Spring 4-30-2020

# Target Control of Networked Systems

Isaac S. Klickstein
*University of New Mexico*

**Isaac Klickstein**

*Candidate*

**Mechanical Engineering**

*Department*

This dissertation is approved, and it is acceptable in quality and form for publication:

*Approved by the Dissertation Committee:*

**Professor Francesco Sorrentino**

*Chair*

**Professor Meeko Oishi**

*Member*

**Professor Marios Pattichis**

*Member*

**Professor John Russell**

*Member*

**Professor Chris Hall**

*Member*

# TARGET CONTROL OF NETWORKED SYSTEMS

**BY**

**ISAAC KLICKSTEIN**

B.S., Mechanical Engineering, University of New Mexico, 2015

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

**Doctor of Philosophy**
**Engineering**

The University of New Mexico
Albuquerque, New Mexico

**May, 2020**

# DEDICATION

*It is easier to avoid the snares of love than to escape once*
*you are in that net whose cords and knots are strong;*
*but even so, enmeshed, entangled, you can still get out unless, poor fool,*
*you stand in your own way.*

***Lucretius***

# ACKNOWLEDGEMENTS

I would like to express thanks to my advisor, Professor Francesco Sorrentino, for supporting my research and giving me the opportuntity to develop my own topics which have lead to a number of publications. With his guidance, I was able to navigate and eventually become comfortable in academia, now having published over a dozen peer reviewed articles with him.

I would like to thank the rest of my thesis committee as well; Professors John Russell, Meeko Oishi, Marios Pattichis, and Chris Hall for taking the time to help me through the final stages of my Ph.D.

Also, I must thank Afroza Shirin for putting up with me as an office-mate for around four years. Through our close proximity, we taught each other an inordinate amount of skills.

My siblings, Jacob and Naomi, who have persevered with convincing looks of interest through my most monotonous discussions of the technical details of my research deserve acknoledgement as well.

Most importantly, my parents, who never gave up on me despite my obstinate determination to not go to college. It is because of them that this thesis exists.

# TARGET CONTROL OF NETWORKED SYSTEMS

by

**Isaac Klickstein**

**B.S., Mechanical Engineering, University of New Mexico, 2015**
**Ph.D., Engineering, University of New Mexico, 2020**

## Abstract

The control of complex networks is an emerging field yet it has already garnered interest from across the scientific disciplines, from robotics to sociology. It has quickly been noticed that many of the classical techniques from controls engineering, while applicable, are not as illuminating as they were for single systems of relatively small dimension. Instead, properties borrowed from graph theory provide equivalent but more practical conditions to guarantee controllability, reachability, observability, and other typical properties of interest to the controls engineer when dealing with large networked systems. This manuscript covers three topics investigated in detail by the author: (i) the role of the choice of target nodes (system outputs) on the control effort, (ii) creating and analyzing graphs with symmetry, and (iii) the relationship between graph structural properties and control effort.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

The study of the control of complex networks is an emerging field that combines the important results from graph theory, control theory, optimal control, optimization, and operations research. The integration of these fields is imperative in providing meaningful answers to the important questions such as controllability, efficiency, resource allocation, and many others. Results have been applied to many fields such as in synchronization in power grids, reprogramming cell populations, and flocking in networks of multi-agent systems.

Some of the major results developed in the field concern (i) the number of independent control inputs required to guarantee controllability, (ii) the control effort, or energy, required to perform a particular control action, and (iii) the most efficient allocation of resources, or control inputs, in order to perform a desired control action. Each of these questions does not have a simple answer, and many assumptions must be made about the underlying network. In fact, due to this important dependence on properties of the network which are sometimes not made clear, contradictory and/or misleading results have lead to unfortunate widespread confusion. In the following chapters we attempt to qualify many of the existing results and create a taxonomy of network control as well as to introduce new concepts which counter some of the more pessimistic existing results.

This dissertation covers the following publications, pre-prints, presentations, and ongoing projects:

1. Isaac Klickstein, Afroza Shirin, and Francesco Sorrentino. “Energy scaling of targeted optimal control of complex networks”. In: *Nature Communications* 8 (Apr. 2017), p. 15145 [1]

2. Afroza Shirin, Isaac Klickstein, and Francesco Sorrentino. “Optimal control of complex networks: Balancing accuracy and energy of the control action”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 27.4 (Apr. 2017), p. 041103 [2]

3. Isaac Klickstein, Afroza Shirin, and Francesco Sorrentino. “Locally Optimal Control of Complex Networks”. In: *Physical Review Letters* 119.26 (Dec. 2017), p. 268301 [3]

4. Isaac Klickstein et al. “Energy Scaling with Control Distance in Complex Networks”. In: *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, May 2018, pp. 1–5 [4]

5. Isaac Samuel Klickstein and Francesco Sorrentino. “Control Distance and Energy Scaling of Complex Networks”. In: *IEEE Transactions on Network Science and Engineering* (2018) [5]

6. Isaac Klickstein and Francesco Sorrentino. “Control Energy of Lattice Graphs”. In: *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, Dec. 2018, pp. 6132–6138 [6]

7. Ishan Kafle et al. “Optimal control of networks in the presence of attackers and defenders”. In: *Chaos* 28.5 (2018) [7]

8. Isaac Samuel Klickstein and Francesco Sorrentino. “Generating Graphs with Symmetry”. In: *IEEE Transactions on Network Science and Engineering* (2018) [8]

9. Isaac Klickstein and Francesco Sorrentino. “Generating symmetric graphs”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 28.12 (Dec. 2018), p. 121102 [9]

10. Isaac Klickstein “Generating Graphs with Symmetry” (Part of “Effects of Symmetries and Partitions on Dynamics in Networks”). Presented at SIAM Conference on Applications of Dynamical Systems (DS19).

11. Afroza Shirin et al. “Prediction of Optimal Drug Schedules for Controlling Autophagy”. In: *Scientific Reports* 9.1 (Dec. 2019), p. 1428 [10]

12. Afroza Shirin et al. “Optimal regulation of blood glucose level in Type I diabetes using insulin and glucagon”. In: *PLOS ONE* 14.3 (Mar. 2019). Ed. by Abhyudai Singh, e0213665 [11]

13. Isaac Klickstein, Louis Pecora, and Francesco Sorrentino. “Symmetry induced group consensus”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29.7 (July 2019), p. 073101 [12]

14. Afroza Shirin, Isaac S. Klickstein, and Francesco Sorrentino. “Stability analysis of reservoir computers dynamics via Lyapunov functions”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29.10 (Oct. 2019), p. 103147 [13]

15. Isaac Klickstein and Francesco Sorrentino. “The controllability Gramian of lattice graphs”. In: *Automatica* 114 (Apr. 2020), p. 108833 [14]

16. Fabio Della Rossa et al. “Symmetries and Cluster Synchronization in Multilayer Networks”. In: *Nature Communications* ACCEPTED (2020) [15]

17. Current Project: Isaac Klickstein and Francesco Sorrentino. “Controlling Network Ensembles” . arxiv:

18. Current Project: Isaac Klickstein and Francesco Sorrentino. “Structure Based Actuator Placement in Complex Networks”. arXiv:

19. Current Project: Isaac Klickstein, David Phillips, Lou Pecora, and Francesco Sorrentino. “Approximate Equitable Partitions.”

This manuscript is split into four major chapters with conclusions made in the last chapter.

Chapter 2 presents the notation, definitions, and theorems which are used throughout the rest of the dissertation before turning to the main topic of *controllability* as it pertains to *networks*. The main goal of chapter 2 is to provide a taxonomy of network control to alleviate some of the confusion that persists in the field. The classification scheme developed handles the myriad of results for different types of networks and different controllers. Optimal control is also briefly described and, in particular, Pontryagin’s minimum principle. Pontryagin’s minimum principle is used to derive the solution of the minimum energy control problem for linear systems which forms the basis of the results throughout this dissertation. Finally, numerical methods for optimal control are presented and the results in applications to regulating autophagy [10] and regulating blood glucose in diabetics [11] are discussed.

Chapter 3 describes target controllability in depth. Results concerning the scaling of control energy [1], the control energy trade-off with accuracy [2], and most recently, the control energy of network ensembles are presented in turn. Results on the controllability Gramian are used to define a geometry based controller for nonlinear systems [3] and results on target controllability are used to develop counter measures in the precense of attackers [7].

Chapter 4 covers our results concerning symmetries in graphs. Symmetries have been shown to make more difficult the control of complex networks so understanding how they may emerge in networks is important. In particular, our method to generate graphs with symmetries [8, 9] is described in detail. An extension to multilayer networks is described [15] as well. On the other hand, if the control goal is synchronization, symmetries actually aid in this goal, and so creating networks with symmetries is beneficial. The role symmetry plays in consensus problems [12] is also presented, in particular, how one can determine if consensus occurs regardless of the stability of the dynamical system. Finally, current work concerning approximate equitable partitions that incorporate edge weights is discussed.

Chapter 5 discusses our numerical and analytical results concerning the relationship between control energy and graph properties. Initially, we investigate our ability to make predictions about control energy using simple graph models numerically [4]. Analytic results are then discussed [5, 6] and the control energy of generic lattice graphs is presented using a discrete time Fourier transform to solve for the controllability Gramian in terms of its independent elements. Finally, current work uses the above analytic results to derive a heuristic for the driver node selection problem, a type of resource allocation problem, which has been shown to be NP-hard.

# Chapter 2

# Background

This chapter presents the preliminary definitions, theorems, and notation needed for the remainder of this manuscript. Previous work on controlling complex networks is discussed as well to place the research contained in the following chapters in context within the larger field of network control. Special care is taken to classify each result in terms of the type of graph being analyzed and the type of control applied. Misunderstanding of the proper context of some results has lead to confusion which this chapter attempts to alleviate.

## 2.1 General Notation

Unless otherwise specified, normal font lower case letters, $a, b, \theta, \omega$, are scalars, lower case bold face letters, $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\theta}, \boldsymbol{\omega}$, are vectors, and upper case letters, $A, B, \Theta, \Omega$, are matrices. The set of real numbers is denoted $\mathbb{R}$, the set of complex numbers is denoted $\mathbb{C}$, and the set of integers is denoted $\mathbb{Z}$. The space of real (complex) vectors of length $n$ is denoted $\mathbb{R}^n$ ($\mathbb{C}^n$) and matrices of dimensions $n \times m$ is denoted $\mathbb{R}^{n\times m}$ ($\mathbb{C}^{n\times m}$). The identity matrix of dimension $n$ is denoted $I_n$ and the matrix of all zeros of dimension $m \times n$ is $O_{m,n}$. The vector of all ones of length $n$ is $\mathbf{1}_n$ and of all zeros $\mathbf{0}_n$. Sets are denoted by calligraphic font such as $\mathcal{C}$ or $\mathcal{S}$.

## 2.2 Graph Theory and Networks

A graph consists of a set, $\mathcal{V}$, and a binary relation on that set, $\mathcal{E} : \mathcal{V} \times \mathcal{V} \mapsto \{0, 1\}$, denoted $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The set $\mathcal{V}$ consists of $n = |\mathcal{V}|$ vertices and the set of edges can alternatively be thought of as a subset of all pairs of nodes $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. An edge $(v_j, v_k) \in \mathcal{E}$ implies that node $v_k$ receives information from node $v_j$, so it may be read as 'from node $v_j$ to node $v_k$.' The nature of this connection varies by the underlying system which the network is meant to represent. For example, in a social media network edges represent 'following' or 'friendship' so that user $v_k$ can see what user $v_j$ posts. For a power grid, edges represent whether the phase of a bus $v_k$ is directly affected by the phase of $v_j$ according to the swing equations. In a foodweb, an edge represents whether the population of specics $v_k$ is affected by the population of $v_j$, whether through predation or competition. The following definitions are used to classify types of graphs.

**Definition 2.2.1** (Directedness). *A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is said to be* undirected *if,*

$$(v_j, v_k) \in \mathcal{E} \Leftrightarrow (v_k, v_j) \in \mathcal{E}$$

*If a graph is undirected, often one counts each* undirected edge *once, rather than counting $(v_j, v_k)$ and $(v_k, v_j)$ separately. If a graph is not undirected then it is* directed.

A graph is assumed to be directed unless otherwise stated explicitly. An undirected graph can be treated as a directed graph by duplicating each edge and assigning opposite directions to the copies. If $\mathcal{G}$ is directed, one can create an *induced undirected graph* by augmenting the set of edges with $(v_k, v_j)$ for each $(v_j, v_k) \in \mathcal{E}$.

**Definition 2.2.2** (Bipartite Graphs). *A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is bipartite if the set of nodes can be bisected into two parts, $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$, $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$, such that every edge $(v_j, v_k) \in \mathcal{E}$ has $v_j \in \mathcal{V}_1$ and $v_k \in \mathcal{V}_2$ (or vice versa).*

**Definition 2.2.3** (Neighbors)**.** *The* neighbors *of a node $v_j \in \mathcal{V}$ is a set of nodes,*

$$\mathcal{N}_j = \{v_k \in \mathcal{V} | (v_k, v_j) \in \mathcal{E}\}$$

Specifically, Def. 2.2.3 is the set of nodes which send information to node $v_j$. Alternatively the set of nodes to which node $v_j$ sends its information, denoted $\bar{\mathcal{N}_j} = \{v_k \in \mathcal{V} | (v_j, v_k) \in \mathcal{E}\}$, may be useful. If the graph is undirected according to Def. 2.2.1, then $\mathcal{N}_j = \bar{\mathcal{N}_j}$ for every $v_j \in \mathcal{V}$.

**Definition 2.2.4** (Degree)**.** *The in-degree of a node (sometimes referred to as simply the degree) is the number of neighbors a node has.*

$$\kappa_j = |\mathcal{N}_j|, \quad \forall v_j \in \mathcal{V}$$

*Similarly, the out-degree of a node is the number of outgoing neighbors a node has.*

$$\bar{\kappa}_j = |\bar{\mathcal{N}_j}|, \quad \forall v_j \in \mathcal{V}$$

*If a graph is undirected, then $\kappa_j = \bar{\kappa}_j$.*

**Definition 2.2.5** (Degree Distribution)**.** *The most common way graphs are classified is by the distribution of their degrees (Def. 2.2.4), defined as,*

$$P(\kappa) = \frac{\textit{No. of nodes with degree } \kappa}{|\mathcal{V}|}$$

Some common degree distributions are discussed in Subsection 2.2.1.

**Definition 2.2.6** (Weighted Graphs)**.** *A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is said to be* weighted *if there exists a function $w : \mathcal{E} \mapsto \mathbb{R}$ where $w(v_j, v_k) = w_{j,k}$ is the weight of edge $(v_j, v_k) \in \mathcal{E}$. If a graph is not weighted then it is* unweighted.

Some of the types of control defined later in this chapter are concerned with graph structure, where weights do not play a role, while other types of control requires weights to be assigned to each edge. An unweighted graph can be treated as a weighted graph, typically by defining $w_{j,k} = 1$ for every edge $(v_j, v_k) \in \mathcal{E}$.

**Definition 2.2.7** (Strength)**.** *The in-strength of a node (sometimes referred to as simply the strength), denoted $\sigma_j$, is the sum of the incoming weights.*

$$\sigma_j = \sum_{v_k \in \mathcal{N}_j} w_{j,k}$$

*Similarly, the out-strength of a node is the sum of the weights of outgoing neighbors a vertex has.*

$$\bar{\sigma}_j = \sum_{v_k \in \bar{\mathcal{N}}_j} w_{j,k}$$

*If a graph is undirected, then $\sigma_j = \bar{\sigma}_j$ for all vertices $v_j \in \mathcal{V}$.*

If a graph is unweighted, then a vertex's degree (Def. 2.2.4) and strength (Def. 2.2.7) are usually considered equal.

**Definition 2.2.8** (Loops)**.** *A loop is an edge with the same node as its tail and head, $(v_j, v_j) \in \mathcal{E}$.*

Loops play a substantially different role in the dynamics to be defined on a network in Subsection 2.2.3 and so we often treat them separately than the other edges. In fact, the inclusion or exclusion of loops is critical to understanding many results on the control of complex networks, especially structural ones, and so they will be treated with care in the following sections.

**Definition 2.2.9** (Subgraphs)**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph. A subgraph, $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ is a graph such that $\mathcal{V}' \subseteq \mathcal{V}$ and each edge $(v_j, v_k) \in \mathcal{E}'$ must have $v_j, v_k \in \mathcal{V}'$ and $(v_j, v_k) \in \mathcal{E}$.*

**Definition 2.2.10** (Adjacency Matrix)**.** *A graph can be alternatively represented as a square $n - by - n$ matrix, $A$, whose entries are,*

$$A_{j,k} = \begin{cases} w_{k,j}, & \textit{if } (v_k, v_j) \in \mathcal{E} \\ 0 & \textit{otherwise} \end{cases}$$

*If the graph is unweighted then often $w_{k,j} = 1$ for each edged $(v_k, v_j) \in \mathcal{E}$.*

Note the swap in the order of the index from the weights which are written $w_{k,j}$ (from $v_k$ to $v_j$) as compared to $A_{j,k}$ (to $v_j$ from $v_k$). This swap is necessary as the role of the adjacency matrix is different than that of the graph. If each node is assigned a value, $\boldsymbol{x}_j$, for all $v_j \in \mathcal{V}$, then the operation $A\boldsymbol{x}$ describes the incoming signal witnessed by each node. The inclusion of loops, treated separately from the graph, in the adjacency matrix may be done with a function of the nodes, $\ell : \mathcal{V} \mapsto \mathbb{R}$ so that $A_{j,j} = \ell(v_j) = \ell_j$ for $j = 1, \ldots, n$.

**Definition 2.2.11** (Paths). *A path from node $v_j$ to node $v_k$ in a graph is a sequence of edges,*

$$(v_j, v_{\ell_1}), (v_{\ell_1}, v_{\ell_2}), \ldots, (v_{\ell_s}, v_k)$$

*such that the first edge starts at node $v_j$ and the last ends at $v_k$. The length of a path is the number of edges it traverses.*

By the *distance* between two nodes we mean the length of the shortest path between two nodes. There may be pairs of nodes for which no path exists.

**Definition 2.2.12** (Connectedness). *A graph is* connected *if there exists a path between all pairs of nodes of its induced undirected graph (see the discussion just after Def. 2.2.1). A graph is* strongly connected *if there exists a path between all pairs of nodes without ignoring edge directionality.*

A graph can be decomposed into subgraphs representing its strongly connected components (SCCs) which is an important concept in the control of complex networks. A root SCC is an SCC of a graph with no incoming edges from any other SCC. The concept of a graph matching, which is fundamental to the control of complex networks, is presented here.

**Definition 2.2.13** (Matching). *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an* undirected *graph. A matching is a subset of the edges, $\mathcal{M} \subseteq \mathcal{E}$ such that no two edges in $\mathcal{M}$ share a node.*

A maximal matching is a matching such that, by the addition of any other edge, the set is no longer a matching. The maximum matching is the maximal matching $\mathcal{M}$ of maximum cardinality over all matchings. If the graph $\mathcal{G}$ is *bipartite* (see Def. 2.2.2) then one can find the maximum matching in polynomial time using the Hopcroft-Karp algorithm [16].

### 2.2.1 Types of Graphs

The connectivity pattern, or topology, of real systems is quite varied [17]. The most common method to characterize types of graphs is by their *degree distribution* (see Def. 2.2.5). The first two graph models extensively studied in the literature were the *lattice graph* (especially in physics [18, 19]) and the *random graph* [20].

**Definition 2.2.14** (Lattice Graph)**.** *A lattice graph is a graph whose set of edges follow some pattern independent of the index of any particular node. For example, the ring lattice assigns the nodes positions around a circle from $v_0$ to $v_{n-1}$, and consists of edges $(v_j, v_{j\pm k}) \in \mathcal{E}$ where the arithmetic is done modulo $n$, for $k = 1, \ldots, \frac{d}{2}$ for $d < n$.*

The control of lattice graphs makes up the basis of an extensive study in Chapter 5.

**Definition 2.2.15** (Erdős-Rényi Graph or Random Graph [20])**.** *A random graph of $n$ nodes, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, has edges $(v_j, v_k) \in \mathcal{E}$ with probability $p$.*

Many properties of random graphs have been studied in detail, for instance, the degree distribution of the random graph is binomial, which, in the limit of large $n$, becomes Poisson. While this model can be used to represent physical systems which have a Poisson degree distribution, for many systems of interest, this assumption does not hold.

In the late 1990s, a renewed interest in applied graph theory emerged driven by the increased availability of large datasets. These datasets did not exhibit the same properties of either lattices or random graphs and so new models had to be created to accurately represent them. Two concepts are required to demonstrate why the previous models are inadequate in many situations: average distance and average clustering.

**Definition 2.2.16** (Clustering Coefficient [21])**.** *The clustering coefficient of a node $v_j$ is the fraction of common neighbors shared among the neighbors of a node over the total possible number of shared neighbors.. That is, using the notation defined in this chapter,*

$$c_j = \frac{\sum_{v_k \in \mathcal{N}_j} |\mathcal{N}_k \cup \mathcal{N}_j|}{\kappa_j(\kappa_j - 1)}$$

*If the graph is undirected, an additional multiplicative factor of* 2 *is included so as to not count undirected edges twice.*

As defined previously (see the discussion after Def. 2.2.11), the distance between two nodes is the length of the shortest path between them. For the two graphs described so far, lattices (see Def. 2.2.14) typically have high clustering and long average distance, while Erdős-Rényi graphs have small clustering coefficients but short average distances.

In [21], an alternative random graph model was introduced which attempted to marry two concepts witnessed in datasets available at the time: a high degree of clustering like lattices but short average distances like random graphs. The method was to start with a lattice as defined in Def. 2.2.14 and to randomly *rewire* edges with some probability $p$. If $p = 0$, the original lattice is preserved, while if $p = 1$, the lattice becomes a random graph. Let $\langle c(p) \rangle$ and $\langle d(p) \rangle$ be the average clustering and average distance, respectively. It was shown that there exists a range of $p$ such that $\langle c(p) \rangle$ is large while $\langle d(p) \rangle$ is small during the transition from lattice to random graph. A small numerical example demonstrates this region in Fig. 2.1. By choosing $p$ properly, one can construct a graph which has both short average distance and high clustering.

In the seminal paper [22] that helped create the field of complex network analysis separate and apart from classic graph theory, it was shown many real systems have

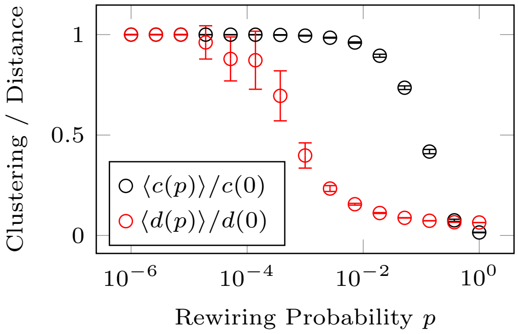


**Figure 2.1:** The average clustering and the average distance for Watts-Strogatz graphs for different values of rewiring probability $p$. It is clear there is a range of $p$ when $\langle c(p)\rangle$ remains large, i.e., high clustering, while $\langle d(p)\rangle$ shrinks, i.e., short average distance.

*power-law* degree distributions.

