , these nodes are considered to be adjacent and are known as neighbors, and are denoted for a node $P_i(s)$ by $\mathcal{N}_i = \{j : (n_i, n_j) \in \mathcal{E}(s)\}$. As before, neighbors can be distinguished based upon whether they are associated with incoming or outgoing arcs. Thus, we can define three dynamic adjacency matrices:

1. The incoming dynamic adjacency matrix $A^{in}(s) = [a_{ij}^{in}(s)]$, is defined by

$$a_{ij}^{in}(s) = \begin{cases} \sum_{j \in \mathcal{N}_i} e_{ij}(s); \text{ (coming into } n_i \text{ from } n_j) & \text{if } i \neq j \\ 0 & \text{otherwise.} \end{cases}$$

2. The dynamic outgoing adjacency matrix $A^{out}(s) = [a_{ij}^{out}(s)]$, is defined by

$$a_{ij}^{out}(s) = \begin{cases} \sum_{j \in \mathcal{N}_i} e_{ij}(s); \text{ (going out from } n_i \text{ into } n_j) & \text{if } i \neq j \\ 0 & \text{otherwise.} \end{cases}$$

3. The dynamic adjacency matrix $A(s) = [a_{ij}(s)]$, is defined by

$$a_{ij}(s) = \begin{cases} \text{total } \sum_{j \in \mathcal{N}_i} e_{ij}(s); \text{ (between } n_i \text{ and } n_j) & \text{if } i \neq j \\ 0 & \text{otherwise.} \end{cases}$$

Notice that $A(s) = A^{in}(s) + A^{out}(s)$. To illustrate, for the example shown in **Figure 8**, these matrices are given by:

$$\begin{aligned}
 A^{in}(s) &= \begin{bmatrix} 0 & \lambda_{21}(s) & \lambda_{31}(s) & \lambda_{41}(s) \\ \lambda_{12}(s) & 0 & 0 & \lambda_{42}(s) \\ \lambda_{13}(s) & 0 & 0 & \lambda_{43}(s) \\ \lambda_{14}(s) & \lambda_{24}(s) & \lambda_{34}(s) & 0 \end{bmatrix}; \\
 A^{out}(s) &= \begin{bmatrix} 0 & \lambda_{12}(s) & \lambda_{13}(s) & \lambda_{14}(s) \\ \lambda_{21}(s) & 0 & 0 & \lambda_{24}(s) \\ \lambda_{31}(s) & 0 & 0 & \lambda_{34}(s) \\ \lambda_{41}(s) & \lambda_{42}(s) & \lambda_{43}(s) & 0 \end{bmatrix}; \\
 A(s) &= \begin{bmatrix} 0 & \lambda_{12}(s) + \lambda_{21}(s) & \lambda_{13}(s) + \lambda_{31}(s) & \lambda_{14}(s) + \lambda_{41}(s) \\ \lambda_{21}(s) + \lambda_{12}(s) & 0 & 0 & \lambda_{24}(s) + \lambda_{42}(s) \\ \lambda_{31}(s) + \lambda_{13}(s) & 0 & 0 & \lambda_{34}(s) + \lambda_{43}(s) \\ \lambda_{41}(s) + \lambda_{14}(s) & \lambda_{42}(s) + \lambda_{24}(s) & \lambda_{43}(s) + \lambda_{34}(s) & 0 \end{bmatrix}.
 \end{aligned}
 \tag{27}$$

Another type of matrix is the dynamic incident matrix. For that matrix, we outline two incident matrices: one that indicates the direction of the edges connected to a node, where for node $P_i(s)$ the edge $e_{ij}(s)$ is given a value based upon being disconnected, incoming, or outgoing and denoted as B_S^{in} where “S” refers to static. Meanwhile, another can matrix capture the transfer functions of the edges and is with this denoted as $B_D^{in}(s)$ where “D” refers to the dynamic. Thus, the dynamic and static incoming incident matrices are defined as $B_D^{in}(s) = [b_{ij-D}^{in}(s)]$, $B_S^{in} = [b_{ij-S}^{in}]$, where,

$$b_{ij-D}^{in}(s) = \begin{cases} +\lambda_{ij}(s) & \text{if arc } j \text{ enters node } n_i \\ 0 & \text{otherwise.} \end{cases}$$

$$b_{ij-S}^{in} = \begin{cases} +1 & \text{if arc } j \text{ enters node } n_i \\ 0 & \text{otherwise.} \end{cases}$$