**Definition 2.2.17** (Scale-Free Graphs)**.** *A scale-free graph is one which has a power-law degree distribution.*

$$P(\kappa) \sim \kappa^{-\gamma}$$

*An example of a scale free graph is shown in Fig. 2.2.*

Originally [22], the supposed prevalence of scale-free graphs in real systems was attributed to two mechanisms, growth and preferential attachment. The resulting Barabási-Albert model starts with a seed graph with $n_0$ nodes. The growth arises by adding additional nodes to the graph at discrete time steps. Edges are added between these newly created nodes and already existing nodes with probability proportional to the nodes' degrees, the so-called preferential attachment. This result sparked an explosion in publications investigating every property of these scale-free graphs from their spectrum [23], the scaling properties of other models of growth and preferential attachment [24], diffusion processes [25], percolation [26], etc.

The omnipresence of scale-free graphs in data has been questioned though [27, 28] leading to heated debate into this year, such as [29] and the response [30]. Alternative degree distributions such as the exponential distribution with cut-off have been proposed to explain why it appeared so many datasets had power-law degree distribution.

One last model is presented which can be used to construct graphs of arbitrary

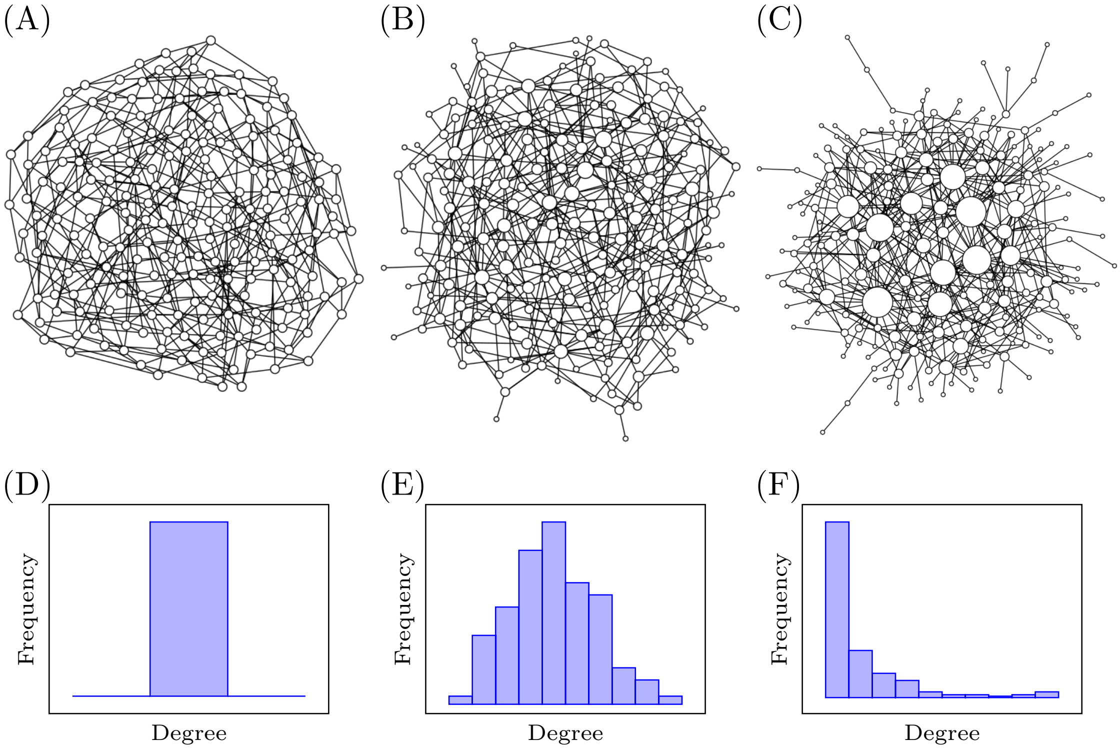


**Figure 2.2:** Graph models based on degree distribution where the nodes' sizes are proportional to their degree. (A) A sample $k$-regular graph, where every node has equal degree $k$. (B) An Erdős-Rényi graph as defined in Def. 2.2.15. (C) A scale-free graph as defined in Def. 2.2.17. (D) The degree distribution of a $k$-regular graph is $P(\kappa) = \delta(\kappa = k)$. (E) The degree distribution of an Erdős-Rényi graph is Poisson. (F) The degree distribution of a scale-free graph is a power law.

degree distribution. The configuration model [31, 32] takes as input $n$ disconnected nodes with degree sequence $\kappa_j \; \forall v_j \in \mathcal{V}$ (where $\sum_{v_j \in \mathcal{V}} \kappa_j$ is an even number) and returns an undirected graph. Each node, $v_j$, has $\kappa_j$ stubs to represent its future degree. While there exist nodes $v_j$ whose degree is less than $\kappa_j$, choose two stubs randomly and add an edge, ensuring (i) such an edge does not already exist and (ii) the two stubs are not attached to the same node to avoid multi-edges and self-loops, respectively. If the original degree sequence was chosen from a power law distribution, then the resulting graph is scale-free. The configuration model can also be used to construct directed graphs by creating two degree sequences, an in-degree sequence $\kappa_j$ and an out-degree sequence $\bar{\kappa}_j$ such that $\sum_{v_j \in \mathcal{V}} \kappa_j = \sum_{v_j \in \mathcal{V}} \bar{\kappa}_j$.

### 2.2.2 Graph Symmetry

A concept that will reappear many times throughout this dissertation is *graph symmetry* [33]. Before discussing it though, we present some additional definitions concerning partitioning and permuting graphs.

**Definition 2.2.18** (Partitions (Clustering))**.** *Let $\mathcal{S}$ be a set. A partition, $\mathcal{C}$, is a set of subsets of $\mathcal{S}$, defined as,*

$$\mathcal{C} = \left\{ \mathcal{C}_k \subseteq \mathcal{S} \;\middle|\; \mathcal{C}_j \cap \mathcal{C}_k = \emptyset \wedge \bigcup_{k=1}^{p} \mathcal{C}_k = \mathcal{S} \right\}$$

In words, every pair of subsets in the partition are *disjoint* and the union of all of the subsets in the partition is *complete*. Sometimes we will refer to partitions as clusterings which has the same definition. We also assume there is no subset $\mathcal{C}_k = \emptyset$ which ensures each partition has a unique representation as $\mathcal{C}$ up to a relabeling of the cluster indices. In particular, we examine partitions of nodes of a graph.

**Definition 2.2.19** (Cluster Neighbors)**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an unweighted graph and its nodes are clustered according to $\mathcal{C}$, a partition (see Def. 2.2.18). The cluster* *neighbor set of node $v_j \in \mathcal{V}$ in cluster $\mathcal{C}_k$ is defined as*

$$\mathcal{N}_{j,k} = \{v_\ell \in \mathcal{C}_k | (v_\ell, v_j) \in \mathcal{E}\}$$

*The cluster degree is denoted $\kappa_{j,k} = |\mathcal{N}_{j,k}|$.*

The cluster degrees for a clustering of the nodes can be succinctly written as vectors $\boldsymbol{\kappa}_j \in \mathbb{Z}^{|\mathcal{C}|}$.

**Definition 2.2.20** (Equitable Partition)**.** *An equitable partition is a partition such that the cluster degree vectors as defined in Def. 2.2.19 for every pair of nodes in the same cluster are equal, that is,*

$$\boldsymbol{\kappa}_j = \boldsymbol{\kappa}_k, \quad \forall v_j, v_k \in \mathcal{C}_\ell, \quad \forall \mathcal{C}_\ell \in \mathcal{C}.$$

Every graph has at least one equitable partition, specifically, the singleton clustering, $\mathcal{C}_k = \{v_k\}$, $\forall v_k \in \mathcal{V}$. Of more interest is the *minimum equitable partition*, that is the partition of smallest cardinality which is equitable. The minimum equitable partition of a graph can be found in polynomial time [34].

**Definition 2.2.21** (Permutations)**.** *A permutation is a bijection on the set of nodes, $\pi : \mathcal{V} \mapsto \mathcal{V}$.*

A permutation of a set of $n$ nodes can be represented as a square binary matrix $P$ where $P_{j,k} = 1$ if $\pi(v_j) = v_k$ and $P_{j,k} = 0$ otherwise. A permutation applied to a graph is denoted $\mathcal{G}^\pi = (\mathcal{V}, \mathcal{E}^\pi)$. Note that in general, $\mathcal{E}^\pi \neq \mathcal{E}$ but that $\mathcal{G}^\pi$ is *isomorphic* to $\mathcal{G}$ [35]. The action of applying a permutation to a graph can be represented in its adjacency matrix as $PAP^T = A^\pi$, where in general, $A \neq A^\pi$.

**Definition 2.2.22** (Symmetry)**.** *A permutation $\pi$ is a symmetry if $\mathcal{G} = \mathcal{G}^\pi$ which can be verified if,*

$$(v_j, v_k) \in \mathcal{E} \quad \Leftrightarrow \quad (\pi(v_j), \pi(v_k)) \in \mathcal{E}.$$

In terms of the adjacency matrix defined in Def. 2.2.10, if $P$ is the permutation matrix associated with a symmetry and the graph is unweighted, then $PAP^T = A$.

If $A$ is weighted, then even if $P$ represents a symmetry, it is not necessary that $PAP^T = A$. An alternative definition of *approximate equitable partition* is presented in Sec. 4.8 which modifies Def. 2.2.20 to include edge weights. We now present the definition of the automorphism group of a graph.

**Definition 2.2.23** (Automorphism Group)**.** *The set of all symmetries of a graph, along with function composition, is the automorphism group of a graph, defined as,*

$$Aut(\mathcal{G}) = \{\pi : \mathcal{V} \mapsto \mathcal{V} | \mathcal{G} = \mathcal{G}^\pi\}.$$

Computing the automorphism group of a graph is famously not a simple task [36, 37], as it is unknown if the problem is either P or NP.

Symmetries play an integral role in the controllability of complex networks which will be discussed in the following sections. Some original work on graph symmetries is presented in Chapter 4.

### 2.2.3 Networks

Generally, a network is a system described by a graph and a set of time-varying states assigned to each node in the graph which evolve according to a set of rules [38–41]. Examples of such systems include biological systems [42–44], neuronal networks [45–47], the power grid [48–50], social media [51–53], epidemics [25, 54, 55], transportation/routing [56, 57], and many many more [39, 58]. In this manuscript we focus on networks where each state assigned to a node evolves continuously in time (as opposed to discrete time networks, although most of the results in this manuscript can be extended to discrete time systems). The evolution of the state of each node is governed by system specific dynamics.

$$\frac{d}{dt}x_j = f_j(x_j, x_{k_1}, x_{k_2}, \dots, x_{k_{\kappa_j}}), \quad v_{k_\ell} \in \mathcal{N}_j, \tag{2.2.1}$$

where $f_j : \mathbb{R}^{\kappa_j+1} \mapsto \mathbb{R}$ describes how node $v_j$ is influenced by its neighbors. The dynamics of complex networks may not be as general as Eq. (2.2.1) suggests and instead the influence of each node $v_k$ on the dynamics of node $v_j$ may be pair-wise, modelled as a function of $x_k$ and $x_j$ alone,

$$\frac{d}{dt}x_j = \sum_{v_k \in \mathcal{N}_j} f_{j,k}(x_j, x_k) \tag{2.2.2}$$

Often, due to the pair-wise nature of the edges of a graph, in modeling dynamical networks, Eq. (2.2.2) is more applicable than Eq. (2.2.1). Biological systems are often described by Hill functions which describe the on-off switching of different regulatory enzymes [59], or power systems are described by the well known swing equations [60], or networks of coupled phase oscillators are described by phase difference coupling [61]. As each different application consists of different dynamical equations, for the majority of this manuscript we restrict ourselves to systems of linear equations, that is, each $f_{j,k}(x_j, x_k) = w_{k,j}x_k$ in Eq. (2.2.2). This simplification allows us to investigate the role of the graph topology, or its connectivity, on dynamical properties of the network, without losing generality to specific dynamics.

## 2.3 Control of Networks

The control of networks is a relatively young field and as such suffers from the growing pains of mixed terminology and a lack of standard notation. To make this section coherent, a categorization is imposed, (consisting of a mix of the most widely accepted definitions and some novel inventions). Before turning to the control of networks specifically, we present some of the classic results on the controllability of linear systems.

**Definition 2.3.1** (Linear System). *A linear system is defined as the triplet* $(A, B, C)$ *where* $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times n_d}$ *and* $C \in \mathbb{R}^{n_t \times n}$ *such that,*

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= A\boldsymbol{x}(t) + B\boldsymbol{u}(t) \\ \boldsymbol{y}(t) &= C\boldsymbol{x}(t) \end{aligned} \tag{2.3.1}$$

*where* $\boldsymbol{x} : \mathbb{R}^+ \mapsto \mathbb{R}^n$ *is the evolution of the states,* $\boldsymbol{u} : \mathbb{R}^+ \mapsto \mathbb{R}^{n_d}$ *is the* input *and*

$\boldsymbol{y} : \mathbb{R}^+ \mapsto \mathbb{R}^{n_t}$ *is the* output*.*

Along with Eq. (2.3.1), we require an *initial condition*, $\boldsymbol{x}(0)$, that describes the system completely at time $t = 0$. The theory of linear systems is extensive [62, 63]. If $C = I_n$, the $n$-by-$n$ identity matrix, then one may ignore the output and succinctly define the system as the pair $(A, B)$.

### 2.3.1 Classic Results on Control

Control of linear systems theory mainly concerns itself with the question 'what input $\boldsymbol{u}$ will achieve a desired output $\boldsymbol{y}(t_f)$ satisfying Eq. (2.3.1) at time $t_f$?' To answer this question one typically first asks if Eq. (2.3.1) is *controllable.*

**Definition 2.3.2** (Controllability [64] [65])**.** *A linear system $(A, B)$ (see Def. 2.3.1) is* controllable *if there exists an input $\boldsymbol{u}$, for every initial condition $\boldsymbol{x}(0)$, that is able to achieve any desired state $\boldsymbol{x}(t_f)$, for $t_f > 0$. Similarly, a linear system $(A, B, C)$ is* output controllable *if there exists an input $\boldsymbol{u}$, for every initial condition $\boldsymbol{x}(0)$, that is able to achieve any desired output $\boldsymbol{y}(t_f)$, for $t_f > 0$. If a system is not controllable (output controllable) then it is said to be* uncontrollable *(*output uncontrollable*).*

A fact that will arise often is that a system may both be uncontrollable and output controllable. A number of theorems exist with which one may test whether or not a system is controllable. Often the first test a student of control theory learns is the Kalman rank criterion.

**Theorem 2.3.1** (Kalman Rank Criterion [66])**.** *Define the Kalman matrix*

$$K = \left[B|AB|A^2B|\cdots|A^{n-1}B\right] \tag{2.3.2}$$

*The pair $(A, B)$ is controllable if and only if rank$(K) = n$. Similarly, the triplet $(A, B, C)$ is output controllable if and only if rank$(CK) = n_t$.*

If $n$ is small then one can construct $K$ using Eq. (2.3.2) explicitly, but for large $n$, applying Thm. 2.3.1 directly can become difficult due to numerical overflow when determining the rank with finite precision arithmetic. A common alternative test is the PBH criterion,

**Theorem 2.3.2** (PBH Criterion [67]). *A pair $(A, B)$ is uncontrollable if and only if there exists a left eigenvector of $A$, $\boldsymbol{v} \in \mathbb{C}^n$, such that,*

$$\boldsymbol{v}^T B = \boldsymbol{0}^T \quad \textit{and} \quad \boldsymbol{v}^T A = \lambda \boldsymbol{v}^T$$

*where $\lambda \in \mathbb{C}$ is the corresponding eigenvalue.*

While Thm. 2.3.2 does not require finding the rank of a poorly conditioned matrix, it does require intimate knowledge of the eigendecomposition of the matrix $A$. Depending on the structure of $A$ this may or may not be a reasonable request. A popular alternative to classic controllability in the field of complex networks is *structural controllability.*

**Definition 2.3.3** (Structural Controllability [68, 69]). *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph. Let $A$ be the adjacency matrix of $\mathcal{G}$ (see Def. 2.2.10) associated with weight function $w$ (see Def. 2.2.6). The pair $(A, B)$ is structurally controllable if $\forall \epsilon > 0$ there exists a pair $(A', B')$ (i.e., another weight function $w'$ defined on the same graph $\mathcal{G}$) that* is controllable *such that $||A - A'|| < \epsilon$ and $||B - B'|| < \epsilon$.*

In other words, if $\mathcal{G}$ is structurally controllable, then if a particular weight function $w$ leads to the system $(A, B)$ becoming uncontrollable, then there exists another pair, $(A', B')$, such that $||A - A'|| < \epsilon$ and $||B - B'|| < \epsilon$ that is controllable for any $\epsilon > 0$. This weaker definition of controllability is sometimes more applicable in the complex networks field [70] as long as one can assume that the edge weight function $w$ is not known precisely and that there are not relationships between weights. Structural controllability can also be proved using graph properties which avoids the numerical difficulties using Thms. 2.3.1 or 2.3.2.

**Definition 2.3.4** (Strong Structural Controllability [71]). *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with edge weight function $w$ and associated adjacency matrix $A$. The graph is strongly*

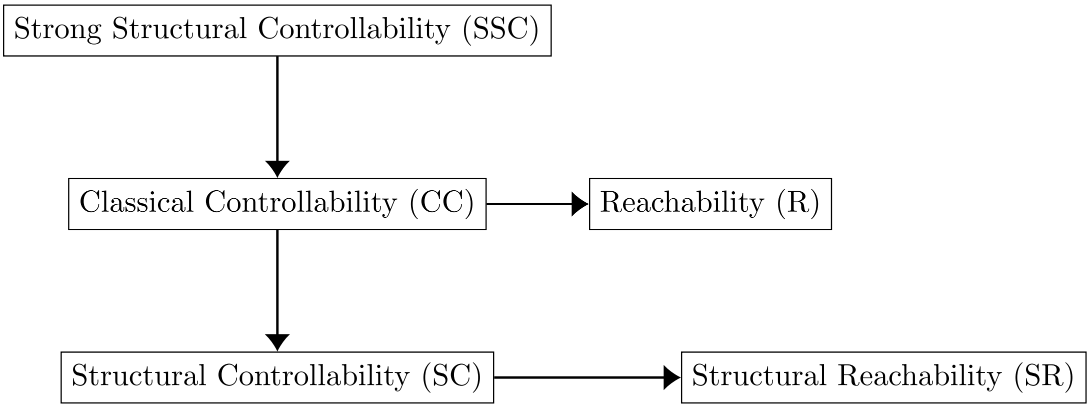

Strong Structural Controllability (SSC)
Classical Controllability (CC)
Reachability (R)
Structural Controllability (SC)
Structural Reachability (SR)

**Figure 2.3:** Implication relationships between the defined types of controllability.

*structurally controllable if the pair $(A, B)$ is controllable for all weight functions such that $w(v_j, v_k) \neq 0$.*

Strong structural controllability is an even stronger definition of controllability than classic controllability. One last definition before moving on concerns *reachability.*

**Definition 2.3.5** (Reachability [72])**.** *Let $(A, B)$ be a system with initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$ and final condition $\boldsymbol{x}(t_f) = \boldsymbol{x}_f$. The pair $(A, B)$ is reachable if there exists an input $\boldsymbol{u} : [0, t_f] \mapsto \mathbb{R}^{n_d}$ such that the state is transferred from $\boldsymbol{x}_0$ to $\boldsymbol{x}_f$.*

If a system is controllable, then it is also reachable for any pair of initial condition and final condition, while an uncontrollable system may still be reachable. Additional relationships between the types of controllability are outlined in Fig. 2.3. Understanding the proper type of control called for by an application is imperative, as requiring too strict a definition can lead to unnecessarily pessimistic conclusions.

### 2.3.2 Control of Complex Networks: The Basics

Due to the age of the field of controlling networks, conflicting definitions concerning control of networks exist [73, 74]. As such, some of the confusion present in the field is mitigated by defining concepts with what we believe are the most accepted and/or the most intuitive definitions and notation. We make the following classification based on the type of controllability. First, we must classify the type of control input matrix $B$ with which we are interested [73].

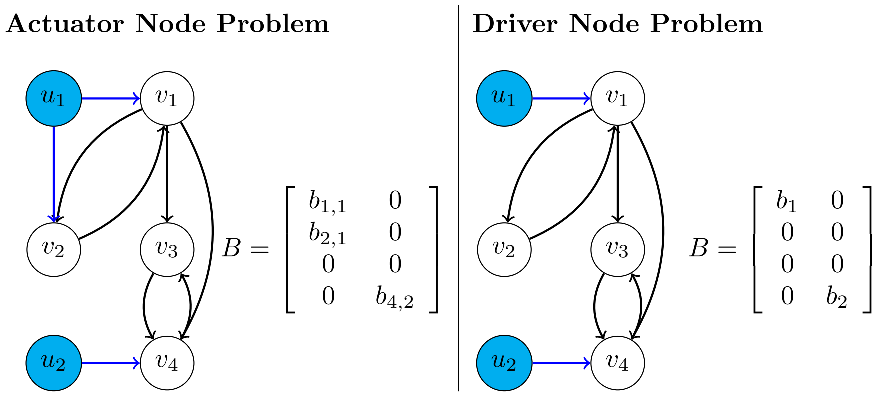

Actuator Node Problem
Driver Node Problem
$u_1$
$u_2$
$v_1$
$v_2$
$v_3$
$v_4$
$B = \begin{bmatrix} b_{1,1} & 0 \\ b_{2,1} & 0 \\ 0 & 0 \\ 0 & b_{4,2} \end{bmatrix}$
$B = \begin{bmatrix} b_1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & b_2 \end{bmatrix}$

**Figure 2.4:** Types of Minimum Controllability Problems. The blue nodes are the external inputs and the unfilled nodes represent the graph. (A) Actuator Problems allow each column of $B$ to have multiple non-zeros. (B) Driver Node Problems restrict each column of $B$ to have a single non-zero value.

1. **Actuator Node Framework (ANF):** In these problems, each column of the matrix $B$ is allowed to have multiple non-zero elements. Let $\mathcal{I}$ be the set of inputs corresponding to each column of $B$ so $|\mathcal{I}| = n_d$ and let $\mathcal{A}$ be the set of nodes whose corresponding rows in $B$ have at least one non-zero element. The function $\mathcal{B} : \mathcal{I} \mapsto 2^{\mathcal{A}}$ maps each input to the nodes whose rows in the input's column have non-zero elements. The matrix $B$ can thus be constructed,

$$B_{j,k} = \begin{cases} b_{j,k} \neq 0 & \text{if } v_j \in \mathcal{B}(k) \\ 0 & \text{otherwise} \end{cases}$$

   The values of $b_{j,k}$ for $v_j \in \mathcal{B}(k)$ are determined based on the particular method with which $B$ is found.

2. **Driver Node Framework (DNF):** In these problems, each column is restricted to having a single non-zero elements. Using the notation above, the input to actuator mapping is now a bijection $\mathcal{B} : \mathcal{I} \mapsto \mathcal{A}$ and so differentiating between inputs and actuators is redundant. We simplify the discussion of driver node problems by defining $\mathcal{D} \subseteq \mathcal{V}$ as the set of nodes which receive a control

input. The control input matrix is simply,

$$B_{j,k} = \begin{cases} 1 & \mathcal{B}(k) = j \\ 0 & \text{otherwise} \end{cases}$$

The distinction between ANF and DNF is exceptionally important and the lack of clarity in much of the literature has lead to confusion. An example of these two types of input frameworks is shown in Fig. 2.4 where we see in the ANF, some columns of $B$ may have more than one non-zero entry, while in the DNF, ever column of $B$ is restricted to having a single non-zero entry.

Let us additionally classify the types of controllability as well.

1. **Structural Controllability (SC):** From Def. 2.3.3, structural controllability is a property of the *structure*, $\mathcal{G}$, and the input matrix $B$. A system that is structurally controllable is denoted $(\mathcal{G}, B) \in \mathcal{SC}$ (as the specific weights do not play a role).

2. **Classic Controllability (CC):** From Def. 2.3.2, classic controllability is a property of the adjaceny matrix, $A(\mathcal{G}, w)$ for a specific weight function $w$ and input matrix $B$. A system that is controllable is denoted $(A, B) \in \mathcal{CC}$.

3. **Strong Structural Controllability (SSC):** From Def. 2.3.4, strong structural controllability is a property of the structure $\mathcal{G}$ and the input matrix $B$. A system that is strong structurally controllable is denoted $(\mathcal{G}, B) \in \mathcal{SSC}$.

4. **Reachability:** From Def. 2.3.5, reachability is a property of the adjacency matrix, $A(\mathcal{G}, w)$ for a specific weight function $w$, the input matrix $B$, and an initial and final condition $\boldsymbol{x}_0$ and $\boldsymbol{x}_f$. A system that is reachable is denoted $(A(\mathcal{G}, w), B, \boldsymbol{x}_0, \boldsymbol{x}_f) \in \mathcal{RE}$.

There are three main types of results concerning the control of complex networks.

1. **Minimum Controllability Problems (MCP):** These problems, in either the ANF or DNF, look for matrices $B$ with the minimum number of columns, $n_d$, to

satisfy one of the controllability criteria listed above. A slight variation to this problem is to set the number of columns of $B$ and instead minimize the number of non-zero elements of $B$.

2. **Efficient Control Problems (ECP):** These problems look for matrices $B$ which both (i) ensure one of the controllability criteria is satisfied and (ii) are *efficient* with respect to some metric satisfying the appropriate restrictions on $B$.

3. **Control Classification Problems (CCP):** These problems try to either classify (or rank) nodes based on some control related metric.

These three problems are closely related. For an ECP problem to be well defined, some knowledge of the minimum number of columns of $B$ found by an MCP is required. A useful CCP result can inform a better MCP or ECP formulation. The remainder of this section describes the major results for each of the above control problems.

### 2.3.3 Minimum Controllability Problems (MCP)

Minimum controllability problems on graphs attempt to minimize either the number of columns of $B$ or the number of non-zeros in $B$. Let the norm $||B||_0$ count the number of non-zero elements in the matrix $B \in \mathbb{R}^{n \times n_d}$ and let $\mathcal{CON}$ be one of the definitions of controllability ($\mathcal{SC}$, $\mathcal{CC}$, $\mathcal{SSC}$, or $\mathcal{RE}$) defined above.

1. **Minimum Actuator Node Problems (MANP):** In this type of problem, the number of columns of $B$ is fixed and the number of non-zero elements of $B$ to satisfy some controllability condition is minimized.

$$\begin{aligned} \min \quad & ||B||_0 \\ \text{s.t.} \quad & B \in \mathbb{R}^{n \times n_d} \\ & (\mathcal{G}, B) \in \mathcal{CON} \end{aligned}$$

2. **Minimum Input Problems (MIP):** In this type of problem, the number of

columns of $B$ is minimized while leaving the number of non-zero elements per column free.

$$\begin{aligned} \min \quad & |\mathcal{I}| \\ \text{s.t.} \quad & B \in \mathbb{R}^{n \times n_d} \\ & (\mathcal{G}, B) \in \mathcal{CON} \end{aligned}$$

The set of inputs $\mathcal{I}$ is defined in the description of the ANF above.

3. **Minimum Driver Node Problems (MDNP):** In this type of problem, the number of columns of $B$ is minimized while also constraining each column of $B$ to have a single non-zero element.

$$\begin{aligned} \min \quad & |\mathcal{D}| \\ \text{s.t.} \quad & B_{j,k} = \begin{cases} 1 & \mathcal{B}(k) = j \\ 0 & \text{otherwise} \end{cases} \\ & (\mathcal{G}, B) \in \mathcal{CON} \end{aligned}$$

The set of driver nodes $\mathcal{D}$ is defined in the description of the DNF above.

With the above taxonomy in mind, the current state of each problem is addressed.

The first major contribution to minimum controllability problems [75] solved the *Minimum Input Problem* to guarantee *Structural Controllability* (MIP-SC).

$$\begin{aligned} \min \quad & |\mathcal{I}| \\ \text{s.t.} \quad & (\mathcal{G}, B) \in \mathcal{SC} \end{aligned} \tag{2.3.3}$$

Before presenting the result, we define a useful induced graph which is used often in network control problems.

**Definition 2.3.6** (Bipartite Representation of a Directed Graph)**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed graph. The undirected graph $\mathcal{H} = (\mathcal{V}^+ \cup \mathcal{V}^-, \bar{\mathcal{E}})$ is induced by $\mathcal{G}$ where*

- *each node $v_j \in \mathcal{V}$ is represented by two nodes, $v_j^+ \in \mathcal{V}^+$ and $v_j^- \in \mathcal{V}^-$ and*
- *each directed edge $(v_j, v_k) \in \mathcal{E}$ is represented by an undirected edge $(v_j^+, v_k^-) \in \bar{\mathcal{E}}$.*

*The induced graph $\mathcal{H}$ is bipartite (see Def. 2.2.2) where the two sets of nodes are $\mathcal{V}^+$ and $\mathcal{V}^-$.*

The authors [75] called their answer to Eq. (2.3.3) the Minimum Input Theorem [75].

**Theorem 2.3.3** (Minimum Input Theorem [75])**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed graph and $\mathcal{H}$ be the induced undirected bipartite graph according to Def. 2.3.6. Let $\mathcal{M}^*$ be the maximum matching of $\mathcal{H}$ which can be found in polynomial time (see Def. 2.2.13). Then, the minimum number of inputs to guarantee structural controllability is equal to the number of unmatched nodes in $\mathcal{V}^-$.*

The Minimum Input Theorem determines in polynomial time the minimum number of columns of the matrix $B$ to guarantee structural controllability. The obvious follow-up question is how to construct the matrix $B$ of minimum dimensions. To find $B$, the authors use results from [68] that decompose graphs into stems (paths) and buds (cycles) which is detailed in their supplementary information [75].

1. Construct the matrix $\mathcal{H}$ according to the procedure in Def. 2.3.6 and compute a maximum matching of $\mathcal{G} = (\mathcal{V}, \mathcal{E}$ such that $\mathcal{M}^* \subseteq \mathcal{E}$.

2. Initialize the set of actuator nodes to the set of all unmatched nodes in $\mathcal{V}^-$.

3. The structure induced by the maximum matching $\mathcal{M}^*$ is a *cactus* which consists of unidirectional paths (or stems) and cycles (or buds). The root of each path is an unmatched node

As edge weights are typically not known precisely, this result is quite practical for many real linear systems described by graphs.

The minimum number of columns of $B$ was found to be quite large for many real datasets, especially gene regulatory networks, which sometimes were found to require $n_d \approx 0.8n$, or 80% of the network size. A response [76] made the point that controllability is less meaningful in a biological context as one is interested in phenotype expression, a particular final state, not the ability to drive the system to any point in phase space. Without using the terminology explicitly, the authors of the response describe a *reachability* problem (see Def. 2.3.5). By implication, the Minimum Input Problem for Structural Reachability (MIP-SR) has solution which is upper-bounded by the Minimum Input Theorem.

The expected size of the maximum matching of random graphs with simple degree distributions (such as $k$-regular graphs and Erdős-Rényi graphs) has been analyzed using the cavity method from statistical mechanics developed in [77–79].
If one is given a structurally uncontrollable system $(A, B)$, an iterative method to add additional columns to $B$ (either in the ANF or the DNF) until achieving structural controllability is presented in [80].

Another response to [75] examined the case where there is a self-loop (see Def. 2.2.8), at every node $v_j \in \mathcal{V}$ [81]. For this problem, the loops are treated as edges in the graph. If the graph is strongly connected (see Def. 2.2.12), the maximum matching of the induced bipartite graph is trivially $\mathcal{M}^* = \{(v_j, v_j) | \forall v_j \in \mathcal{V}\}$ and so a single control input is all that is required to guarantee structural controllability. The resulting matrix $B$ is a vector of all non-zero elements. The authors claim the Minimum Input Theorem for structural controllability is unable to provide meaningful results for graphs with many self-loops.

The MIP-SC in the DNF was solved in [82] where the authors provide a theorem using maximum matching and the SCC decomposition. Let $\mathcal{G}$ be decomposed into its SCCs with $\beta$ root SCCs.

**Theorem 2.3.4** (Minimum Driver Node Theorem [82])**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph and let $\mathcal{H}$ be the induced undirected bipartite graph constructed as in Def. 2.3.6. Let $\mathcal{M}^*$ be a maximum matching of $\mathcal{H}$ and let $\mathcal{G}$ have $\beta$ root SCCs. Let $\mathcal{U}(\mathcal{M}^*)$ be the set of* unmatched nodes *for a maximum matching $\mathcal{M}^*$. For each maximum matching, we count the number of SCCs with an unmatched node, and set the largest of these to be $\alpha$. The minimum number of driver nodes to guarantee structural controllability is equal to*

$$n_d = n - |\mathcal{M}^*| + \beta - \alpha$$

This result makes intuitive sense as $n - |\mathcal{M}^*|$ is the number of unmatched nodes and $\beta - \alpha$ is the minimum number of root SCCs without an unmatched node and so require a driver node as well.

Other MCPs for structural controllability including the following have been addressed. For the DNF where some nodes are forbidden from being a driver node, a polynomial time algorithm is given to find the minimum number of driver nodes [83]. A combined minimum input and output selection problem to guarantee structural controllability and observability is given in [84]. If there is a non-uniform cost associated with making a node an actuator node (or a driver node) then one can use a weighted maximum matching to solve the weighted MCP [85, 86].

For the case Thm. 2.3.3 returns the result that only a single input is required for structural controllability, [87] supplies an algorithm to find the vector $B = \boldsymbol{b}$ of maximum sparsity (i.e. minimizing the number of actuator nodes) to guarantee the pair $(\mathcal{G}, \boldsymbol{b})$ is structurally controllable, answering the MANP-SC question for this case. In a similar fashion [88], if one is given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, an algorithm based on Thm. 2.3.3 finds the minimum number of edges to add to $\mathcal{E}$ until a perfect matching exists, thus requiring a single control input (but possibly many actuator nodes).

Solving the Minimum Input Problem for Strong Structural Controllability (MIP-SSC) was shown to be NP-hard [89].

Additional work on applying structural controllability to temporal networks [90, 91], that is networks whose set of edges evolves in time, has received recent attention. The structural controllability of a graph's line graph (that is the graph $\mathcal{L}(\mathcal{G}) = (\mathcal{E}, \mathcal{F})$ where each edge in the original graph $\mathcal{E}$ is a node in its line graph and the new edges $(e_j, e_k) \in \mathcal{F}$ if the edges $e_j$ and $e_k$ share an endpoint node, is investigated in [92]. While structural controllability is appealing in that it is used to derive an efficient algorithm for determining the minimum number of independent inputs, if some of the edge weights are not independent from one another, or if some edge weights are known exactly, then the main assumption of Thm. 2.3.3 no longer holds. In particular, if the graph is undirected, then structural controllability cannot be applied directly as each pair of edges $(v_j, v_k)$ and $(v_k, v_j)$ have an equal weight and are not independent. A version of structural controllability applied to undirected graphs is investigated in [93] where a heuristic to solve the MIP-SC is presented. A number of independent research groups derived methods to determine both the minimum number of control inputs and which nodes should receive these inputs for the traditional definition of controllability.

Turning to the Minimum Input Problem for classic controllability (MIP-CC), in [94], the minimum number of inputs is shown to be equal to the maximum geometric multiplicity of the eigenvalues of the adjacency matrix $A$. This is an immediate result from the PBH rank criterion in Thm. 2.3.2. The paper suffers from some opacity about whether the result falls in the ANF or the DNF, but it can be shown that their solution is in the ANF (that is, if there is no repeated eigenvalue, then the single input may be attached to multiple actuator nodes). This result is applied specifically to circulant networks in [95]. The second part of [94] presents a method to construct the set of actuator nodes (not necessarily the minimum though as proven in [96]). The method requires transforming the matrix $A$ to column echelon form and determining which rows are multiples of the previous rows. The nodes corresponding to these rows require an input. Their method is generalized to multiplex networks in [97].

For the case where the adjacency matrx has no repeated eigenvalues, the result of [94] states that only a single input is needed. In this case, the control input matrix can be thought of as a vector $\boldsymbol{b} \in \mathbb{R}^n$ with no clear insight about the number of non-zeros elements in $\boldsymbol{b}$. The problem of determining the vector $\boldsymbol{b}$ with a minimum number of non-zeros is NP-hard as shown in [96] where the input selection problem is demonstrated to be equivalent to the *hitting set problem*, one of Karp's original 21 NP-complete problems [98].

**Definition 2.3.7** (Hitting Set Problem)**.** *Let $\mathcal{C}$ be a set of subsets of $\{1, 2, \ldots, m\}$. The minimum hitting set problem asks for the subset of $\mathcal{C}$ such that every integer in $\{1, 2, \ldots, m\}$ appears in at least one chosen element in the subset of $\mathcal{C}$.*

The main idea within the proof [96] is derived directly from the PBH criterion in

Thm. 2.3.2. Define $\mathcal{C}$ to be the support of each of the left eigenvectors of $A$, then $\boldsymbol{b}$ must have nonzeros corresponding to at least part of the support of each eigenvector. To find $\boldsymbol{b}$ with minimum non-zero elements in the support of each eigenvector, an instance of the Hitting Set Problem is constructed as defined in Def. 2.3.7 where each set $\mathcal{C}_k \in \mathcal{C}$ contains the support of the $k$'th eigenvector.

A particularly negative result [99] concerns *reachability* (see Def. 2.3.5). In [99], they show that the Minimum Input Problem for Reachability (MIP-RE) in the DNF to satisfy reachability can not be solved in polynomial time, it cannot be approximated in polynomial time either.

Taken togther, structural control problems are typically *easy* to solve (solvable in polynomial time) while classic control or strong structural control problems are NP-hard. Nonetheless, the literature provides a number of approximation algorithms to approximate the minimum number of inputs in either the ANF or the DNF. So far though, the problems examined focus on the case that we want to guarantee one of the controllability criteria on *all of the nodes*. The case where the controllability criteria must only be applied to a subset of the nodes is addressed in the following subsection.

### 2.3.4 Target Control

A topic we explore in great detail is *target controllability* which is a specific type of output controllability. Let $A$ be the adjacency matrix of a graph and $B$ be the control input matrix in either the actuator node framework or the driver node framework. If the matrix $C$ consists of rows with only a single non-zero entry each, that is each output is the state of a single node, and the triplet $(A, B, C)$ is output controllable, then equivalently it is *target controllable*. The problem of target controllability coincides with the problem of full controllability, discussed in previous subsection, when $C = I_n$, and so target controllability is a more general problem. Every type of problem mentioned previously has an immediate extension to the target control case.

The MCP for target control, even for structural control, does not map to an exact solution method, such as maximum matching [65]. Instead, a greedy heuristic is presented in [100]. If in addition, some nodes should not witness the affects of any control input, then one can use the ILP presented in [101]. Strong target structural control (the extension of Def. 2.3.4 to target control) is explored in [102, 103].

### 2.3.5 Controllability and Symmetry

The automorphism group of a graph (see Def. 2.2.23) and controllability properties are intimately related [104–106]. It is proven [104] that if there exists a symmetry (see Def. 2.2.22) such that $PB = B$, then the system is uncontrollable (a sufficient condition). On the other hand, the lack of any such symmetry does not prove the system is controllable.

This result is strengthened [105] to state that if there exists any matrix $P$ such that $AP = PA$ and $PB = B$, then the system is uncontrollable. The matrix $P$ is a fractional permutation corresponding to a fractional symmetry [107].

### 2.3.6 Nodal Control Metrics

Before moving on to other topics, here we briefly cover some controllability based metrics. A node's control centrality [108] is defined as the number of nodes which said node is able to individually control. Theoretically, this can be determined by the rank of the Kalman matrix $\mathcal{K}$ in Thm. 2.3.1 where $B = \boldsymbol{e}_k$, the $k$'th elementary vector. More practically, using Hosoe's theorem [109] this can be calculated as the number of nodes in the *stem-bud disjoint* subgraph (see Def. 2.2.9) rooted at this node. A seemingly identical definition, called *control range*, is given in [110]. In [111], nodes are called critical if they appear in every minimum actuator node set ensuring structural controllability. These critical nodes can be determined by examining the minimum dominating set [112, 113] of the graph. The theory of dominating sets has been used to analyze the critical nodes of biological networks [114]. More generally, the *control capacity* of a node is the fraction of minimum driver node sets in which it appears [115]. The emergence of critical nodes is investigated using core percolation in [116]. Nodes which must be assigned to be actuator nodes or driver nodes are classified in [117] as either nodes with no incoming edges (sources), surplus sinks (nodes with no outgoing edges), or internal dilations.

### 2.3.7 Efficient Control Problems (ECP)

In the previous subsections, we were interested in MCPs which try to find the input matrix $B$ which in some sense is *minimal* (number of columns, number of non-zeros) that ensures some sense of *controllability* (structrual controllability, classic controllability, etc.). While clearly an important endeavor, it is only concerned with the minimum requirement of controllability, while offering nothing to compare two matrices $B$ and $B'$, of equal dimension or equal number of non-zeros depending on the context, which appear as local minima in the solution of the MCP.

For the rest of this section, we will assume that $B$ ensures the system is *controllable in the intended sense.* Consider Eq. (2.3.1) along with some prescribed final output $\boldsymbol{y}_f$. As we have assumed $B$ yields output controllability, there exists an infinite number of control inputs $\boldsymbol{u}$ which can drive the system from its initial condition $\boldsymbol{x}_0$ to $\boldsymbol{y}_f$. The control energy of any such control input is defined as,

$$E = \int_0^{t_f} \boldsymbol{u}^T(t)\boldsymbol{u}(t)dt \tag{2.3.4}$$

As the control to perform the task is not unique, we look for the control input which minimizes Eq. (2.3.4).

To find the control input which minimizes Eq. (2.3.4), let us define the particular optimal control problem (see the next section which discusses optimal control in more detail) which is the focus in this manuscript.

$$\begin{aligned} \min \quad & J = \frac{1}{2}\int_0^{t_f} \boldsymbol{u}^T(t)\boldsymbol{u}(t)dt \\ \text{s.t.} \quad & \dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + B\boldsymbol{u}(t) \\ & \boldsymbol{x}(0) = \boldsymbol{x}_0, \quad \boldsymbol{y}_f = C\boldsymbol{x}(t_f) \end{aligned} \tag{2.3.5}$$

Pontryagin's minimum principle is used to solve Eq. (2.3.5) in Appendix B.1 and B.2. The optimal control input is found to be,

$$\boldsymbol{u}^*(t) = B^T e^{A^T(t_f - t)} C^T \left(CW(t_f)C^T\right)^{-1} (\boldsymbol{y}_f - Ce^{At_f}\boldsymbol{x}_0) \tag{2.3.6}$$

where $W(t_f) \in \mathbb{R}^{n\times n}$ is the controllability Gramian, a symmetric positive semi-definite matrix, which can be formally written,

$$W(t_f) = \int_0^{t_f} e^{A\tau} BB^T e^{A^T\tau} d\tau. \tag{2.3.7}$$

The controllability Gramian is the solution of the differential Lyapunov equation [118, 119]

$$\dot{W}(t) = AW(t) + W(t)A^T + BB^T, \quad W(0) = O_n \tag{2.3.8}$$

which is a linear ODE with constant nonhomogeneous term. If $A$ is Hurwitz, then there exists a single stable fixed point solution, $W$, of Eq. (2.3.8) which can be found by setting $\dot{W}(t) = O_n$. Computing $W(t_f)$ can be done using numerical differential equation solvers with Eq. (2.3.8), but often the steady state controllability Gramian $W$ is used instead as many real systems are stable and the time-varying portion of $W(t)$ decays exponentially if $A$ is Hurwitz. To do this, we use the algorithm presented in [120] and implemented in SLICOT [121]. While this method is reasonable up to dimension $n = \mathcal{O}(100)$, to find controllability Gramians for larger systems, a low rank approximation method is used [122–124]. The controllability Gramian plays an extremely important role in both qualifying and quantifying the controllability of linear systems.

**Theorem 2.3.5** (Singularity of the Controllability Gramian [118]). *A system $(A, B)$ is controllable if and only if the controllability Gramian $W(t_f)$ is non-singular. A system $(A, B, C)$ is output controllable if and only if the matrix $CW(t_f)C^T$ is non-singular.*

Moreover, if $(A, B, C)$ is output controllable, and using the fact that $W(t_f)$ is postive semi-definite, then $CW(t_f)C^T$ must be positive definite. The minimum control energy (using Eq. (2.3.4) and Eq. (2.3.6)) is,

$$E^* = \boldsymbol{\beta}^T (CW(t_f)C^T)^{-1} \boldsymbol{\beta}, \tag{2.3.9}$$

where $\boldsymbol{\beta} = \boldsymbol{y}_f - Ce^{At_f}\boldsymbol{x}_0$ is the *control maneuver*. The minimum control energy in Eq. (2.3.9) is a useful metric to judge the control efficacy for a particular choice of $B$.

Recently, Eq. (2.3.9) has garnered a lot of attention throughout the complex networks field. For the case $C = I_n$, and assuming $||\boldsymbol{\beta}||_2 = 1$, it is clear from the Rayleigh-Ritz theorem that,

$$\frac{1}{\mu_{\max}} \leq E^* \leq \frac{1}{\mu_{\min}}$$

where $\mu_{\max}$ and $\mu_{\min}$ are the largest and smallest eigenvalues of $W(t_f)$, respectively. The behavior of $\frac{1}{\mu_{\max}}$ and $\frac{1}{\mu_{\min}}$ in both the small and large $t_f$ regimes are presented in [125]. Let the eigenvalue-eigenvector pairs of the controllability Gramian be denoted $W(t_f)\boldsymbol{\xi}_k = \mu_k \boldsymbol{\xi}_k$ for $k = 1, \ldots, n$. The eigen-energies of the system are found by setting $\boldsymbol{\beta} = \boldsymbol{\xi}_k$ so that the $k$'th eigen-energy is $E_k = \frac{1}{\mu_k}$. The distribution of the eigen-energies is investigated in [126] for both artifical networks and real networks. For networks with unbounded degree distributions, such as scale-free graphs, the eigen-energies are shown to also follow a power-law distribution.

Another consequence of the spectrum of eigen-energies is that, as some of the eigenvalues of $W$ are extremely small, its condition number may be smaller than the finite precision arithmetic can handle. To compensate, many of the studies on the controllability Gramian require extended precision using tools like GMP [127], MPFR [128], MPC [129] and those libraries built on top of them like Advanpix [130]. A more important question than simply whether the system is controllable then is whether the system is *numerically controllable.* A system is numerically controllable if the condition number of the controllability Gramian is less than machine $\epsilon$ associated with the level of precision used in its computation. The transition from numerical uncontrollablility to numerical controllablility is called the *controllability transition*

[131].