To illustrate, for the example shown in **Figure 8** these matrices are:

$$\begin{aligned}
 B_D^{in}(s) &= \begin{bmatrix} 0 & \lambda_{21}(s) & 0 & \lambda_{31}(s) & 0 & \lambda_{41}(s) & 0 & 0 & 0 & 0 \\ \lambda_{12}(s) & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_{42}(s) & 0 & 0 \\ 0 & 0 & \lambda_{13}(s) & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_{43}(s) \\ 0 & 0 & 0 & 0 & \lambda_{14}(s) & 0 & \lambda_{24}(s) & 0 & \lambda_{34}(s) & 0 \end{bmatrix} \\
 B_S^{in} &= \begin{bmatrix} 0 & +1 & 0 & +1 & 0 & +1 & 0 & 0 & 0 & 0 \\ +1 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & 0 & 0 \\ 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & +1 \\ 0 & 0 & 0 & 0 & +1 & 0 & +1 & 0 & +1 & 0 \end{bmatrix}
 \end{aligned} \tag{28}$$

Similarly, we can define a dynamic and static outgoing incident matrices for a dynamic graph by, $B_D^{out}(s) = [b_{ij-D}^{out}(s)]$, $B_S^{out} = [b_{ij-S}^{out}]$, where,

$$\begin{aligned}
 b_{ij-D}^{out}(s) &= \begin{cases} -\lambda_{ij}(s) & \text{if arc } j \text{ leaves node } n_i \\ 0 & \text{otherwise.} \end{cases} \\
 b_{ij-S}^{out} &= \begin{cases} -1 & \text{if arc } j \text{ leaves node } n_i \\ 0 & \text{otherwise.} \end{cases}
 \end{aligned}$$

To illustrate, for the example shown in **Figure 8**, these matrices are

$$\begin{aligned}
 B_D^{out}(s) &= \begin{bmatrix} -\lambda_{12}(s) & 0 & -\lambda_{13}(s) & 0 & -\lambda_{14}(s) & 0 & 0 & 0 & 0 & 0 \\ 0 & -\lambda_{21}(s) & 0 & 0 & 0 & 0 & -\lambda_{24}(s) & 0 & 0 & 0 \\ 0 & 0 & 0 & -\lambda_{31}(s) & 0 & 0 & 0 & 0 & -\lambda_{34}(s) & 0 \\ 0 & 0 & 0 & 0 & 0 & -\lambda_{41}(s) & 0 & -\lambda_{42}(s) & 0 & -\lambda_{43}(s) \end{bmatrix} \\
 B_S^{out} &= \begin{bmatrix} -1 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & -1 \end{bmatrix}
 \end{aligned} \tag{29}$$

Also, we can define the dynamic and static incident matrices for a directed dynamic graph as $B_D(s) = [b_{ij-D}(s)]$, $B_S = [b_{ij-S}]$ where

$$\begin{aligned}
 b_{ij-D}(s) &= \begin{cases} -\lambda_{ij}(s) & \text{if arc } j \text{ leaves nodes } n_i \\ +\lambda_{ij}(s) & \text{if arc } j \text{ enters node } n_i \\ 0 & \text{otherwise.} \end{cases} \\
 b_{ij-S} &= \begin{cases} -1 & \text{if arc } j \text{ leaves node } n_i \\ +1 & \text{if arc } j \text{ enters node } n_i \\ 0 & \text{otherwise.} \end{cases}
 \end{aligned}$$