We are now ready to ask ‘which set of driver nodes is the most efficient (or effective) for a given task?’ Let us break down this question in some detail as it is deceptively complicated. There are three components:

1. The number of inputs (in the ANF) or the number of driver nodes (in the DNF) which have lower bounds corresponding to the minimum controllability questions answered previously in Thms. 2.3.3 and 2.3.4, respectively. We may also impose an upper bound due to financial or spatial restrictions.

2. The metric which we would like to maximize. If we know the exact control maneuver we must make, $\boldsymbol{\beta}$, then one can attempt to minimize $\boldsymbol{\beta}^T W^{-1} \boldsymbol{\beta}$ but this problem will be shown to be NP-hard. Alternative surrogate cost functions, or control metrics, with beneficial properties are often used instead.

3. The maximum allowed control energy (or related metric). In this case, the number of driver nodes is minimized while satisfying some maximum allowed control energy constraint.

In the DNF, if one restricts the number of driver nodes to be some $n_d < n$, one can attempt to do a brute force search over the $\binom{n}{n_d}$ to maximize a control energy metric but for large $n$, the required computational time is infeasible as this is a *combinatorial* problem.

One of the first attempts to quantitatively explore the trade-offs between the number of driver nodes and the control energy was presented in [132]. The focus was on the smallest eigenvalue of the controllability Gramian as an arbitrary control maneuver $\boldsymbol{\beta}$ will have at least some component in the eigen-direction corresponding to it. The results therein are applied to brain networks in [133]. In terms of discrete time systems, a similar trade-off was explored between number of driver nodes and control time [134].

Some control metrics are discussed in the series of papers [135–138]. In this series of papers, each metric is shown to either be or not be a submodular set function.

**Definition 2.3.8** (Submodular Set Functions [139])**.** *Let $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ be a function that maps the power set of $\mathcal{V}$ to the real numbers. Let $\mathcal{S}_2 \subseteq \mathcal{S}_1 \subseteq \mathcal{V}$ and let $a \in \mathcal{V}$ and $a \notin \mathcal{S}_1$. Then $f$ is* submodular *if,*

$$f(\mathcal{S}_2 \cup \{a\}) - f(\mathcal{S}_2) \geq f(\mathcal{S}_1 \cup \{a\}) - f(\mathcal{S}_2) \tag{2.3.10}$$

Note the *diminishing returns* property of Eq. (2.3.10), that is, adding an element to a larger set offers a smaller improvement than adding the same element to a smaller set. Let the gain be denoted $\Delta(a|\mathcal{S}) = f(\mathcal{S} \cup \{a\}) - f(\mathcal{S})$. The benefit of proving a set function, $f$, is submodular is that maximizing $f$ subject to a cardinality constraint $|\mathcal{S}| \leq k$ can be approximated with a greedy algorithm. For submodular set function optimization, the greedy algorithm returns solution $\mathcal{S}_{greedy}$ while the true optimal solution $\mathcal{S}_{optimal}$ satisfies the bound [140, 141],

$$\frac{f(\mathcal{S}_{greedy})}{f(\mathcal{S}_{optimal})} \geq 1 - \left(\frac{k-1}{k}\right)^k \geq \frac{e-1}{e} \approx 0.63$$

where we assume $f(\emptyset) = 0$, The first metric examined was the trace of the controllability Gramian (or output controllability Gramian) [135]. The controllability Gramian for single driver nodes is explored in [142]. Next, the trace of the inverse Gramian, the log determinant of the Gramian, and the rank of the Gramian, were all shown to be submodular set functions [136] using Def. 2.3.8. Submodular set function optimization is used on metrics for observability as well [143, 144] for optimal sensor placement problem.

Leader selection is a similar problem to the controllability problems described previously with the main difference being the inputs $\mathcal{I} \subseteq \mathcal{V}$, that is, the inputs are selected from the nodes within the graph. Set function optimization is used as a way to choose such *leader nodes* in consensus problems to minimize convergence error [145, 146] and in the precense of *link noise* [147].

In [148] (and its follow-up [149]) the following NP-hard optimization problem is investigated,

$$\begin{aligned} \min_{\mathcal{D}\subseteq\mathcal{V}} \quad & |\mathcal{D}| \\ \text{s.t.} \quad & (A,B) \text{ is controllable} \\ & \boldsymbol{\beta}^T W^{-1} \boldsymbol{\beta} \leq E \end{aligned} \tag{2.3.11}$$

for some upper energetic threshold $E$. Note in this case the authors do not replace the control energy with an energetic metric, but rather face the control energy expression head-on. An approximation algorithm is provided [148] which guarantees a solution up to a factor $\mathcal{O}(\log n)$ using results on supermodular constrained set covering problems [150]. The optimization problem in Eq. (2.3.11) is reinvestigated in [149] along with

$$\begin{aligned} \min_{\mathcal{D}\subseteq\mathcal{V}} \quad & \text{trace}(W^{-1}) \\ \text{s.t.} \quad & |\mathcal{D}| \leq n_d \end{aligned}$$

More sophisticated approximation algorithms are presented for both NP-hard optimization problems in [149].

A radically different approach was taken in the series of papers [151–155] where the authors developed a gradient descent type optimization applied directly to Eq. (2.3.7). The gradient descent based algorithm is used to find local minima of the optimization problem,

$$\begin{aligned} \min_{B\in\mathbb{R}^{n\times n_d}} \quad & \text{trace}(W^{-1}X_f) \\ \text{s.t.} \quad & \text{trace}(B^T B) - n_d - \epsilon = 0 \end{aligned} \tag{2.3.12}$$

where $X_f = e^{At_f}(\boldsymbol{x}_0\boldsymbol{x}_0^T)e^{A^T t_f}$, $n_d$ is the requested number of inputs and $\epsilon > 0$ is a small tolerance. Note that this problem sets the number of columns of $B$ constant and the number of nonzeros in each column of $B$ is not restricted, that is, this problem is an ANP. They call their algorithm the Projected Gradient Method (PGM) which they prove (i) it converges and (ii) it converges to a local minimum. They derive explicit expressions for the derivatives of the cost function and the constraint in Eq. (2.3.12) with respect to $B$. To convert the solution of Eq. (2.3.12) to the DNF, the authors use a rounding of the continuous variables in $B$ which they call *key node selection*. Each row of $B$ is summed, $\boldsymbol{r} = B\mathbf{1}_{n_d}$, and then the nodes corresponding to the $n_d$ largest values of $\boldsymbol{r}$ are chosen to be the driver nodes, and $B$ is constructed accordingly. The algorithm is revisted in more detail in [155]

The subsequent papers added more functionality and improved performance to PGM. A second order derivative of the cost function in Eq. (2.3.12) is derived and a trust region method [156] is used in [152]. The cost function is modified to include an $\int \boldsymbol{x}^T \boldsymbol{x} dt$ term in [154]. Extensions to target control and the inclusion of a probabilistic gradient descent method are described in [153]. The PGM described requires knowledge of the entire network. Instead, in [157], only local information is used to find optimal driver node sets using a hybrid of matching and minimizing the longest path between a driver node and a target node.

## 2.4 Optimal Control

The controllability Gramian arises as the central part of the solution of the Minimum Energy Control Problem, a famous optimal control problem. More generally, in this section, we address the theory of optimal control for any nonlinear cost function and nonlinear dynamical system. In particular, Pontryagin's method is described to derive necessary conditions for the solution of an optimal control problem and pseudospectral optimal control is described as a numerical method for solving optimal control problems. Two biological applications of optimal control using pseudospectral optimal control [10, 11] are presented at the end.

### 2.4.1 Pontryagin's Minimum Principle

Optimal control is concerned with finding a control input to make a system perform a certain action while satisfying a set of constraints on the end-points and throughout the trajectory. The general optimal control problem [158, 159] can be expressed as follows.

$$\begin{aligned}
\min \quad & J = E(\boldsymbol{x}(t_0), \boldsymbol{x}(t_f), t_0, t_f) + \int_{t_0}^{t_f} F(\boldsymbol{x}(t), \boldsymbol{u}(t), t)dt \\
\text{s.t.} \quad & \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \\
& \boldsymbol{e}(\boldsymbol{x}(t_0), \boldsymbol{x}(t_f), t_0, t_f) = \boldsymbol{0} \\
& \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) = \boldsymbol{0}
\end{aligned} \tag{2.4.1}$$

The cost function consists of two parts, a function $E : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ which applies a weight to the state of the system at the initial and final times and a function $F : \mathbb{R}^n \times \mathbb{R}^{n_d} \times \mathbb{R} \mapsto \mathbb{R}$ which applies a weight for the duration of the control action that depends on the state and control time traces. Both of these functions are assumed to be at least once differentiable in each of their arguments. The dynamical equation $\boldsymbol{f} : \mathbb{R}^n \times \mathbb{R}^{n_d} \times \mathbb{R} \mapsto \mathbb{R}^n$ describes the controlled evolution of the system. The end-point constraints $\boldsymbol{e} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}^{n_e}$ consist of $n_e$ expressions to constraint the states at the initial and final times, such as an initial condition. The path constraints $\boldsymbol{h} : \mathbb{R}^n \times \mathbb{R}^{n_d} \times \mathbb{R} \mapsto \mathbb{R}^{n_h}$ are used to impose realism on the system or to ensure safety constraints are preserved. Without going into details, any solution to Eq. (2.4.1) must also satisfy the Hamiltonian minimization problem,

$$\begin{aligned}
\min \quad & F(\boldsymbol{x}(t), \boldsymbol{u}(t), t) + \boldsymbol{\lambda}^T(t)\boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \\
\text{s.t.} \quad & \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) = \boldsymbol{0}
\end{aligned} \tag{2.4.2}$$

where $\boldsymbol{\lambda}(t) \in \mathcal{R}^n$ is the vector of co-states and the end-point minimization problem,

$$\begin{aligned}
\min \quad & E(\boldsymbol{x}(t_0), \boldsymbol{x}(t_f), t_0, t_f) \\
\text{s.t.} \quad & \boldsymbol{e}(\boldsymbol{x}(t_0), \boldsymbol{x}(t_f), t_0, t_f) = \boldsymbol{0}
\end{aligned} \tag{2.4.3}$$

An optimal control that is at least a local minimum is found from the stationarity condition for Eq. (2.4.2) by differentiating its Langragian with respect to the input,

$$\frac{\partial}{\partial \boldsymbol{u}} \big( F(\boldsymbol{x}, \boldsymbol{u}, t) + \boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t) + \boldsymbol{\mu}^T \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}, t) \big) = \boldsymbol{0} \tag{2.4.4}$$

where the co-states evolve according to the derivative of the Lagrangian of the Hamiltonian minimization problem with respect to the states,

$$\dot{\boldsymbol{\lambda}}(t) = -\frac{\partial}{\partial \boldsymbol{x}} \left( F(\boldsymbol{x}, \boldsymbol{u}, t) + \boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t) + \boldsymbol{\mu}^T \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}, t) \right) \tag{2.4.5}$$

with initial and final values found from the transversality condition determined by differentiating the Lagrangian of the end-point ,

$$\begin{aligned} \boldsymbol{\lambda}(t_0) &= -\frac{\partial}{\boldsymbol{x}(t_0)} \left( E(\boldsymbol{x}(t_0), \boldsymbol{x}(t_f), t_0, t_f) + \boldsymbol{\nu}^T \boldsymbol{e}(\boldsymbol{x}(t_0), \boldsymbol{x}(t_f), t_0, t_f) \right) \\ \boldsymbol{\lambda}(t_f) &= \frac{\partial}{\partial \boldsymbol{x}(t_f)} \left( E(\boldsymbol{x}(t_0), \boldsymbol{x}(t_f), t_0, t_f) + \boldsymbol{\nu}^T \boldsymbol{e}(\boldsymbol{x}(t_0), \boldsymbol{x}(t_f), t_0, t_f) \right) \end{aligned} \tag{2.4.6}$$

The necessary conditions written in Eqs. (2.4.4), (2.4.5), and (2.4.6) are used as validation when solving the original optimal control problem in Eq. (2.4.1) numerically. These conditions are used to solve the general minimum energy target control problem and derive the controllability Gramian in Appendix A. The following subsection presents a numerical scheme to solve Eq. (2.4.1) which is applied to two biological systems by the author.

### 2.4.2 Pseudospectral Optimal Control

In general, there is no analytic solution to Eq. (2.4.1) and so numerical methods must be used instead. A number of methods using the relationship between the evolution of the states in the constraints of Eq. (2.4.1) and the costates in Eqs. (2.4.5) and (2.4.6) have been used such as shooting methods and boundary value problem solving methods. These methods which require solving for the evolution of the costates are called indirect methods. With the advances in computing speeds and storage capabilities, direct methods are now competitive alternatives to these earlier methods which instead attempt to solve the optimization problem directly by transcribing it to a nonlinear programming problem. The transcription is done by the following procedure [160–163]:

1. A set of $n_n$ *collocation points* are computed as the roots of an othogonal poly-

nomial at which we will compute the states and controls.

2. To discretize the states and controls, we approximate them on a basis of Lagrange interpolating polynomials.

$$L_j = \prod_{\substack{k=0 \\ k \neq j}}^{n_n} \frac{\tau - \tau_k}{\tau_j - \tau_k}, \quad j = 0, \ldots, n_n \tag{2.4.7}$$

The time-varying vector of states and controls are represented as $n_n$ vectors $X_k$ and $U_k$, $k = 0, \ldots, n_n$, respectively, which can be recovered using,

$$\begin{aligned} \boldsymbol{x}(\tau) \approx \boldsymbol{X}(\tau) &= \sum_{k=0}^{n_n} \boldsymbol{X}_k L_k(\tau) \\ \boldsymbol{u}(\tau) \approx \boldsymbol{U}(\tau) &= \sum_{k=0}^{n_n} \boldsymbol{U}_k L_k(\tau) \end{aligned} \tag{2.4.8}$$

3. The dynamical equations are represented as,

$$\dot{x}(\tau_j) \approx \sum_{k=0}^{n_n} X_k \dot{L}_j(\tau_k) \tag{2.4.9}$$

Often the derivatives of the Lagrange interpolating polynomials in Eq. (2.4.7) are denoted in matrix form $D_{k,j} = \dot{L}_j(\tau_k)$. The integral in the cost function, similarly, is approximated with Gauss-Legendre quadrature as we have discretized the states and controls at the roots of a Legendre polynomial.

4. After applying the approximate forms of the dynamical equations in Eq. (2.4.9), the NLP is composed,

$$\begin{aligned} \min \quad & \hat{J} = E(\boldsymbol{X}_0, \boldsymbol{X}_f, t_0, t_f) + \sum_{k=0}^{n_n} w_k F(\boldsymbol{X}_k, \boldsymbol{U}_k, \tau_k) \\ \text{s.t.} \quad & \sum_{j=0}^{n_n} D_{k,j} \boldsymbol{X}_j - \frac{t_f - t_0}{2} \boldsymbol{f}(\boldsymbol{X}_k, \boldsymbol{U}_k, \tau_k) = \boldsymbol{0}, \quad k = 0, \ldots, n_n \\ & \boldsymbol{e}(\boldsymbol{X}_0, \boldsymbol{X}_f, t_0, t_f) = \boldsymbol{0} \\ & \boldsymbol{h}(\boldsymbol{X}_k, \boldsymbol{U}_k, \tau_k) = \boldsymbol{0}, \quad k = 0, \ldots, n_n \end{aligned} \tag{2.4.10}$$

The nonlinear programming in Eq. (2.4.10) can be solved using many different techniques such as interior point algorithms (implemented in IPOPT) [164] or sequential quadratic programming (implemented in SNOPT) [165]. The states and controls can be recovered using Eq. (2.4.8). A number of implementations of pseudospectral optimal control exist such as the proprietary codes DIDO [166] and GPOPS [163] and open source implementations such as $\mathcal{PSOPT}$ [167]. In the applications of pseudospectral control contained in the following subsections, we use $\mathcal{PSOPT}$ which performs the discretization described above to transcribe an optimal control problem to a nonlinear programming problem (NLP), and then we use IPOPT [164] to solve the resulting NLP.

Two applications of numerical optimal control to which the author contributed are discussed in the following two subsections. The first examines a simple model of autophagy and attempts to determine combinations of drug interventions which can either up-regulate or down-regulate the number of autophagic vesicles produced. The second examines the FDA approved model of diabetes and determines a minimum input (minimum insulin) regimen to maintain a diabetic patient's blood glucose level after ingesting a meal.

### 2.4.3 Controlling and Regulating Autophagy [10]

Autophagy is a cellular recycling process that plays an important role in protein synthesis through degradation of cytoplasmic contents [168]. It has been shown to be a 'double-edged sword' [169] in terms cancer, where autophagy can either reduce the stress of the tumor environment or induce cell death if recycling is excessive. Thus, there may be benefits to either up-regulate (increase) or down-regulate (decrease) the current level of autophagic vesicle production in the absence of any external drug's influence

The mathematical model of autophagy is constructed with four kinases that have been shown to play critical roles in regulating autophagy and are also potential drug targets. The model captures important features shown to occur when cells under

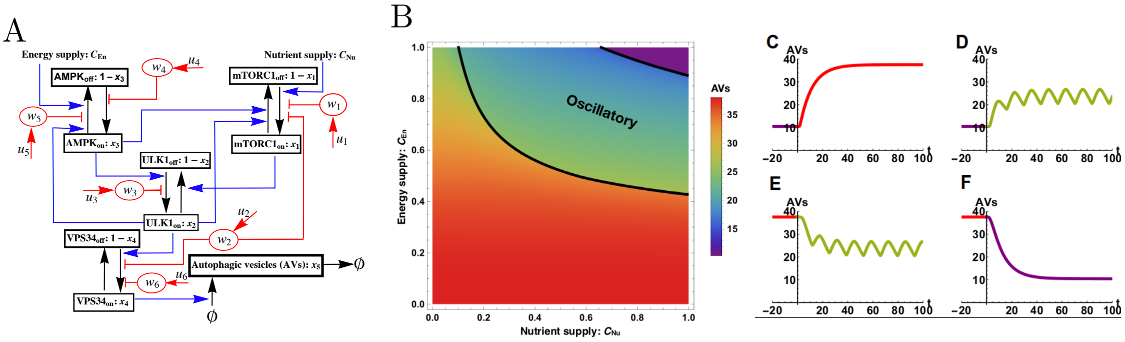


**Figure 2.5:** The model of autophagy using differential equations. (A) The diagram which describes the relationship between each kinase and whether it plays a role turning it on or off, as well as each control input and whether it inhibits or activates each kinase. (B) The steady state behavior of the autophagic vesicle production as a function of the energy supply $C_{En}$ and the nutrient supply $C_{Nu}$. (C)-(F) Sample time trajectories of autophagic vesicle production after a coordinated change in $C_{En}$ and $C_{Nu}$. (C) The response to a drop from $C_{Nu} = C_{En} = 1$ to $C_{Nu} = C_{En} = 0.2$. (D) The response to a drop from $C_{Nu} = C_{En} = 1$ to $C_{Nu} = C_{En} = 0.6$. (E) The response to a rise from $C_{Nu} = C_{En} = 0.2$ to $C_{Nu} = C_{En} = 0.6$. (F) The response to a rise from $C_{Nu} = C_{En} = 0.2$ to $C_{Nu} = C_{En} = 1$. Adapted from [10].

low stress (low level of autophagy), moderate stress (oscillatory behavior) and high stress (high level of autophagy). A schematic of the dynamical system is shown in Fig. 2.5(A) where each state is represented by two black boxes, one which represents the 'off-state' and one which represented the 'on-stage' for the four kinases, and one additional state for the number of autophagic vesicles. The interactions between states are drawn as blue arrows. The six drugs introduced to this system are drawn in red, where each circle represents the drug concentration state. Red edges represent how the drug concentrations may either activate or inhibit each of the kinases. For the set of parameters considered (collected in Table A.1), the steady state behavior of the autophagic vesicles, as a function of the parameters $C_{Nu}$ and $C_{En}$ are shown in Fig. 2.5(B) where three distinct regions are scene. For either low $C_{En}$ and/or low $C_{Nu}$, there is high autophagic vesicle production. For intermediate values of $C_{En}$ and $C_{Nu}$, the number of autophagic vesicles oscillates about a moderate value. Finally, for high $C_{En}$ and high $C_{Nu}$, the autophagic vesicle production is low.

Four step responses to changes in $C_{En}$ and $C_{Nu}$ are shown in Figs. 2.5(C) thru 2.5(F).

The dynamical system that describes autophagy in an average cell consists of 5 differential equations that describe the internal state of the cell and an additional 6 differential equations that describe the current drug concentration of possible drug interventions.

$$\begin{aligned}
\dot{x}_1(t) &= (1 - x_1(t))C_{Nu}H(w_1(t))H(w_2(t)) - x_1(t)h_{12}(x_2(t))h_{13}(x_3(t)) \\
\dot{x}_2(t) &= (1 - x_2(t))h_{23}(x_3(t))H(w_3(t)) - x_2(t)h_{21}(x_1(t)) \\
\dot{x}_3(t) &= (1 - x_3(t))k_1H(w_4(t)) - C_{En}x_2(t)x_3(t)H(w_5(t)) \\
\dot{x}_4(t) &= (1 - x_4(t))h_{42}(x_2(t))H(w_2(t))H(w_6(t)) - k_{@}x_4(t) \\
\dot{x}_5(t) &= k_3x_4(t) - k_4x_5(t) \\
\dot{w}_k(t) &= u_k(t) - \delta_k w_k(t), \quad k = 1, \ldots, 6
\end{aligned} \tag{2.4.11}$$

where the kinase interaction Hill functions $h_{kj}$ have the form,

$$h_{kj}(x_j(t)) = r_{b,kj} + (r_{m,kj} - r_{b,kj})\frac{x_j^{n_{kj}}(t)}{x_j^{n_{kj}}(t) + \theta_{kj}^{n_{kj}}} \tag{2.4.12}$$

and the drug concentrations also appear in Hill functions $H$ of the form,

$$H(w_k(t)) = r_m - (r_m - r_b)\frac{w_k^n(t)}{w_k^n(t) - \theta^n} \tag{2.4.13}$$

The Hill functions in Eqs. (2.4.12) and (2.4.13) have sinusoidal behavior in terms of the input.

A therapy is defined as a subset of the six possible drugs, denoted $\mathcal{T} \subseteq \mathcal{D} = \{1, 2, 3, 4, 5, 6\}$. We are interested in answering two questions; (i) whether a particular therapy is capable of up-regulating or down-regulating autophagy given particular values of $C_{En}$ and $C_{Nu}$ and (ii) which therapies are the most efficient at performing the particular tasks. These tasks can be framed as a two phase optimal control problem,

$$\begin{aligned}
\min \quad & J = \sum_{j \in \mathcal{T}} \int_0^{t_f} u_j(t)dt \\
\text{s.t.} \quad & \dot{\boldsymbol{X}}(t) = \boldsymbol{f}(\boldsymbol{X}(t), \boldsymbol{u}(t)), \quad 0 \le t \le t_f \\
& u_j(t) = 0, \quad \forall j \in \mathcal{D} \backslash \mathcal{T}, \quad 0 \le t \le t_f \\
& x_5^f - \epsilon \le x_5(t) \le x_5^f + \epsilon, \quad t_1 \le t \le t_f \\
& 0 \le w_j(t) \le w_j^{\max}, \quad \forall j \in \mathcal{D} \\
& 0 \le u_j(t), \quad \forall j \in \mathcal{D} \\
& \boldsymbol{X}(0) = \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{0}_6 \end{bmatrix}
\end{aligned} \tag{2.4.14}$$

where $\boldsymbol{X}(t) = \begin{bmatrix} \boldsymbol{x}^T(t) & \boldsymbol{w}^T(t) \end{bmatrix}^T$ is a vector of all eleven states satisfying Eq. (2.4.11) with initial condition $x_k(0) = x_{k,0}$ for $k = 1, \ldots, 5$ and $w_k(0) = 0$ for $k = 1, \ldots, 6$, i.e., initially there is no drug present in the system. The drug concentrations have an upper bound determined as the level at which the drug becomes toxic, denoted $w_j^{\max}$ for $j = 1, \ldots, 6$. If a drug is not present in the therapy $\mathcal{T}$, then $u_j(t) = 0$ for all time. The first phase of the optimal control problem drives the number of AVs to the desired level, between $x_5^f - \epsilon \le x_5(t) \le x_5^f + \epsilon$, while the second phase maintains the number of AVs to be within this interval.

To describe the optimal control, we plot the integral of the drug, i.e., the total amount of drug administered up to time $t$, which we denote $r_{k,j}^*(t) = \int_{t_0}^t u_k(\tau)dt$ where $j$ denotes the number of drugs in the therapy. This can be more enlightening as the optimal control input we find has pulsatile behavior, i.e., as the cost function is linear in $\boldsymbol{u}$, there may be Dirac-delta function-like behavior. Not all of the drugs are capable of performing the desired control goals laid out in the optimal control problem in Eq. (2.4.14). Drugs 2 and 6 were found capable of down-regulating the number of AVs to certain levels on their own as shown in the first two rows of Fig. 2.6. Drug 5 is capable of up-regulating the number of AVs alone as shown in the third row of Fig. 2.6. From the first column of Fig. 2.6, we see that there are a number of jumps, representing

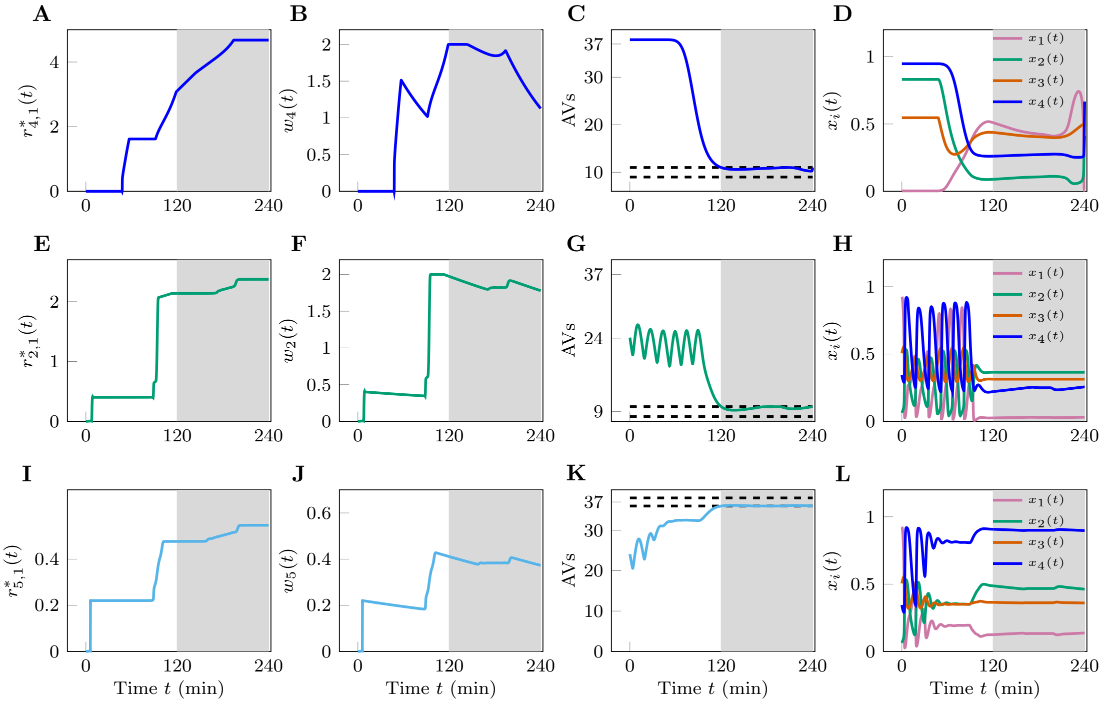


**Figure 2.6:** Single drug therapies for down-regulating and up-regulating autophagy. In the first row, drug 4 is used to down-regulate autophagy to an interval around $x_5^* = 10$. In the second row, drug 2 is used to down-regulate autophagy to an interval around $x_5^* = 9$. In the third row, drug 5 is used to up-regulate autophagy to an interval around $x_5^* = 37$. The first column shows the optimal amount of drug developed as time grows. The second column shows the current drug concentration as a function of time. The number of autophagic vesicles as a function of time is shown in the third column and the time evolution of the other four states is shown in the fourth column.

pulses in the optimal control input. A detailed analysis of the single drug therapies is shown in the supplementary information of [10] in order to understand which dual drug therapies are more efficient. A selection of dual drug therapies is shown in Fig. 2.7. In the first row, drugs 2 and 6 are used to down-regulate the number of AVs to a level that neither drug is capable of doing alone when the parameters are set to $C_{Nu} = C_{En} = 0.1$. This non-obvious therapy is the type of result that numerical models are useful for uncovering, as one may be unwilling to try a therapy composed of two drugs incapable of performing the desired task. The second row shows the dual therapy composed of drugs 2 and 6 again, but with the parameters $C_{Nu} = C_{En} = 0.6$. For these values of $C_{Nu}$ and $C_{En}$, drug 2 can perform the desired control goal but drug 6 cannot. When combined into a dual therapy, the optimal solution reduces the amount of drug 2 administered by over 80%. This result suggests that one can reduce the amount of drug 2 required to down-regulate autophagy by introducing drug 6. The third row shows the dual therapy composed of drugs 3 and 6 to down-regulate autophagy with the parameters set to $C_{Nu} = C_{En} = 0.6$. Similar to the second row, drug 3 is capable of performing the desired control goal while drug 6 cannot, but when combined the amount of drug 3 required is reduced. The fourth row shows the optimal control to down-regulate autophagy using drugs 1 and 6 with the parameters set to $C_{Nu} = C_{En} = 0.6$. This combination is particularly interesting as neither drug is capable of performing the control goal alone, but drug 1 is uncapable of down-regulating autophagy at all. Dual therapies like drugs 1 and 6 work in tandem with each other which can be seen in panels Figs. 2.7(O) and 2.7(P) where first drug 1 in a single pulse increases the number of AVs, outside the oscillatory regime, at which points drug 6, in a single pulse, down-regulates the number of AVs to the desired level.

While the results shown in Figs. 2.6 and 2.7 are positive, the optimal control problem is capable of determining non-effective dual therapies as well, that is, those which are neither no better than the respective single drug therapies, or are incapable of performing the desired control goal. Beyond the dual therapies, triple or quadruple

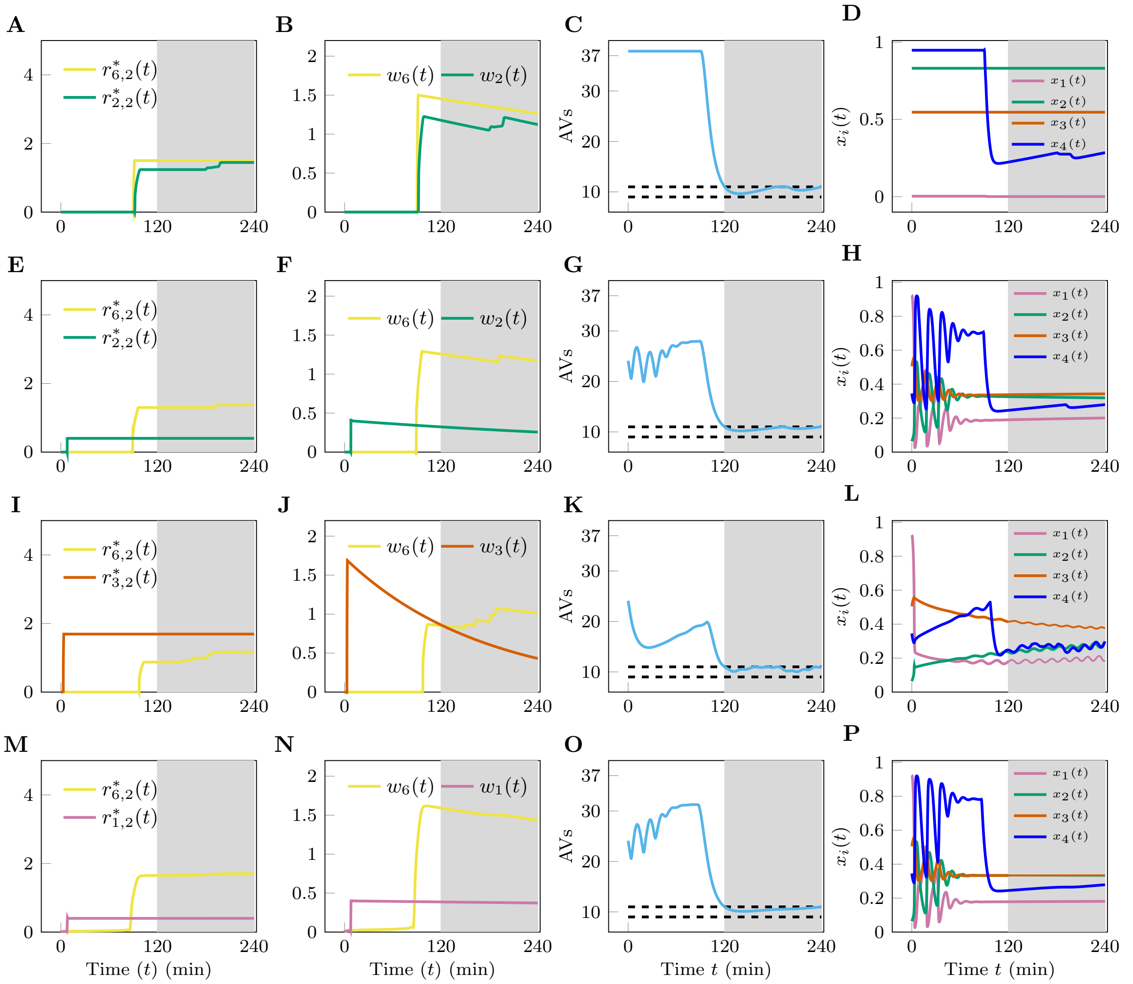


**Figure 2.7:** Duel drug therapies for down-regulating autophagy. The first row uses drugs 2 and 6 to down-regulate autophagy for $C_{Nu} = C_{En} = 0.1$. The second row uses drugs 2 and 6 to down-regulate autophagy for $C_{Nu} = C_{En} = 0.6$. The third row uses drugs 3 and 6 to down-regulate autophagy for $C_{Nu} = C_{En} = 0.6$. The four row uses drugs 1 and 6 to down-regulate autophagy for $C_{Nu} = C_{En} = 0.6$. The first column is the amount of drug used up to time $t$. The second column is the current drug concentration at each time. The third column is the amount of AVs and the fourth column is the time trajectory of the remaining four states.

therapies can be systematically investigated, as running numerical simulations are vastly cheaper than in vivo tests. Successful numerical results can guide which drug combinations should be investigated in vivo though, especially those combinations which are non-obvious when examining each drug's individual capabilities.

### 2.4.4 Regulating Glucose in Type I Diabetes [11]

A second example of numerical optimal control concerns blood glucose levels in Type I diabetics. Patients with Type I Diabetes are unable to produce insulin due to their immune system attacking the beta cells which would normally produce it. Instead, insulin is typically injected by the patient prior to injesting glucose, where the amount to inject is determined from the current blood glucose level, the size of the meal, and the patient's sensitivity to insulin. Since the 1990s though, insulin pumps were developed which are capable of both delivering a basal level of insulin constantly as well as delivering an insulin bolus prior to meals, similar to the injections self-delivered previously. A number of control techniques have been developed for the insulin available pumps, such as PID, linear MPC, and probabilistic predict control. Each of these control methods though use simplified versions of the full FDA approved *in silico* model of blood glucose regulation.

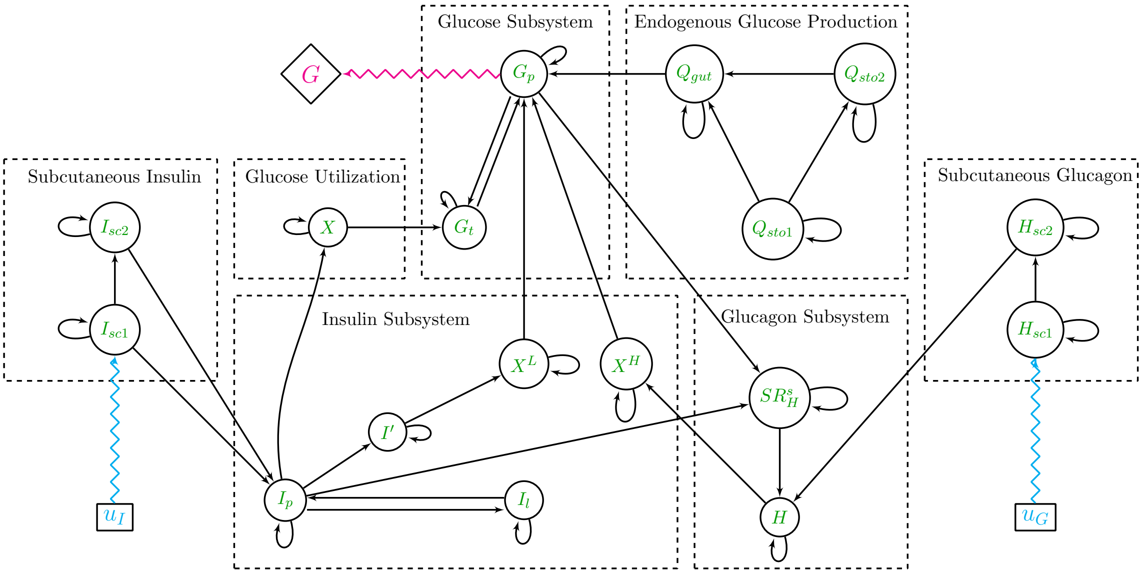
Rather than using a simplified version of the model, we develop an optimal control problem using the complete model. In addition to the model described by 17 differential equations, $x_k(t)$, we include two inputs, insulin $u_I(t)$ and glucagon $u_G(t)$, and a meal of known size and time, $D(t)$. To avoid off-target affects, the amount of insulin and glucagon is limited with an integral constraint and the amount of glucose must not drop below a certain value (which would lead to hypoglycemia) or exceed a certain value (which would lead to hyperglycemia). In addition to the time-varying portion of the model, there also exist a set of parameters, $\Theta_{G,B}$. The complete optimal control problem can be stated as,

$$\begin{aligned}
\min \quad & J = \int_{t_0}^{t_f} \left[\alpha_p BGI(G(t)) + \alpha_I u_I^p(t) + \alpha_G u_G^p(t)\right] dt, \quad p \in \{1, 2\} \\
\text{s.t.} \quad & \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), D(t), \Theta_{G_b}), \quad \boldsymbol{u}(t) = [\boldsymbol{u}_I(t), \quad \boldsymbol{u}_G(t)]^T \\
& G(t) = x_1(t)/V_G \\
& G_L \leq G(t) \leq G_U, \quad u_I^L \leq u_I(t) \leq u_I^U, \quad 0 \leq U_G(t) \leq u_G^U \\
& 0 \leq \int_{t_0}^{t_f} u_I(t)dt \leq \Phi_I^U, \quad 0 \leq \int_{t_0}^{t_f} u_G(t)dt \leq \Phi_G^U \\
& \mathbf{x}(t_0) = \bar{\boldsymbol{x}}
\end{aligned} \tag{2.4.15}$$

where the function $BGI(G(t))$ is called the *Blood Glucose Index* defined as,

$$BGI(G(t)) = 10\left(1.509\left((\ln G(t))^{1.084} - 5.3811\right)\right)^2 \tag{2.4.16}$$

assigns a risk of hypoglycemia or hyperglycemia to a particular blood glucose level. Details of the dynamical equations are contained in Appendix A.2. A network representation of the dynamical equations that govern the glucose-insulin-glucagon system is shown in Fig. 2.8. The dashed line blocks are the various subsystems which compose the full system. The two driver nodes, $I_{sc1}$ and $H_{sc1}$, directly receive the inputs $u_I$ (insulin) and $u_G$ (glucagon), respectively. The single target node is the blood glucose level, $G_p$. A plot of the cost $BGI(G(t))$, defined in Eq. (2.4.16), is shown in Fig. 2.9(A) with the minimum represented as $G_d$, the desired blood glucose level. The convexity of this function in the range of $G(\cdot)$ of interest is clear. In type I diabetics, a useful application of the pump beyond responding to the input of glucose from a meal, is to provide a constant basal level of insulin so that in the absence of a meal their blood glucose remains near to $G_d$. In Fig. 2.9(B), the blood glucose level for three selections of basal insulin inputs, $u_b$ are shown, 0, $1.2 \times 10^{-3}$, and $2.4 \times 10^{-3}$, the third of which is capable of lowering the blood glucose level to the desired level in the absence of a meal. This value appears in the optimal control problem in Eq. (2.4.15) as the lower bound for the insulin input $u_I(t)$. To judge the quality of a



$\mathcal{V} = \{G_p, G_t, I_l, I_p, I', X^L, Q_{sto1}, Q_{sto2}, Q_{gut}, X, SR^s_H, H, X^H, I_{sc1}, I_{sc2}, H_{sc1}, H_{sc2}\}$
$\mathcal{D} = \{I_{sc1}, H_{sc1}\}$
$\mathcal{T} = \{G_p\}$

**Figure 2.8:** A graph representation of the FDA approved dynamical model of the insulin-glucose system in humans. An edge from state $x_k$ to $x_j$ implies that $x_k(t)$ appears in the dynamical equation for $\dot{x}_j(t)$. The blocks comprising sets of nodes represent the subsystems that make up the entire system. The inputs are injected into states $I_{sc1}(t)$ and $H_{sc1}(t)$, making them the driver nodes, and the target, $G_p(t)$, is the value which we would like to control.

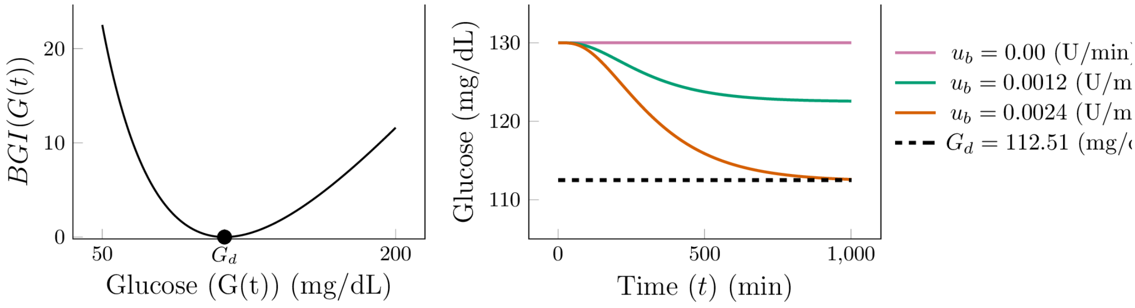


**Figure 2.9:** On the left, the plot of $BGI(G)$ showing that there exists a single minimum located at the desired glucose level, $G_d$. On the right, plots of the blood glucose, $G(t)$, responding to constant basal insulin rates, $u_b$. This analysis is used to determine what constant value of insulin should be provided through the pump so that, in the absence of a meal, $G(t) \to G_d$.

solution, from an optimal solution we determine the minimum value that the blood glucose achieves, denoted $G^{\min}$, the maximum value that the blood glucose achieves, denoted $G^{\max}$, and the integral of the $BGI$, denoted,

$$\Delta = \int_{t_0}^{t_f} BGI(G(t))dt \tag{2.4.17}$$

Also used is the total amount of insulin delivered to the patient, denoted,

$$\phi_I = \int_0^{t_f} u_I(t)dt \tag{2.4.18}$$

The first cost function we consider sets $\alpha_G = 0$, $\alpha_I = 1$, and $\alpha_p = \epsilon$. In Fig. 2.10, the effect of different choices of $\epsilon$ and $\phi_I$ are shown. For $p = 1$, we call the problem the Regulation and Minimum Fuel (ReMF) problem while for $p = 2$, we call the problem the Regulation and Minimum Energy (ReME) problem. These two choices yield different insulin schedules, where the ReMF problem typically results in a pulsatile solution, that is $u_I(t)$ may have a point discontinuity, while the ReME problem typically results in a continuous insulin regime.

In all of the simulations, we choose the final time $t_f = 300$ and the patient consumes a meal of 70 grams of glucose at time $t = 60$. The meal is represented by a delta function, $D(t) = 70\delta(t-60)$. We compare the optimal solutions found with the *standard therapy* which consists of 10 units of insulin injected 30 minutes before the meal.

As a function of $\epsilon$, there is a dynamic range for each of the metrics, $G^{\min}$, $G^{\max}$, $\Delta$ (defined in Eq. (2.4.17)), and $\phi_I$ (defined in Eq. (2.4.18), outside of which exponential changes in $\epsilon$ lead to little to no change in these metrics. From this analysis, we are able to find a value of $\epsilon$ that yields a solution with small $\Delta$ without $\epsilon$ becoming excessively large for both the ReMF and the ReME frameworks, which are circled in each of the panels in Fig. 2.10. These choices of $\epsilon$ are used to solve the optimal control problem in the ReMF and ReME, respectively, shown in Fig. 2.11. The time trace of the blood glucose level using the ReMF framework (blue), the ReME framework

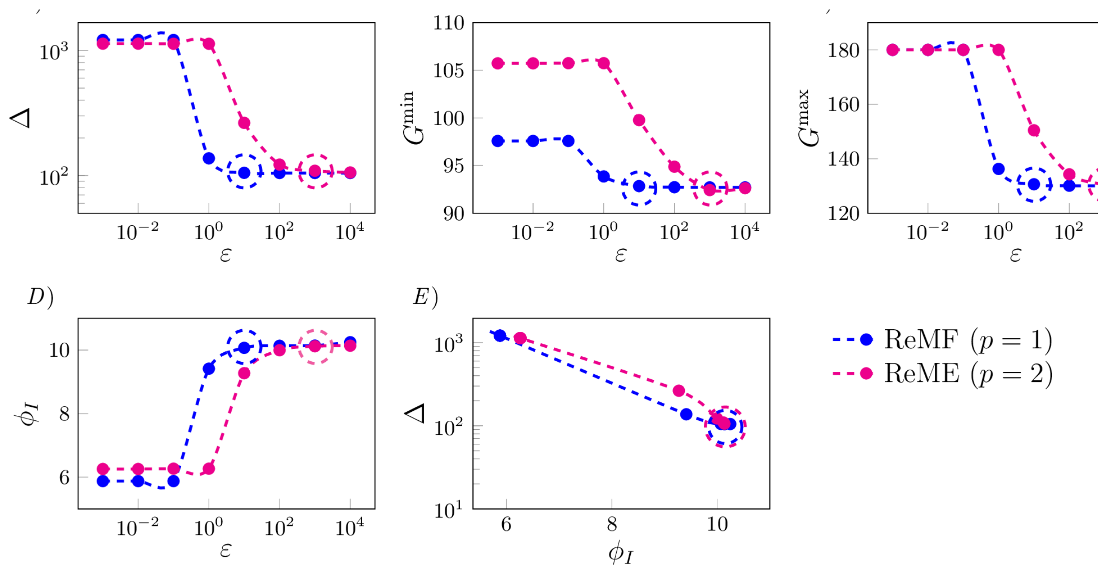


**Figure 2.10:** The effect of the weighting term $\epsilon$ on the metrics used to judge the quality of an optimal solution. We see for each metric there is a dynamic range for which it changes rapidly with $\epsilon$ but that outside this range of $\epsilon$, there is little or no change. The blue dashed lines correspond to the ReMF framework while the pink dashed lines correspond to the ReME framework. The circled points correspond to the value of $\epsilon$ chosen as a trade-off between keeping $\Delta$ small while ensuring $\phi_I$ is not excessively large.