Notice that $B_D(s) = B_D^{in}(s) + B_D^{out}(s)$ and $B_S = B_S^{in} + B_S^{out}$. To illustrate, for the example shown in **Figure 8**, the dynamic and static incident matrices are:

$$\begin{aligned}
 B_D(s) &= \begin{bmatrix} -\lambda_{12}(s) & \lambda_{21}(s) & -\lambda_{13}(s) & \lambda_{31}(s) & -\lambda_{14}(s) & \lambda_{41}(s) & 0 & 0 & 0 & 0 \\ \lambda_{12}(s) & -\lambda_{21}(s) & 0 & 0 & 0 & 0 & -\lambda_{24}(s) & \lambda_{42}(s) & 0 & 0 \\ 0 & 0 & \lambda_{13}(s) & -\lambda_{31}(s) & 0 & 0 & 0 & 0 & -\lambda_{34}(s) & \lambda_{43}(s) \\ 0 & 0 & 0 & 0 & \lambda_{14}(s) & -\lambda_{41}(s) & \lambda_{24}(s) & -\lambda_{42}(s) & \lambda_{34}(s) & -\lambda_{43}(s) \end{bmatrix} \\
 B_S &= \begin{bmatrix} -1 & 1 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & -1 & 1 & -1 \end{bmatrix}
 \end{aligned} \tag{30}$$

We can now give the definition of dynamic Laplacian matrix equivalent with the derived dynamic degree, adjacency, and incident matrices. This matrix mentioned above has spectral properties that indicate many conditions of a graph. An undirected dynamic graph has a corresponding dynamic Laplacian matrix defined by $L(s) = D(s) - A(s)$. More specifically, the dynamic Laplacian matrix we propose here can be defined as $L(s) = [l_{ij}(s)]$, where

$$l_{ij}(s) = \begin{cases} \sum_{j \in \mathcal{N}_i} \lambda_{ij}(s) & i = j \\ -\lambda_{ij}(s) & i \neq j \text{ and } (i, j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \tag{31}$$

$$\begin{aligned}
 D^{out}(s) &= \begin{bmatrix} \lambda_{12}(s) + \lambda_{13}(s) + \lambda_{14}(s) & 0 & 0 & 0 \\ 0 & \lambda_{21}(s) + \lambda_{24}(s) & 0 & 0 \\ 0 & 0 & \lambda_{31}(s) + \lambda_{34}(s) & 0 \\ 0 & 0 & 0 & \lambda_{41}(s) + \lambda_{42}(s) + \lambda_{43}(s) \end{bmatrix}, \\
 A^{out}(s) &= \begin{bmatrix} 0 & \lambda_{12}(s) & \lambda_{13}(s) & \lambda_{14}(s) \\ \lambda_{21}(s) & 0 & 0 & \lambda_{24}(s) \\ \lambda_{31}(s) & 0 & 0 & \lambda_{34}(s) \\ \lambda_{41}(s) & \lambda_{42}(s) & \lambda_{43}(s) & 0 \end{bmatrix}, \\
 L^{out}(s) &= \begin{bmatrix} \lambda_{12}(s) + \lambda_{13}(s) + \lambda_{14}(s) & -\lambda_{12}(s) & -\lambda_{13}(s) & -\lambda_{14}(s) \\ -\lambda_{21}(s) & \lambda_{21}(s) + \lambda_{24}(s) & 0 & -\lambda_{24}(s) \\ -\lambda_{31}(s) & 0 & \lambda_{31}(s) + \lambda_{34}(s) & -\lambda_{34}(s) \\ -\lambda_{41}(s) & -\lambda_{42}(s) & -\lambda_{43}(s) & \lambda_{41}(s) + \lambda_{42}(s) + \lambda_{43}(s) \end{bmatrix}
 \end{aligned} \tag{32}$$

Here, we also outline the aforementioned matrix of an undirected dynamic graph as

$$L(s) = BD(s)B^T = D(s) - A(s), \tag{33}$$

where $D(s) \in \mathbb{C}^{m \times m}$ is the dynamic degree matrix formed by the dynamic degree of the m edges, $B \in \mathbb{R}^{n \times m}$ is the static incident matrix that captures the orientations of the edges, and $A(s) \in \mathbb{C}^{n \times m}$ is the dynamic adjacency matrix.

We define the dynamic Laplacian matrix through the implementation of the the dynamic degree and dynamic adjacency matrices which distinguish between incoming and outgoing conventions. Examples include: $L^{in}(s) = D^{in}(s) - A^{in}(s)$ and $L^{out}(s) = D^{out}(s) - A^{out}(s)$. With these definitions, $L(s) = L^{in}(s) + L^{out}(s)$. Particular caution must be taken in noting that while $L = BB^T$ in the static case, $L(s) \neq B(s)B(s)^T$, $L^{in}(s) \neq B^{in}(s)B^{in}(s)^T$ and $L^{out}(s) \neq B^{out}(s)B^{out}(s)^T$. To overcome this, in the sequel, we will use $L^{out}(s)$, where the outgoing, dynamic Laplacian matrix can be defined using the incident matrices as follow:

$$L^{out}(s) = D^{out}(s) - A^{out}(s), \quad (34)$$

where $D^{out}(s)$ and $A^{out}(s)$ are the dynamic-outgoing degree and adjacency matrices, respectively. These matrices can be defined in a static case using the incident matrices as $D^{out} = B^{out} B^{out^T}$ and $A^{out} = -B^{out} B^{in^T}$. Thus, the outgoing, dynamic degree and adjacency matrices can be defined as:

$$\begin{aligned} D^{out}(s) &= B_D^{out}(s) B_S^{out^T}, \\ A^{out}(s) &= -B_D^{out}(s) B_S^{in^T}. \end{aligned} \quad (35)$$

By combining (Eqs. (34) and (35)), the outgoing, dynamic Laplacian matrix $L^{out}(s)$ can be defined as

$$\begin{aligned} L^{out}(s) &= B_D^{out}(s) B_S^{out^T} + B_D^{out}(s) B_S^{in^T} \\ &= B_D^{out}(s) (B_S^{out^T} + B_S^{in^T}) = B_D^{out}(s) B_S^T. \end{aligned} \quad (36)$$

A similar definition can be given for $L^{in}(s)$.

To illustrate, regarding the example noted in **Figure 8**, the associated dynamic degree, adjacency, and Laplacian matrices are given by (Eq. (32)). Notice that $\lambda_{ij}(s)$ depicted in (Eq. (32)) are transfer functions that describe the interconnections (edges) between the nodes (**Figure 9**).

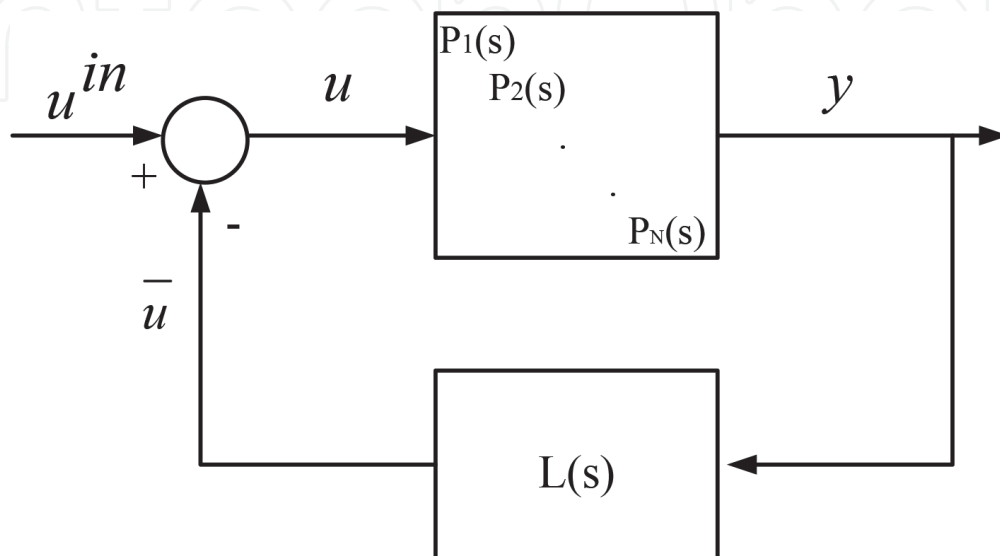


Figure 9.
Block diagram of the dynamic graph.