(pink), and the standard therapy (orange) is shown in Fig. 2.11(A). We see that the standard therapy, while having a smaller initial minimum value before the meal and a smaller overshoot after the meal, the blood glucose is significantly lower than the other two therapies. The optimal glucose time traces are almost indistinguishable, despite the fact that in Fig. 2.11(B) we see that the ReMF insulin infusion rate has much sharper, and larger peak than the ReME insulin infusion rate. The cumulative insulin, $r_I(t) = \int_0^t u_I(\tau) d\tau$, for each of the three therapies is shown in Fig. 2.11(C). We see that for both ReMF and ReME frameworks, the integrals of the two curves are nearly equal as seen by the sharp increases. The linear slope afterwards is due to the basal level of insulin also provided from time $t = 50$ to the final time $t_f = 300$.

## 2.5 Conclusion

This chapter presented many of the key definitions and theorems used throughout the remainder of this dissertation concerning graph theory, control theory, networks, and the control of complex networks. An attempt was made to coalesce the many

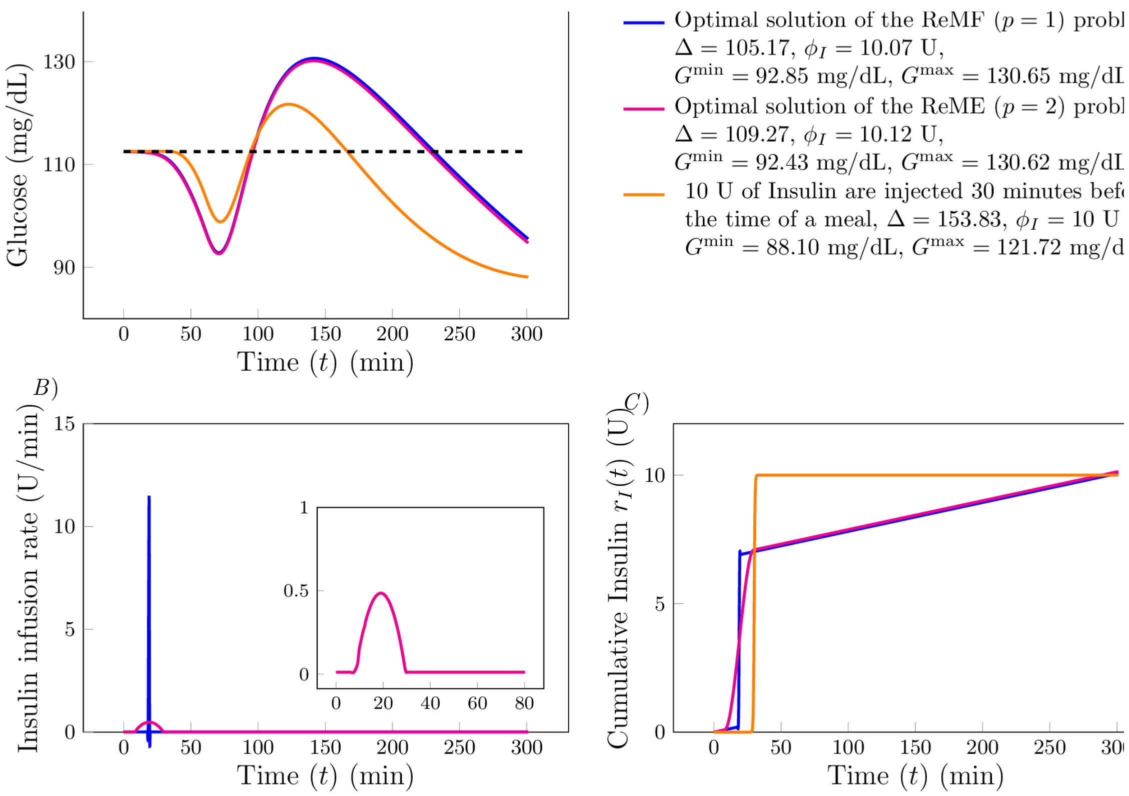


**Figure 2.11:** For the values of $\epsilon$ chosen in Fig. 2.10, the optimal solutions in the ReMF and ReME frameworks are computed and plotted in blue and pink, respectively. The time trace of the glucose for each of the therapies is shown in (A), the insulin infusion rate is shown in (B), and the cumulative insulin supplied is shown in (C). The time trace of the glucose and the cumulative insulin used for the standard therapy are shown in orange for comparison in (A) and (C), respectively.

definitions of types of control problems (structural controllability versus classical controllability), types of input frameworks (driver nodes versus actuator nodes), and optimization problems (minimal controllability versus efficient control). The review of the current state of the field of controlling complex networks is necessary to properly place the results contained in the following sections as the confusion between structural controllability and classic controllability especially continues to plague the field.

Throughout this dissertation, only the classic notion of controllability 2.3.2 and the driver node framework is used whenever we choose an input matrix $B$. In the next chapter, random graphs are constructed using methods described above such Erdős-Rényi graphs and Scale Free graphs and driver nodes are selected randomly such that controllability is ensured according to the Minimum Driver Node Theorem 2.3.4 or the positive definiteness of the controllability Gramian using Thm 2.3.5.

Graph symmetries are investigated in detail in Chapter 4 where we make use of permutations 2.2.21, symmetries 2.2.22, and the automorphism group 2.2.23.

The controllability Gramian in 2.3.8 is investigated in detail for lattice graphs in Chapter 5 where exact expressions for the control energy, Eq. (2.3.9), in terms of graph properties are derived.

# Chapter 3

# Target Control

## 3.1 Introduction

The motivation for this chapter finds its origin from the results in [126] which, as discussed in the previous chapter, found that the spectrum of eigen-energies spans multiple orders of magnitude. This result implies that, as an arbitrary control maneuver $\boldsymbol{\beta}$ has components along each eigen-direction of the controllability Gramian, the control energy is composed of a term proportional to the *worst-case energy.*

$$\boldsymbol{\beta}^T W^{-1}(t_f) \boldsymbol{\beta} = \sum_{k=1}^{n} \frac{1}{\mu_k} \left( \boldsymbol{\beta}^T \boldsymbol{\xi}_k \right)^2 \tag{3.1.1}$$

On the one hand, it would seem that controlling complex networks is hopeless as for generic $\boldsymbol{\beta}$, $\boldsymbol{\beta}^T \boldsymbol{\xi}_1 \neq 0$ and $\frac{1}{\mu_1}$ is extremely large. On the other hand, one should ask whether the control application truly demands complete controllability. This chapter presents two alternative frameworks to that of complete controllability:

1. **Target Control:** If only a subset of all of the nodes, $\mathcal{T} \subset \mathcal{V}$ must be driven to some final state, then the matrix $W^{-1}(t_f)$ in Eq. (3.1.1) is replaced by $(CW(t_f)C^T)^{-1}$ where $CW(t_f)C^T$ is a principal submatrix of $W(t_f)$, which has a provably larger minimum eigenvalue than the original matrix.

2. **Balanced Control:** One issue that is not often mentioned in the literature is the requirement that, at time $t_f$, the state must be equal *exactly* to the

prescribed final state $\boldsymbol{x}(t_f) = \boldsymbol{x}_f$. If one relaxes this constraint so that instead we demand $||\boldsymbol{x}(t_f) - \boldsymbol{x}_f||$ is small, the control energy can be reduced.

The general problem, where some nodes have a final constraint (target control) and other nodes have a final cost (balanced control) is addressed in Appendix B.1. Numerical results for the combined problem are informative for demonstrating the transition between target control and balanced control. The general result is then specialized to both the target control case in Section 3.2 and the balanced control case Section 3.3. An extension of balanced control to handle uncertainty in the system matrix is demonstrated in Section 3.4. Additionally, at the end of the chapter, we discuss applications to an iterative control strategy using the geometry of minimum energy control in Section 3.5 and then an application to countermeasures against malicious attacks in Section 3.6.

## 3.2 Scaling of Control Energy [1]

This section covers the main results published in [1] where spectral properties of principal submatrices of the controllability Gramian (see Eqs. (2.3.7) and (2.3.8)) are investigated numerically and some scaling behavior is explained analytically. The results for the general problem (with both final costs and final constraints) derived in Appendix B.1 is specialized to the case when there are only final constraints. Details of the specializations are contained in Appendix B.2 so only the results are presented here for brevity.

The optimal control problem that is solved in this section is,

$$\begin{aligned} \min \quad & J = \frac{1}{2}\int_0^{t_f} \boldsymbol{u}^T(t)\boldsymbol{u}(t)dt \\ \text{s.t.} \quad & \dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + B\boldsymbol{u}(t) \\ & \boldsymbol{x}(0) = \boldsymbol{x}_0, \quad C\boldsymbol{x}(t_f) = \boldsymbol{y}_f \end{aligned} \tag{3.2.1}$$

The solution of Eq. (3.2.1) is contained in Appendix B.2, but here we focus on only the control energy. The control energy can be expressed as a quadratic form with the

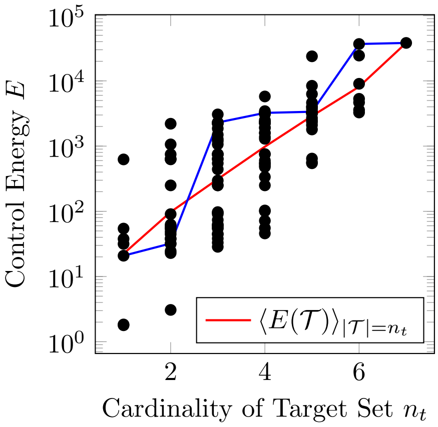


**Figure 3.1:** Control Energy for all possible sets of target nodes in the graph shown in Fig. B.1(A) in Appendix B.1. The final states are always set to $x_{j,f} = 1$ and the final time $t_f = 2$. The red line is the geometric mean of the control energy over each cardinality. The blue line represents the control energy for a particular sequence of adding target nodes one by one, namely in the sequence, $(v_4, v_3, v_5, v_1, v_0, v_6, v_2)$.

inverse of the output controllability Gramian.

$$E(\mathcal{T}) = \boldsymbol{\beta}^T \left(CW(t_f)C^T\right)^{-1} \boldsymbol{\beta} \tag{3.2.2}$$

In this section, we assume that the triplet $(A, B, C)$ is output controllable (see Thm. 2.3.1) so that the matrix $CW(t_f)C^T$ is positive definite (even if $W(t_f)$ is nonsingular). If we instead define $\boldsymbol{\beta} \in \mathbb{R}^n$ and substitute $\boldsymbol{\beta}C^T$ into Eq. (3.2.2), we show that Eq. (3.2.2) is strictly less than Eq. (3.1.1). From the definition of the $C$ matrix, that is, each of its rows is a distinct unit vector corresponding to each of the target nodes, the output controllability Gramian $\bar{W} = CW(t_f)C^T$ is a *principal submatrix* of the controllability Gramian.

Holding the graph's adjacency matrix $A$, the set of driver nodes $\mathcal{D}$ (and thus $B$), and the desired final target state, $\boldsymbol{y}_f$, constant, the control energy in Eq. (3.2.2) can be thought of as a function of the set of targets, $\mathcal{T}$. For each number of target nodes $1 \leq n_t \leq n$, there are $\binom{n}{n_t}$ ways to select the target nodes. The control energy of graphs was first thoroughly investigated in [126] which implicitly considered the case that $\mathcal{T} = \mathcal{V}$, i.e., the traditional concept of *controllability* as opposed to *target controllability*. An important fact that follows directly from the optimal control problem in Eq. (3.2.1) is that the control energy monotonically increases with respect to target node sets $\mathcal{T}_1 \subseteq \mathcal{T}_2$, that is,

$$E(\mathcal{T}_1) \leq E(\mathcal{T}_2) \tag{3.2.3}$$

holding the final conditions for those nodes $v_j \in \mathcal{T}_1$ constant. We plot the control energies for the network shown in Fig. B.1(A) (which can be found in Appendix B.1) where the driver nodes are still selected to be $\mathcal{D} = \{v_1, v_2\}$ and the set of target nodes is iterated over all possible subsets of $\mathcal{V}$ in Fig. 3.1. We see that while clearly the number of target nodes, $n_t$, does not determine the order of magnitude of the control energy, at least for this small graph, a trend can be observed,

$$\langle \log_{10} \left( \mathcal{E}(\mathcal{T}) \right) \rangle_{|\mathcal{T}|=n_t} \sim \eta \frac{n_t}{n} \tag{3.2.4}$$

where the average is taken over all sets of target nodes of cardinality equal to $n_t$ and $\eta$ is the rate of increase with $n_t$. We choose to take the average of the logarithm of the control energies in Eq. (3.2.4) (equivalently, the logarithm of the geometric mean) because even when we hold $n_t$ constant, the control energy values span across multiple orders of magnitude which would lead to the algebraic mean over-representing the few largest control energy target node sets. The geometric mean is shown as the red curve in Fig. 3.1. An example of the monotonicity property of the control energy with respect to a particular target node sequence, as expressed in Eq. (3.2.3), is shown as the blue curve in Fig. 3.1 where the sequence is listed in its caption. The inconsistent rate of increase of $E$ with $n_t$ is typical for generic sequences of nodes.

Note that the control energy in Eq. (3.2.2) depends on the specific choice of the control maneuver $\boldsymbol{\beta}$. To eliminate these additional variables we examine the *worst-case control energy.*

**Definition 3.2.1** (Worst-Case Control Energy [1])**.** *We assume output controllability. From the Rayleigh-Ritz theorem, for any control maneuver $\boldsymbol{\beta}_{\mathcal{T}}$, we can bound the* *control energy using,*

$$0 < \frac{||\boldsymbol{\beta}||_2^2}{\mu_{\max}} \leq \boldsymbol{\beta}^T \bar{W}^{-1} \boldsymbol{\beta} \leq \frac{||\boldsymbol{\beta}||_2^2}{\mu_{\min}} < \infty$$

*where $\mu_{\min}$ and $\mu_{\max}$ are the smallest and largest eigenvalues of $\bar{W}(t_f)$, respectively. Thus, the worst case control energy, i.e., the maximum control energy for constant $||\boldsymbol{\beta}||_2^2$, corresponds to the case the control maneuver is parallel with the eigenvector corresponding to $\mu_{\min}$ and,*

$$E_{\max}(\mathcal{T}) = \max_{||\boldsymbol{\beta}||_2^2=1} E(\mathcal{T}) = \frac{1}{\mu_{\min}} \tag{3.2.5}$$

An important property we use in the derivation is the *Cauchy Interlacing Theorem.*

**Theorem 3.2.1** (Cauchy Interlacing Theorem [170])**.** *Let $M$ be a n-by-n square, positive definite and symmetric matrix and let $M'$ be a principal submatrix of $M$ of dimension $m < n$. Then if the eigenvalues of $M$, denoted $\lambda_k(M)$, $k = 1, \dots, n$ and $M'$, denoted as $\lambda_k(M')$, $k = 1, \dots, m$, are ordered as $\lambda_{k+1}(M) \leq \lambda_k(M)$, and $\lambda_{k+1}(M') \leq \lambda_k(M')$, the eigenvalues* interlace *as,*

$$\lambda_{m+1}(M) \leq \lambda_m(M') \leq \lambda_m(M) \leq \ldots \leq \lambda_2(M) \leq \lambda_1(M') \leq \lambda_1(M)$$

To better describe the iterative process below, we associated each controllability Gramian with a particular set of target nodes with subscripts $W_{\mathcal{T}}$. Define $\mathcal{S}$ as a sequence of the nodes such that $\mathcal{S}_j \subseteq \mathcal{S}_k \subseteq \mathcal{V}$ for $j \leq k$ and $|\mathcal{S}_k| = k$, $k = 1, \dots, n$. To derive the rate of increase of the smallest eigenvalue of the output controllability Gramian, we relate $W_{\mathcal{S}_j} = W_j$ to $W_{\mathcal{S}_{j+1}} = W_{j+1}$ for any generic sequence $\mathcal{S}$. We can relate the two output controllability Gramians,

$$\begin{aligned} W_{j+1} &= \bar{W}_j + dW_j \\ &= \begin{bmatrix} 0 & \mathbf{0}_{n_t}^T \\ \boldsymbol{w}_j & W_j \end{bmatrix} + \begin{bmatrix} w_{j,j} & \boldsymbol{w}_j^T \\ \mathbf{0}_{n_t} & O_{n_t,n_t} \end{bmatrix} \end{aligned} \tag{3.2.6}$$

Let $\mu_{\text{min}}^{(j+1)}$ and $\mu_{\text{min}}^{(j)}$ be the smallest eigenvalues of $W_{j+1}$ and $\bar{W}_j$, respectively, with associated eigenvectors $\boldsymbol{\xi}_{j+1}$ and $\bar{\boldsymbol{\xi}}_j$ and from Thm. 3.2.1 we know $\mu_{\text{min}}^{(j+1)} \leq \mu_{\text{min}}^{(j)}$.

$$W_{j+1}\boldsymbol{\xi}_{j+1} = \mu_{\text{min}}^{(j+1)}\boldsymbol{\xi}_{j+1} \quad \text{and} \quad \bar{W}_j\bar{\boldsymbol{\xi}}_j = \mu_{\text{min}}^{(j)}\bar{\boldsymbol{\xi}}_j$$

Then, pre-multiplying Eq. (3.2.6) by $\boldsymbol{\xi}_{\text{min}}^T(\mathcal{T})$ and post-multiplying by $\bar{\boldsymbol{\xi}}_{\text{min}}(\bar{\mathcal{T}})$ yields,

$$\begin{aligned}
\boldsymbol{\xi} T W_{\mathcal{T}}\bar{\boldsymbol{\xi}} &= \boldsymbol{\xi}^T\bar{W}_{\bar{\mathcal{T}}}\bar{\boldsymbol{\xi}} + \boldsymbol{\xi}^T dW_{\bar{\mathcal{T}}}\bar{\boldsymbol{\xi}} \\
\mu_{\text{min}}(\mathcal{T})\boldsymbol{\xi}^T\bar{\boldsymbol{\xi}} &= \mu_{\text{min}}(\bar{\mathcal{T}})\boldsymbol{\xi}^T\bar{\boldsymbol{\xi}} + \boldsymbol{\xi}^T W_{\mathcal{T}} W_{\mathcal{T}}^{-1} dW_{\bar{\mathcal{T}}}\bar{\boldsymbol{\xi}} \\
\mu_{\text{min}}(\mathcal{T}) &= \mu_{\text{min}}(\bar{\mathcal{T}}) + \mu_{\text{min}}(\mathcal{T})\frac{\boldsymbol{\xi}^T W_{\mathcal{T}}^{-1} dW_{\bar{\mathcal{T}}}\bar{\boldsymbol{\xi}}}{\boldsymbol{\xi}^T\bar{\boldsymbol{\xi}}}
\end{aligned}$$

Using Def. 3.2.1, the increase of control energy from $\bar{\mathcal{T}}$ to $\mathcal{T} = \bar{\mathcal{T}} \cup v_k$ is,

$$E_{\text{max}}(\mathcal{T}) = E_{\text{max}}(\bar{\mathcal{T}})\eta_k$$

Thus, for each sequence of nodes, there are increases $\eta_k$, $k = 1, \ldots, n$. Taking the logarithm we get,

$$\log E_{\text{max}}(\mathcal{T}) - \log E_{\text{max}}(\bar{\mathcal{T}}) = \log \eta_k > 0.$$

Now, taking any two subsets among the sequence $\mathcal{S}_k \subset \mathcal{S}_j$, we define the increase,

$$\log E_{\text{max}}^{(j)} - \log E_{\text{max}}^{(k)} = \sum_{\ell=k}^{j-1} \log \eta_\ell = (j-k)\log \bar{\eta}_{k\to j} \tag{3.2.7}$$

Note that the value $\bar{\eta}_{k\to j}$ depends only on the specific sets $\mathcal{S}_k$ and $\mathcal{S}_j$ and not the path taken represented by the intermediate sets $\mathcal{S}_\ell$, $k < \ell < j$. We see numerically that for a particular graph, the average over all pairs of sets $\mathcal{S}_k \subset \mathcal{S}_j$, $\langle \log \bar{\eta}_{k\to j}\rangle = \frac{\eta}{n}$. This suggests that, fixing $k$, we can write the average control energy as,

$$\begin{aligned}
\langle \log E_{\text{max}}^{(n_t)}\rangle &= \langle \log E_{\text{max}}^{(k)}\rangle - \frac{k}{n}\eta + \frac{n_t}{n}\eta \\
&= \frac{n_t}{n}\eta + C_k \\
&\sim \frac{n_t}{n}\eta
\end{aligned}$$

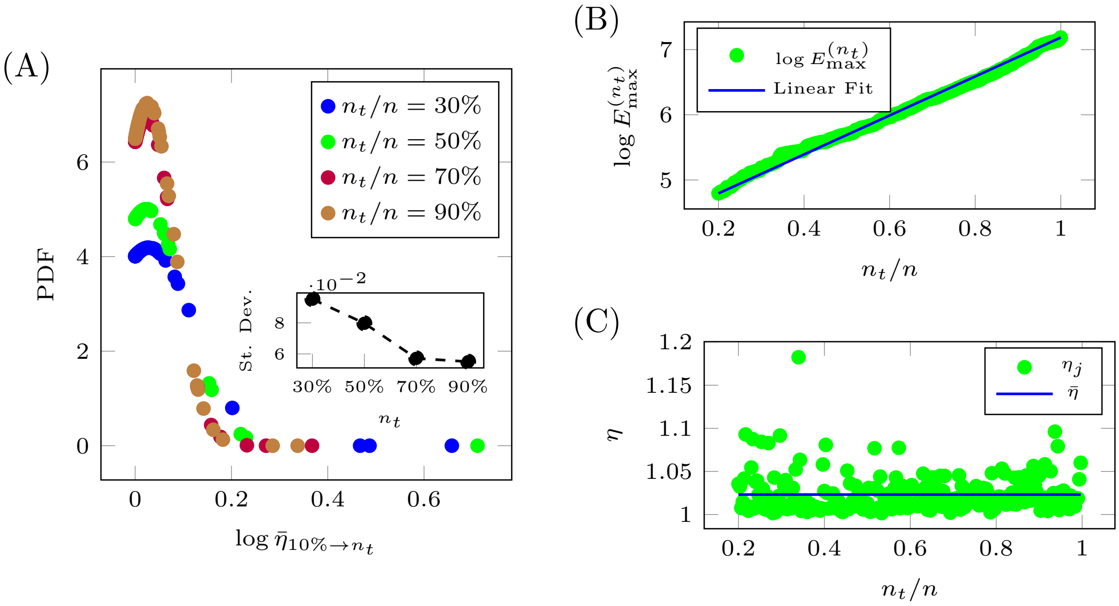


**Figure 3.2:** Example of the behavior of $\eta$. (A) The distribution of values of $\bar{\eta}_{10\% \to n_t}$ for different choices of $n_t$. We see that the distributions are fairly sharp peaked as represented by the standard deviation in the inset. (B) We choose a particular sequence $\mathcal{S}$ and compute the control energy for each set $\mathcal{S}_k$. The increase of $\log E_{\max}^{(n_t)}$ is nearly linear. (C) For each pair of adjacent sets of nodes in the sequence, $\mathcal{S}_j$ and $\mathcal{S}_{j+1}$ we compute $\eta_j = E_{\max}^{(j+1)}/E_{\max}^{(j)}$. The deviation of $\eta_j$ around the mean is small. Adapted from [1].

where the constant $C_k = \langle \log E_{\max}^{(k)} \rangle - \frac{k}{n}\eta$ is found by choosing a reasonable value of $k$.

As an example of our approximation that $\log \bar{\eta}_{k \to j}$, defined in Eq. (3.2.7), is constant, in Fig. 3.2(A) we plot the distribution of the values $\eta$ for $k = n/10$ and $n_t > k$. The probability distributions are fairly sharp peaked, as shown by the standard deviations plotted in the inset. Additionaly, for a typical sequence, we plot the worst-case control energy, $\log E_{\max}^{(n_t)}$, in Fig. 3.2(B) which we see is nearly linear. For this same sequence, in Fig. 3.2(C), we plot the ratio of each adjacent worst-case control energy values, $\eta_j = \frac{E_{\max}^{(j+1)}}{E_{\max}^{(j)}}$, which we see is nearly constant with only a small deviation. The results contained in Fig. 3.2 are for only a sample network, but they are typical of any other network we examined.

To demonstrate the linear scaling of the worst-case energy, we construct directed scale-free graphs using the static model [171] with given power-law exponent $\gamma$ for both the in-degree and out-degree distributions and with average degree $\langle \kappa \rangle$ along

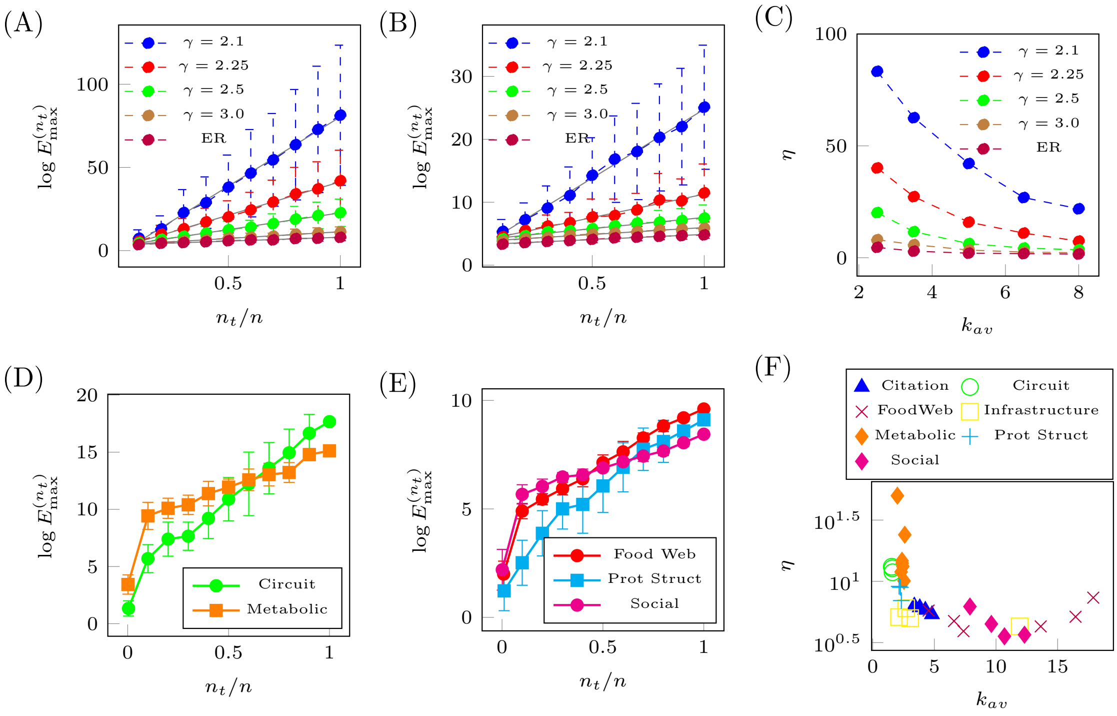


**Figure 3.3:** Worst-case energy scaling with cardinality of the target set in model networks and real networks. For the model networks, each graph has $n = 500$ and fraction of driver nodes $n_d/n = 50\%$. The worst-case control energy is computed using Eq. (3.2.5). The set of target nodes is chosen uniformly at random. The worst-case control energy for various scale-free graphs with different exponents $\gamma$ and Erdos-Renyi graphs with average degree $\langle\kappa\rangle = 2.5$ (A) and $\langle\kappa\rangle = 8.0$ (B). (C) The rate of increase of the control energy, $\eta$, is plotted as a function of the average degree for the graphs of varying heterongeneity. For the real networks, we choose the driver node fraction $n_d/n$ to be at least 50% unless it is required to be larger to guarantee controllability. (D) The worst-case control energy for the networks s420st circuit and the TM metabolic network are shown. (E) The worst-case control energy for the networks Carpinteria food web, the protein structure 1 network, and a Facebook forum network. The source of these datasets is shown in Table 3.1. (F) The values of $\eta$ for the full collection of real datasets in each of the seven classes. Adapted from [1].

with directed Erdős-Rényi graphs with average degree $\langle\kappa\rangle$. We compute the worst-case control energy over random sets of target nodes of increasing size, $n_t$. The edge weights are drawn from a uniform distribution between 0.5 and 1.5, $\mathcal{U}(0.5, 1.5)$. A self-loop is added to each node as well, $A_{j,j} = \epsilon + \delta_j$ such that $\delta_j$ is drawn from a uniform distribution $\mathcal{U}(-1, 1)$ and $\epsilon$ is chosen such that the largest real part of the eigenvalues is equal to $-1$. For average degree $\langle\kappa\rangle = 2.5$, we plot the average worst-case control energy over multiple realizations of scale-free and Erdős-Rényi graphs along with one standard deviation represented by the error bars in Fig. 3.3(A). For small power-law exponent, the degree distribution is more heterogeneous while for $\gamma \to \infty$ the degree distribution becomes Poisson. This relation explains why, as $\gamma$ grows, the behavior of $\log E_{\max}^{(n_t)}$ behaves more like the Erdős-Rényi graph. For denser graphs, $k_{av} = 8$, the same result is shown in Fig. 3.3(B) except that $\eta$, the slope of the energy curves, is much smaller. The combined relationship between control energy and a graph's average degree and degree heterogeneity is shown in Fig. 3.3(C). As the average degree grows, i.e., the graphs become more dense, the rate of increase $\eta$ decreases, while as the graphs become more heterogeneous, $\eta$ increases. This suggests that the graph easiest to control is one which (i) has hetergeneous degree distribution and (ii) is dense.

For the real datasets, we apply edge weights and self-loop values in the same manner as was done for the model networks. The linear energy scaling in the mean of the worst-case control energy is seen again for some sample networks, s420st circuit and the TM metabolic network in Fig. 3.3(D) and the Carpinteria food web, the protein structure 1 network, and a Facebook forum network in Fig. 3.3(E). The sources for these real datasets are collected in Table 3.1. The value of $\eta$ is computed for all of the real datasets considered in Fig. 3.3(F). The same behavior is shown for the model networks, that is, $\eta$ increases with decreasing average degree, i.e., with increasing sparseness.

Beyond the graph properties such as average degree and degree heterogeneity, in Fig. 3.4 we investigate the role control parameters $t_f$ and $n_d$ have on the value of

| **Type** | **Name** | $n$ | $\ell$ | $\langle\kappa rangle$ | $d$ | $\eta$ |
|---|---|---|---|---|---|---|
| Circuit | s208st[172] | 122 | 188 | 1.54 | 14 | 13.16 |
| | s420st[172] | 252 | 399 | 1.58 | 16 | 12.78 |
| | s838st[172] | 512 | 819 | 1.6 | 20 | 11.86 |
| Citation | Kohonen[173] | 3772 | 12731 | 3.38 | 9 | 6.32 |
| | SG[173] | 1024 | 4919 | 4.8 | 11 | 5.37 |
| | SW[173] | 233 | 994 | 4.27 | 7 | 5.84 |
| | Scien[173] | 2729 | 10413 | 3.82 | 13 | 6.12 |
| FoodWeb | Carpinteria[174] | 128 | 2290 | 17.89 | 6 | 7.36 |
| | Florida[173] | 128 | 2106 | 16.45 | 5 | 5.14 |
| | Grassland[173] | 113 | 832 | 7.36 | 3 | 3.92 |
| | LRL[175] | 183 | 2494 | 13.63 | 6 | 4.29 |
| | StMarks[173] | 54 | 356 | 6.59 | 7 | 4.74 |
| | Ythan[176] | 92 | 417 | 4.53 | 3 | 5.77 |
| Infrastructure | AirTrafficControl[177] | 1226 | 2615 | 2.13 | 25 | 5.11 |
| | IEEETG[178] | 118 | 358 | 3.03 | 14 | 5.01 |
| | NorthEuroGrid[179] | 236 | 640 | 2.71 | 23 | 6.04 |
| | USAir500[180] | 500 | 5960 | 11.92 | 9 | 4.29 |
| Metabolic | CE_met[43] | 1173 | 2864 | 2.44 | 30 | 13.24 |
| | EN_met[43] | 916 | 2176 | 2.38 | 28 | 14.72 |
| | SC_met[43] | 1511 | 3833 | 2.54 | 22 | 10.09 |
| | TM_met[43] | 830 | 1980 | 2.39 | 18 | 14.09 |
| | TP_met[43] | 485 | 1117 | 2.3 | 15 | 11.94 |
| | Yu-11 (New)[181, 182] | 1144 | 2293 | 2.0 | 16 | 50.83 |
| | CCSB-YI1 (New)[182, 183] | 1278 | 3450 | 2.7 | 14 | 24.06 |
| ProtStruct | prot_struct_1[172] | 95 | 213 | 2.24 | 11 | 8.95 |
| | prot_struct_2[172] | 53 | 123 | 2.32 | 6 | 7.0 |
| | prot_struct_3[172] | 99 | 212 | 2.14 | 10 | 9.1 |
| Social | EmailURV[184] | 1133 | 10903 | 9.62 | 8 | 4.49 |
| | FBForum[185] | 899 | 7089 | 7.89 | 9 | 6.23 |
| | Jazz[186] | 198 | 5484 | 27.7 | 6 | 3.36 |
| | RHS[187] | 217 | 2672 | 12.31 | 6 | 3.67 |
| | UCIrvine[188] | 1899 | 20296 | 10.69 | 8 | 3.56 |

**Table 3.1:** Real datasets and some graph properties. Taken from [1].

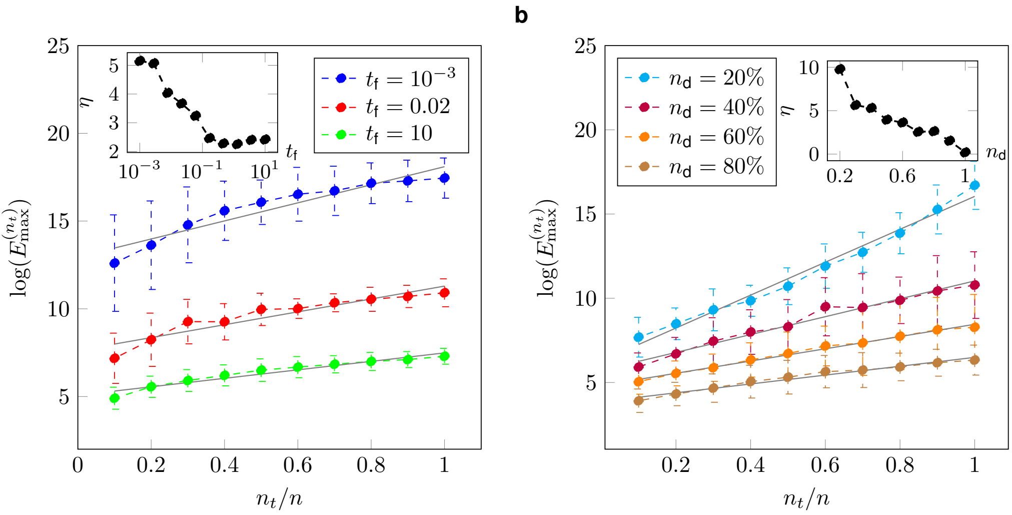


**Figure 3.4:** The effect other control parameters have on the value of $\eta$. (A) The role of the final time $t_f$ in the rate of increase of control energy. (B) The role of the driver node fraction $n_d/n$ in the rate of increase of the control energy. Adapted from [1].

$\eta$. For the case of final time, we find that for small $t_f$, the value of $\eta$ decreases as $t_f$ increases, while after $t_f$ becomes large enough, $\eta$ approaches a constant as shown in the inset in Fig. 3.4. This is expected as for Hurwitz adjacency matrix, the time-varying portion of the controllability Gramian decays to zero as $t_f$ increases.

For the driver node fraction, we see that, for the extreme case when $n_d/n = 1$, i.e., every node is a driver node, $\eta \approx 0$, that is, the control energy does not change as a function of the target node fraction $n_t/n$. On the other hand, as $n_d/n$ decreases, the rate of increase of the control energy increases as seen in the inset in Fig. 3.4(B).

The above discussions provide some insight into the exponential scaling of the control energy as a function of the number of target nodes, and how the specific growth rate is affected by both graph properties such as average degree and degree heterogeneity, as well as control parameters such as the control horizon, $t_f$, and the number of driver nodes, $n_d$. While we can prove that the control energy monotonically increases for a particular sequence of nodes added to the target set, and though we can provide some evidence, both numeric and analytic, that the increase should be exponential, any connections to the underlying structure of the graph remains to be uncovered in Chapter 5.

## 3.3 Balanced Control [2]

One of the main results of the previous section is that, as $|\mathcal{T}| = n_t$ grows linearly, the control energy grows *exponentially* which suggests that, as defined in the problem statement in the introduction, the amount of control energy required will become prohibitively large for even relatively small $n_t$. Remember that the control energy computed is the *minimum control energy* so that any other control input which attempts to satisfy the final constraints in the specified time interval will require *more* control energy.

The optimal control problem addressed in this section replaces the final constraints in Eq. (3.2.1) with final costs.

$$\begin{aligned} \min \quad & J = \frac{1-\alpha}{2}||C\boldsymbol{x}(t_f) - \boldsymbol{y}_f||_2^2 + \frac{\alpha}{2}\int_0^{t_f} ||\boldsymbol{u}(t)||_2^2 dt \\ \text{s.t.} \quad & \dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + B\boldsymbol{u}(t) \\ & \boldsymbol{x}(0) = \boldsymbol{x}_0 \end{aligned} \tag{3.3.1}$$

Details of the solution of Eq. (3.3.1) are presented in Appendix B.3 which is specialized from the general problem in Appendix B.1. For this problem, there are two costs, the control energy and the deviation that are now functions of the weighting parameter $\alpha$ and the set of target nodes, $\mathcal{T}$.

The optimal control energy and deviation are examined in detail before moving to numerical results. In this framework, the optimal control input can be simplified to,

$$\boldsymbol{u}^*(t) = -\frac{1-\alpha}{\alpha} B^T e^{A^T(t_f - t)} C^T \boldsymbol{\gamma}(t_f) \tag{3.3.2}$$

and the optimal difference at the final time is found by solving the linear system,

$$\bar{U}(\alpha)\boldsymbol{\gamma}(t_f) = \alpha\boldsymbol{\beta}, \quad \bar{U}(\alpha) = \big((1-\alpha)\bar{W} + \alpha I\big) \tag{3.3.3}$$

Using Eqs. (3.3.2) and (3.3.3) in the definition of the control energy, $E = \int_0^{t_f} ||\boldsymbol{u}(t)||_2^2 dt$, the optimal control energy can be expressed as a quadratic form that is a function of the weight $\alpha$.

$$E(\alpha) = (1-\alpha)^2 \boldsymbol{\beta}^T \bar{U}^{-1}(\alpha) \bar{W} \bar{U}^{-1}(\alpha) \boldsymbol{\beta} \tag{3.3.4}$$

The worst-case control energy in this case, which is equal to the largest eigenvalue of the matrix $\bar{U}^{-1}(\alpha)\bar{W}\bar{U}^{-1}(\alpha)$, can be computed as a maximization of a function over the eigenvalues of the output controllability Gramian, Spec($\bar{W}$),

$$E_{\max}(\alpha) = \max_{\mu_k \in \mathrm{Spec}(W_{\mathcal{T}'})} \frac{(1-\alpha)^2 \mu_k}{(\alpha + (1-\alpha)\mu_k)^2} \tag{3.3.5}$$

An example of the behavior of $E_{\max}(\alpha)$ is shown in Fig. 3.5(A) in black, found numerically by computing the largest eigenvalue of the matrix in Eq. (3.3.4). The function being maximized in Eq. (3.3.5) is also plotted for each of the five eigenvalues in the sample system's controllability Gramian $\bar{W}$. We see that the worst-case control energy $E_{\max}(\alpha)$ is defined by each of the eigenvalues $\mu_k$, for some interval of $\alpha$. We can find the maximum in Eq. (3.3.5) by letting $\mu \in (0, \infty)$ be a continuous variable, differentiating with respect to $\mu$, and setting the result equal to zero,

$$\frac{\partial}{\partial \mu} \frac{(1-\alpha)^2 \mu}{(\alpha + (1-\alpha)\mu)^2} = -(1-\alpha)^2 \frac{(1-\alpha)x - \alpha}{(1-\alpha)x + \alpha)^3} = 0 \tag{3.3.6}$$

The value of $\mu$ that maximizes the control energy $E$ (and a solution of Eq. (3.3.6)) is $\mu = \frac{\alpha}{1-\alpha}$. The worst-case control energy, $E_{\max}$, can be determined from the two eigenvalues closest to $\frac{\alpha}{1-\alpha}$, one from above and the other one from below. We can see that if $\alpha = 0$, the optimal $\mu = 0$, so we get the worst-case control energy we saw in the previous section, $E_{\max} = \frac{1}{\mu_{\min}}$. On the other hand, if $\alpha = 1$, the worst-case control energy is $E_{\max} = 0$. The behavior of $E_{\max}$ for intermediate values of $\alpha$ is quite complicated as $E_{\max}$ is non-differentiable due to the transitions from eigenvalue to eigenvalue.

In this framework, we must also concern ourselves with the *accuracy* (or deviation)

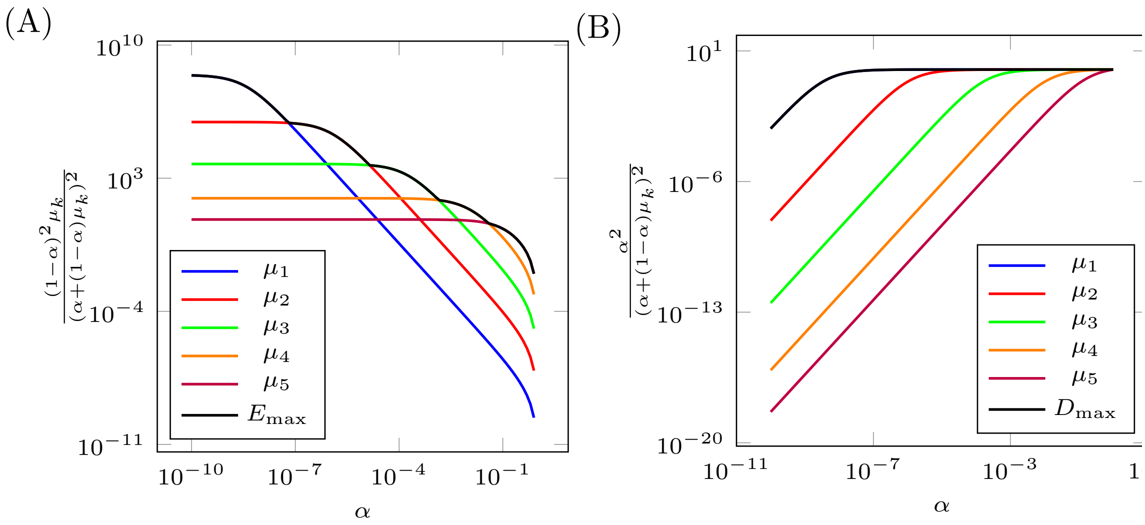


**Figure 3.5:** Relationship between worst-case control energy $E_{\max}$ and worst-case control deviation $D_{\max}$ and the spectrum of $\bar{W}$. The function to maximimze over the spectrum of $\bar{W}$ is plotted for each eigenvalue of $\mu_k$. The black line is the maximum eigenvalue of $\bar{U}^{-1}(\alpha)\bar{W}\bar{U}^{-1}(\alpha)$ found numerically which is seen to coincide with the maximum of the function in Eq. (3.3.5) over the eigenvalues $\mu_k$. In (B), the maximum deviation is plotted in black with the function in Eq. (3.3.8). The function being maximized in Eq. (3.3.8) is also plotted for each of the eigenvalues of $\bar{W}$ where it is clear it is maximized for $\mu_{\min}$.

of the control action, defined previously as,

$$D = \alpha^2 \boldsymbol{\beta}^T \bar{U}^{-1}(\alpha)\bar{U}^{-1}(\alpha)\boldsymbol{\beta} \tag{3.3.7}$$

The worst case deviation can be found in a similar fashion as for the worst case control energy which is equal to the maximum eigenvalue of $\bar{U}^{-1}(\alpha)\bar{U}^{-1}(\alpha)$.

$$D_{\max} = \max_{\mu_k \in \text{Spec}(W_{\mathcal{T}'})} \frac{\alpha^2}{(\alpha + (1-\alpha)\mu_k)^2} = \frac{\alpha^2}{(\alpha + (1-\alpha)\mu_{\min})^2} \tag{3.3.8}$$

The maximum deviation for an example graph is shown in Fig. 3.5(B) where the black line is the numerically found largest eigenvalue of $\bar{U}^{-1}(\alpha)\bar{U}^{-1}(\alpha)$ while the function to maximize in Eq. (3.3.8) is shown for each eigenvalue in the other colors. We see that, as prediced in Eq. (3.3.8), only the smallest eigenvalue of $\bar{W}$ dictates the worst-case control energy. When $\alpha = 1$, the deviation reaches some finite value, $D_{\max} \sim 1$ while when $\alpha = 0$, the deviation (assuming output controllability) goes to zero.