Now, the dynamic of each node can be represented in time domain for $i = 1, 2, \dots, N = 4$ by

$$\begin{aligned} \dot{x}_i(t) &= A_i x_i(t) + B_i \left[u_i^{in}(t) - \sum_{j \in \mathcal{N}_i} \lambda_{ij} \left(\frac{d}{dt} \right) * [x_i(t) - x_j(t)] \right] \\ y_i(t) &= C_i x_i(t), \end{aligned} \quad (37)$$

where, $x_i(t)$, $u_i^{in}(t)$, and $y_i(t)$ are the state, input, and output of the node i , respectively, $\lambda_{ij}(s)|_{s=\frac{d}{dt}}$ is the transfer function of the edge between the nodes i and j , and \mathcal{N}_i is the neighboring set to the node i .

The overall system of the dynamic graph 8 can be written in frequency domain as

$$P(s)^{-1}Y(s) = -L^{out}(s)Y(s) + u^{in}(s), \quad (38)$$

where, $P(s) = -diag(P_1(s), P_2(s), \dots, P_N(s))$; $P_i(s) = C_i(sI_n - A_i)^{-1}B_i$, $u^{in}(s)$, $Y(s)$ are the Laplace transform of the input and output variables, and $L^{out}(s)$ is the outgoing-dynamic Laplacian matrix of the dynamic graph.

The block diagram for the overall system (38) can be depicted as shown in 9.

The dynamic graphs presented here are governed by dynamic consensus protocols as discussed in the previous sections.

4. Conclusion

This Chapter studied a generalization of consensus network problems whereby the network edges' weights are no longer modeled as static gains. Instead, they are represented as dynamic systems coupling the nodes. We call such networks dynamic consensus networks because, under some conditions, all node variables converge to a common value called a consensus. We presented examples of how dynamic graphs can arise in applications. Detailed studies of modeling thermal processes in buildings as directed dynamic graphs were presented. Motivated by these examples, a framework was proposed for dynamic graphs and dynamic consensus networks. This framework introduced the idea of dynamic degree, adjacency, incident, and Laplacian matrices in a way that naturally extends these concepts from the static case. The dynamic consensus networks addressed herein considered various dynamics of nodes and interconnection topology, including (1) directed dynamic networks with integrator nodes and real-rational transfer function edges; (2) undirected dynamic networks with integrator nodes and strictly-positive-real transfer function edges; and (3) undirected dynamic networks with identical linear time-invariant nodes and dynamic edges. We used the established aspects and properties of the defined dynamic graph theory in conjunction with the behavioral approach to developing a controllability-analysis methodology for dynamic networks.

Acknowledgements

The authors would like to thank Professor Kevin L. Moore and Professor Tyrone L. Vincent at the Colorado School of Mines for their help and advice.

IntechOpen

IntechOpen

Author details

Fadel M. Lashhab

Department of Electrical Engineering and Computer Science, College of Engineering and Architecture, Howard University, Washington DC, USA

*Address all correspondence to: fadel.lashhab@howard.edu

IntechOpen

© 2021 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on automatic control*, vol. 49, no. 9, pp. 1520–1533, 2004.
- [2] M. Ji and M. Egerstedt, "A graph-theoretic characterization of controllability for multi-agent systems," *In IEE American Control Conference, ACC'07*, pp. 4588–4593, 2007.
- [3] A. Rahmani and M. Mesbahi, "On the controlled agreement problem," *In IEEE American Control Conference*, pp. 6, 2006.
- [4] Amirreza Rahmani, Meng Ji, Mehran Mesbahi, and Magnus Egerstedt, "Controllability of multi-agent systems from a graph-theoretic perspective," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 162–186, 2009.
- [5] M. Zamani and H. Lin, "Structural controllability of multi-agent systems," *In IEEE American Control Conference, ACC'09*, pp. 5743–5748, 2009.
- [6] Tadeusz Kaczorek "Minimum energy control of positive continuous-time linear systems with bounded inputs," *International Journal of Applied Mathematics and Computer Science*, vol. 23, no. 4, pp. 725–730, 2013.
- [7] Jerzy Klamka "Controllability and minimum energy control problem of fractional discrete-time systems," *In New Trends in Nanotechnology and Fractional Calculus Applications*, pp. 503–509, Springer, 2010.
- [8] X. Liu, H. Lin, and B. Chen, "A graph-theoretic characterization of structural controllability for the multi-agent system with switching topology," *in Proceedings of the 48th IEEE Conference on Decision and Control, held jointly with the 28th Chinese Control Conference*, pp. 7012–7017, IEEE, 2009.
- [9] H. Tanner, "On the controllability of nearest neighbor interconnections," *in Proceedings of the 43rd IEEE Conference on Decision and Control*, vol. 3, pp. 2467–2472, IEEE, 2004.
- [10] M. Ji, A. Muhammad, and M. Egerstedt, "Leader-based multi-agent coordination: Controllability and optimal control," *in Proceedings of 2006 American Control Conference*, pp. 1358–1363, IEEE, 2006.
- [11] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt, "Controllability of multi-agent systems from a graph-theoretic perspective," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 162–186, 2009.
- [12] A. Ramakrishna and N. Viswanadham, "Decentralized control of interconnected dynamical systems," *IEEE Transactions on Automatic Control*, vol. 27, no. 1, pp. 159–164, 1982.
- [13] Vikas Kumar Mishra, Ivan Markovskiy, and Ben Grossmann, "Data-Driven Tests for Controllability," *IEEE Control Systems Letters*, vol. 5, no. 2, pp. 517–522, 2020.
- [14] He, Wangli and Xu, Bin and Han, Qing-Long and Qian, Feng, "Adaptive consensus control of linear multiagent systems with dynamic event-triggered strategies," *IEEE transactions on cybernetics*, vol. 50, no. 7, pp. 2996–3008, 2019.
- [15] Ren, Wei, "Multi-vehicle consensus with a time-varying reference state," *Systems & Control Letters*, vol. 56, no. 7–8, pp. 474–483, 2007.
- [16] K. L. Moore, T. L. Vincent, F. Lashhab, and C. Liu, "Dynamic consensus networks with application to the analysis of building thermal

processes,” *Proceedings of the 18th IFAC World Congress*, vol. 18, no. 1, pp. 3078–3083, 2011.

[17] Fadel Lashhab, “Estimation of Dynamic Laplacian Eigenvalues in Dynamic Consensus Networks,” *IEEE International IoT, Electronics and Mechatronics Conference (IEMTRONICS)*, pp. 1–10, 2020.

[18] Fadel Lashhab, “Dynamic Consensus Networks: Dynamic Graph Definitions and Controllability Analysis Using the Behavioral Approach,” *IEEE International IoT, Electronics and Mechatronics Conference (IEMTRONICS)*, pp. 1–9, 2020.

[19] Fadel Lashhab, Kevin Moore, Tyrone Vincent, Deyuan Meng, and Khalid Kuwairi, “Robust H_∞ controller design for dynamic consensus networks”, *International Journal of Control*, vol. 91, no. 8, pp. 1906–1919, 2018.

[20] D.B. Crawley, L.K. Lawrie, F.C. Winkelmann, WF Buhl, Y.J. Huang, C. O. Pedersen, R.K. Strand, R.J. Liesen, D. E. Fisher, M.J. Witte, et al. EnergyPlus: creating a new-generation building energy simulation program. *Energy & Buildings*, 33(4):319–331, 2001.

[21] J.E. Braun and N. Chaturvedi. An inverse gray-box model for transient building load prediction. *HVAC&R Research*, 8(1):73–100, 2002.

[22] K. Lee and J.E. Braun. Model-based demand-limiting control of building thermal mass. *Building and Environment*, 43:1633–1646, 2008.

[23] X. Xu and S. Wang. Optimal simplified thermal models of building envelope b d on frequency domain regression using genetic algorithm. *Energy and Buildings*, 39: 525–536, 2007.

[24] K.-K. Oh, F. Lashhab, K. L. Moore, T. L. Vincent, and H.-S. Ahn, “Consensus of positive real systems cascaded with a single integrator,” *International Journal of Robust and Nonlinear Control*, vol. 25, no. 3, pp. 418–429, 2015.