The optimal cost can be written in terms of a weighted sum of the control energy

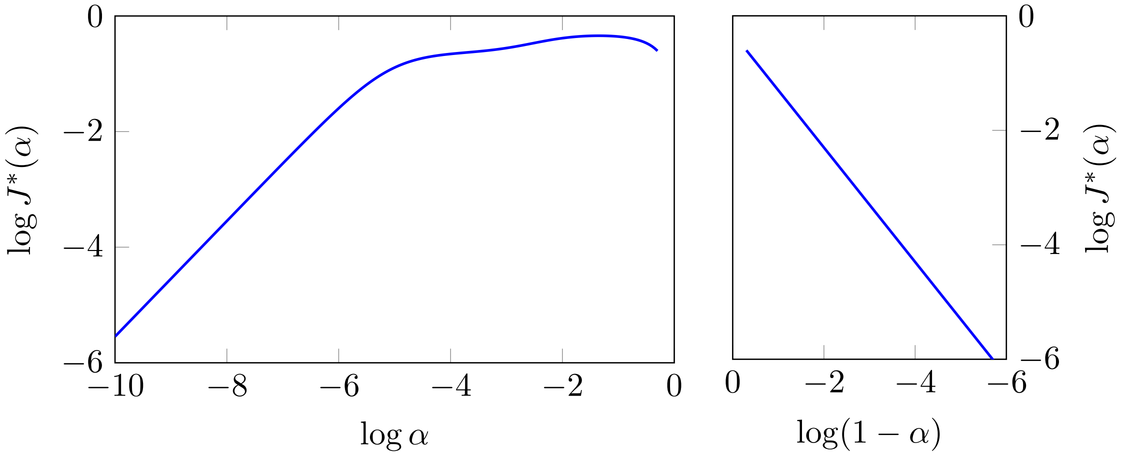


**Figure 3.6:** An example of the optimal cost in Eq. (3.3.9). In the left plot, the optimal cost in the small $\alpha \to 0$ regime decays as $J^* \sim -\frac{\alpha}{\mu_{\text{min}}}$ because $E \to \frac{1}{\mu_{\text{min}}}$ and $D \to 0$. In the right plot, the optimal cost in the $\alpha \to 1$ regime decays as $J^* \sim -(1-\alpha)$

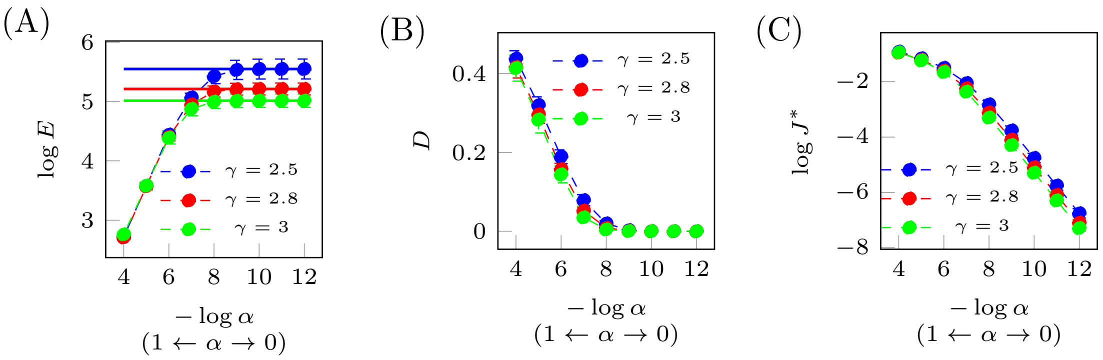


**Figure 3.7:** The trade-off between control energy and deviation. (A) The control energy as a function of $-\log\alpha$ which we see approaches a constant as $\alpha \to 0$ while it approaches zero as $\alpha \to 1$. (B) The deviation as a function of $-\log\alpha$. When $\alpha \to 0$, the deviation $D \to 0$. On the other as $\alpha \to 1$, the deviation increases linearly. The cost function approaches zero as $\alpha \to 0$ as expected. Adapted from [2].

and the deviation.

$$J^* = \frac{1-\alpha}{2}D + \frac{\alpha}{2}E = J_1^* + J_2^* \tag{3.3.9}$$

The control energy $E$ is defined in Eq. (3.3.4) and the deviation $D$ is defined in Eq. (3.3.7). For small $\alpha \to 0$, the deviation $D \to 0$ while the control energy $E$ approaches a constant. For $\alpha \to 1$, the deviation now approaches a constant while the control energy approaches zero. Taken together, the total cost $J \to 0$ for both $\alpha \to 0$ and $\alpha \to 1$, which can be seen in Fig. 3.6. The control energy, deviation, and cost in the small $\alpha$ regime for a variety of scale-free model networks is shown in Fig. 3.7. We

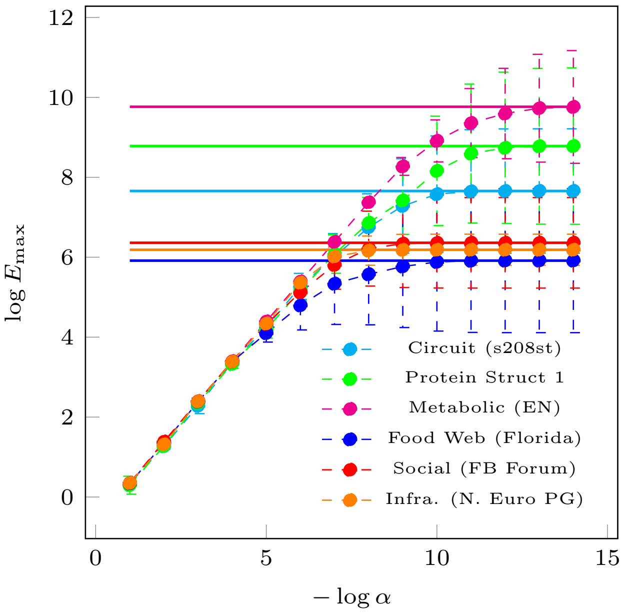


**Figure 3.8:** The control energy of a variety of real datasets whose sources can be found in Table 3.1. The behavior of the control energy in the small $\alpha$ regime is seen approaching a constant as $\alpha \to 0$ and decays to 0 nearly invariant with respect to the system as $\alpha \to 1$. Adapted from [2].

see the expected behavior in Fig. 3.7(A), as $\alpha \to 0$, the control energy approaches a constant, equal to $\frac{1}{\mu_{\min}}$. We also see from Fig. 3.7(A) that as $\alpha \to 1$, the rate of decay of the control energy is invariant with respect to the type of model graph. This behavior is expected as, if $\alpha \gg (1-\alpha)\mu_{\max}$, $E_{\max} \sim \frac{(1-\alpha)^2\mu_{\max}}{\alpha^2}$, where $\mu_{\max}$ does not vary greatly across graphs. Similarly, the deviation in Fig. 3.7(B) is shown to approach zero as $\alpha \to 0$. Finally, the optimal cost is shown to decay as $\frac{\alpha}{\mu_{\min}}$ in Fig. 3.7(C). The control energy for a variety of the systems from the literature is shown in Fig. 3.8. Again, despite the graph topology, the control energy $E_{\max}(\alpha)$ is still seen to approach the constant $\frac{1}{\mu_{\min}}$ as $\alpha \to 0$.

The framework of balanced control is specialized in the next section to the case where the state matrix is not known exactly, but rather exists as one of $N$ possible realizations.

## 3.4 Controlling Network Ensembles

The results of the previous section is used in this section to solve the problem of *controlling network ensembles.* By a network ensemble, we mean a family of weighted networks which satisfy a particular set of constraints [189, 190]. Each network has the same number of nodes, $n$. From this ensemble, we sample $N$ adjacency matrices, which we call $\mathcal{A} = \{A^{(j)} | j = 0, \ldots, N-1\}$. For each realization, we assume that $B$ and $C$ are identical as they are typically designed. This problem is concerned with finding an optimal control capable of driving each system realization to some final output approximately.

$$\begin{aligned} \min \quad & J = \frac{1-\alpha}{2} \sum_{k=0}^{N-1} ||C\boldsymbol{x}_k(t_f) - \boldsymbol{y}_f||_2^2 + \frac{\alpha}{2} \int_0^{t_f} ||\boldsymbol{u}(t)||_2^2 dt, \quad \alpha \in (0,1) \\ \text{s.t.} \quad & \dot{\boldsymbol{x}}_k(t) = A^{(k)} \boldsymbol{x}_k(t) + B\boldsymbol{u}(t), \quad k = 0, \ldots, N-1 \\ & \boldsymbol{x}_k(0) = \boldsymbol{x}_0 \end{aligned} \tag{3.4.1}$$

The optimal control problem in Eq. (3.4.1) can be mapped to the general problem presented at the beginning of this chapter in Eq. (3.3.1) by defining the composite system,

$$\begin{bmatrix} \dot{\boldsymbol{x}}_0(t) \\ \dot{\boldsymbol{x}}_1(t) \\ \vdots \\ \dot{\boldsymbol{x}}_{N-1}(t) \end{bmatrix} = \begin{bmatrix} A^{(0)} & O_n & \cdots & O_n \\ O_n & A^{(1)} & \cdots & O_n \\ \vdots & \vdots & \ddots & \vdots \\ O_n & O_n & \cdots & A^{(N-1)} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_0(t) \\ \boldsymbol{x}_1(t) \\ \vdots \\ \boldsymbol{x}_{N-1}(t) \end{bmatrix} + \begin{bmatrix} B \\ B \\ \vdots \\ B \end{bmatrix} \boldsymbol{u}(t) \tag{3.4.2}$$

as well as setting $\boldsymbol{\gamma} = \begin{bmatrix} (C\boldsymbol{x}_0(t_f) - \boldsymbol{y}_f)^T & \cdots & (C\boldsymbol{x}_{N-1}(t_f) - \boldsymbol{y}_f)^T \end{bmatrix}^T$. Rather than deriving the solution in terms of the composite system in Eq. (3.4.2), it is more enlightening to perform the derivation in terms of the individual system highlighting the special structure present. An example of the type of network ensemble of interest is shown in Fig. 3.9(a) where the structure of the network is known but each edge weight is drawn from a distribution depending on how much information is known

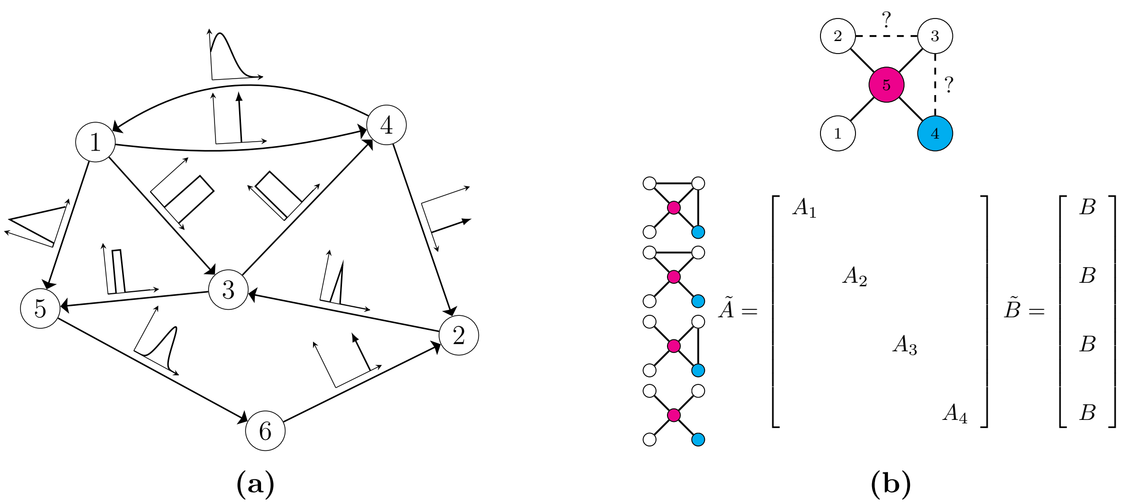


**Figure 3.9:** Description of the method for controlling network ensembles. (a) A network where the weight associated with each edge is only known to be drawn from a distribution. (b) A small graph with $n = 5$ nodes, $n_d = 1$ driver node and $n_t = 1$ target node. Two of the edges may or may not exist, leading to $N = 4$ network configurations, shown on the left. The composite system is a block diagonal matrix shown as $\bar{A}$, and the composite input matrix has 4 copies of $B$ stacked vertically. Similarly, not shown is the composite output matrix which also has 4 copies of $C$ lined up horiztonally.

about it. In Fig. 3.9(b), another type of network ensemble is shown, where in this case two edges may or may not exist in the actual network. The composite system is shown where the matrix $\bar{A}$ is block diagonal and $\bar{B}$ has $N = 4$ copies of $B$ stacked on top of each other.

The three quantities of interest to characterize the solution are the deviation, $D_N(\alpha)$, the control energy $E_N(\alpha)$, and the total cost, $J_N(\alpha)$, repeated here for clarity.

$$\begin{aligned} D_N(\alpha) &= \sum_{k=0}^{N-1} ||\boldsymbol{\gamma}_k(t_f)||_2^2 \\ E_N(\alpha) &= \int_0^{t_f} ||\boldsymbol{u}(t)||_2^2 dt \\ J_N(\alpha) &= \frac{1-\alpha}{2} D_N(\alpha) + \frac{\alpha}{2} E_N(\alpha) \end{aligned} \tag{3.4.3}$$

The vectors $\boldsymbol{\gamma}_k(t_f) = C\boldsymbol{x}_k(t_f) - \boldsymbol{y}_f$ represent the difference between the actual output at the final time and the desired final output of realization $k = 0, \dots, N-1$. The solution is characterized in terms of the composite output controllability Gramian

(COCG), a special version of Eq. (2.3.7) when the matrix $A$ is diagonal.

$$\bar{W} = \begin{bmatrix} CW_{0,0}(t_f)C^T & CW_{0,1}(t_f)C^T & \cdots & CW_{0,N-1}(t_f)C^T \\ CW_{1,0}(t_f)C^T & CW_{1,1}(t_f)C^T & \cdots & CW_{1,N-1}(t_f)C^T \\ \vdots & \vdots & \ddots & \vdots \\ CW_{N-1,0}(t_f)C^T & CW_{N-1,1}(t_f)C^T & \cdots & CW_{N-1,N-1}(t_f)C^T \end{bmatrix} \tag{3.4.4}$$

where each block is the solution of the differential Sylvester equation,

$$\dot{W}_{j,k}(t) = A_j W_{j,k}(t) + W_{j,k}(t)A_k^T + BB^T, \quad W_{j,k}(0) = O_n \tag{3.4.5}$$

From Eq. (3.4.5), it is clear that $W_{j,k}(t) = W_{k,j}^T(t)$ so the composite output controllability Gramian is a symmetric matrix. Also, when $j = k$, the differential Sylvester equation becomes the differential Lyapunov equation in Eq. (2.3.8). Further details are discussed in Appendix B.4. The composite output controllability Gramian is a symmetric semi-positive definite matrix of dimension $Nn_t$. We denote the eigenvalues and eigenvectors of $\bar{W}$ as $\mu_k$ and $\boldsymbol{\xi}_k$, respectively, such that $\mu_k \geq \mu_{k+1}$. The matrix that will appear in the three costs in Eq. (3.4.3) after applying the optimal control and the optimal vector $\boldsymbol{\gamma}$ is a scaled and shifted version of the COCG, denoted

$$\bar{U}(\alpha) = (\alpha I_{Nn_t} + (1-\alpha)\bar{W}) \tag{3.4.6}$$

which is *similar* to the original COCG, i.e., it shares its eigenvectors with $\bar{W}$ but for each eigenvalue $\mu_k$, there is an associated eigenvalue of $\bar{U}(\alpha)$ equal to $\nu_k = \alpha + (1-\alpha)\mu_k$. Using Eqs. (3.4.4) and (3.4.6), the cost functions in Eq. (3.4.3) can be expressed as quadratic forms in terms of similar matrices.

$$\begin{aligned} D_N(\alpha) &= \alpha^2 \boldsymbol{\beta}^T \bar{U}^{-1}(\alpha)\bar{U}^{-1}(\alpha)\boldsymbol{\beta} \\ E_N(\alpha) &= (1-\alpha)^2 \boldsymbol{\beta}^T \bar{U}^{-1}(\alpha)\bar{W}\bar{U}^{-1}(\alpha)\boldsymbol{\beta} \\ J_N(\alpha) &= \frac{(1-\alpha)\alpha}{2}\boldsymbol{\beta}^T \bar{U}^{-1}(\alpha)\boldsymbol{\beta} \end{aligned} \tag{3.4.7}$$

In the following results, we use the inner products of the eigenvectors and the control maneuver, which we denote $\theta_k = \boldsymbol{\beta}^T \boldsymbol{\xi}_k$. From extensive numerical studies, it appears that the following two relationships hold approximately which we will treat as assumptions in the subsequent derivations.

$$\begin{aligned} &\textbf{Assumption 1:} \quad \mu_k \sim \mu_0 r_1^k, && \mu_0 \sim c_1 N n_t \\ &\textbf{Assumption 2:} \quad \theta_k^2 \sim \max\{\theta_0^2 r_2^k, \theta_c^2\}, && \theta_0^2 \sim c_2 N n_t \end{aligned} \tag{3.4.8}$$

In appendix B.4, the assumptions in Eq. (3.4.8) are applied to the summation form of Eq. (3.4.7). As the costs currently stand in Eq. (3.4.7), for constant $\alpha$, the control energy can be shown to grow logarithmically with $N$ so that $\lim\limits_{N\to\infty} E_N(\alpha)$ is not finite. Instead, we design $\alpha$ grow with $N$. The form of $\alpha$ that is imposed to incorporate the growth of $N$ is,

$$\alpha(b) = \frac{Nn_t}{Nn_t + b}, \quad b > 0. \tag{3.4.9}$$

Note that for $b = 0$, $\alpha(0) = 1$, and the original problem in Eq. (3.4.1) becomes one in which we minimize the control energy with no constraint or cost applied to the final output. This problem has the trivial solution that $\boldsymbol{u}^*(t) = \mathbf{0}$ which can be deduced by noting that any time trajectory of the states is not any better than any other in regards to the cost function. At the other extreme, if we let $b \to \infty$ faster than $N$ is growing, then $\alpha(b) \to 0$ and the problem approaches the minimum energy problem which will be shown in the following discussions.

In Appendix B.4, the deviation, control energy, and total cost in Eq. (3.4.7), are rewritten as sums over the eigenvalues of $\bar{W}$, $\mu_k$, $k = 0, \ldots, Nn_t - 1$, and the inner products of the composite control maneuver and each of the eigenvectors, $\theta_k = \boldsymbol{\beta}^T \boldsymbol{\xi}_k$. After applying the assumptions in Eq. (3.4.8) and the form of $\alpha$ in Eq. (3.4.9) the costs becomes,

$$\begin{aligned}
D_N(b)/N &\approx c_2 n_t \sum_{k=0}^{\bar{k}} \frac{r_2^k}{(1+bc_1r_1^k)^2} + \frac{\theta_c^2}{N} \sum_{k=\bar{k}+1}^{Nn_t-1} \frac{1}{(1+bc_1r_1^k)^2} \\
E_N(b) &\approx b^2 c_1 c_2 \sum_{k=0}^{\bar{k}} \frac{(r_1r_2)^k}{(1+bc_1r_1^k)^2} + \frac{b^2c_1\theta_c^2}{Nn_t} \sum_{k=\bar{k}+1}^{Nn_t-1} \frac{r_1^k}{(1+bc_1r_1^k)^2} \qquad (3.4.10) \\
J_N(b) &\approx \frac{bc_2Nn_t}{2(Nn_t+b)} \sum_{k=0}^{\bar{k}} \frac{r_2^k}{1+bc_1r_1^k} + \frac{b\theta_c^2}{2(Nn_t+b)} \sum_{k=\bar{k}+1}^{Nn_t-1} \frac{1}{1+bc_1r_1^k}
\end{aligned}$$

Each of the summations in Eq. (3.4.10) can be upper bounded by convergent geometric series. The upper bounds found in Appendix B.4 are all shown to approach constant values in the $N \to \infty$ limit,

$$\begin{aligned}
\lim_{N\to\infty} D_N(b)/N &\leq \frac{c_2n_t}{1-r_2} + \theta_c^2 n_t \\
\lim_{N\to\infty} E_N(b) &\leq \frac{b^2c_1c_2}{1-r_1r_2} \qquad (3.4.11) \\
\lim_{N\to\infty} J_N(b) &\leq \frac{bc_2}{2(1-r_2)} + \frac{b\theta_c^2}{2}
\end{aligned}$$

From Eq. (3.4.10), one can choose $b$ to set the desired average deviation even in the $N \to \infty$ limit, the control energy remains finite according to Eq. (3.4.11).

Three examples of types of uncertain networks follow where the assumptions in Eq. (3.4.8) are shown to hold at least in the average over network ensembles and the behavior of the costs as functions of both $N$ and $b$ is investigated. In Fig. 3.10, we consider a network ensemble that conists of $N$ unidirectional chain graphs $n$ nodes long where each chain has a common self loop, $p_k$, and a common edge weight, $s_k$, for $k = 0, \dots, N-1$. There is a single driver node $\mathcal{D} = \{v_0\}$, and the target set is some subset of the remaining nodes. A diagram of the composite system is shown in Fig. 3.10(A) where each chain is $n = 4$ nodes long from $v_0$ to $v_3$, and the single control input $u$ is attached to each of the $N$ copies of node $v_0$. The regulation parameters, $p_k$, and the edge weights, $s_k$, $k = 0, \dots, N-1$, are drawn from uniform distributions $\mathcal{U}(2,4)$ and $\mathcal{U}(0.5, 1.5)$, respectively. For target set $\mathcal{T} = \{v_1\}$, the largest eigenvalue, $\mu_0$, and the associated eigenvector value, $\theta_0^2$, of 10 realizations of $N$ unidirectional

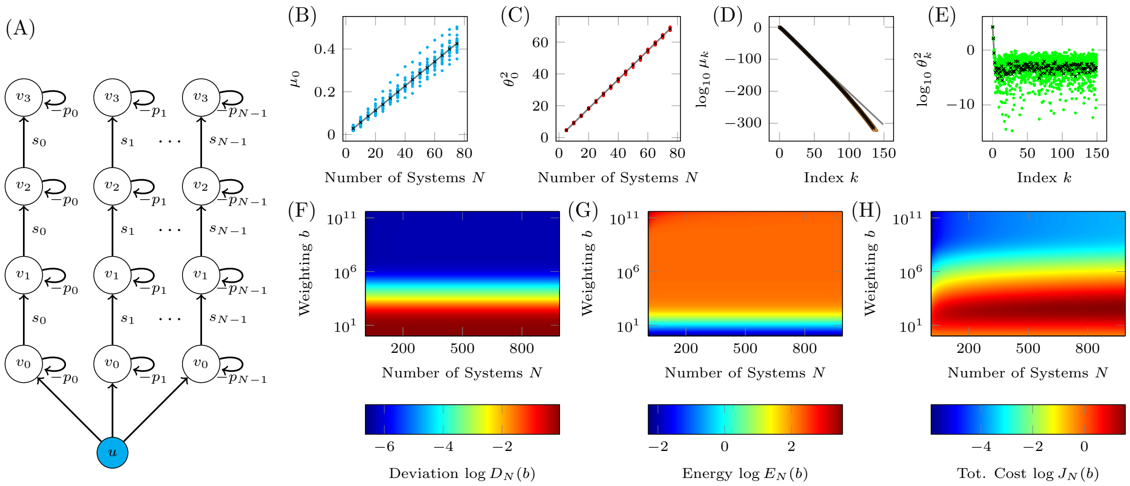
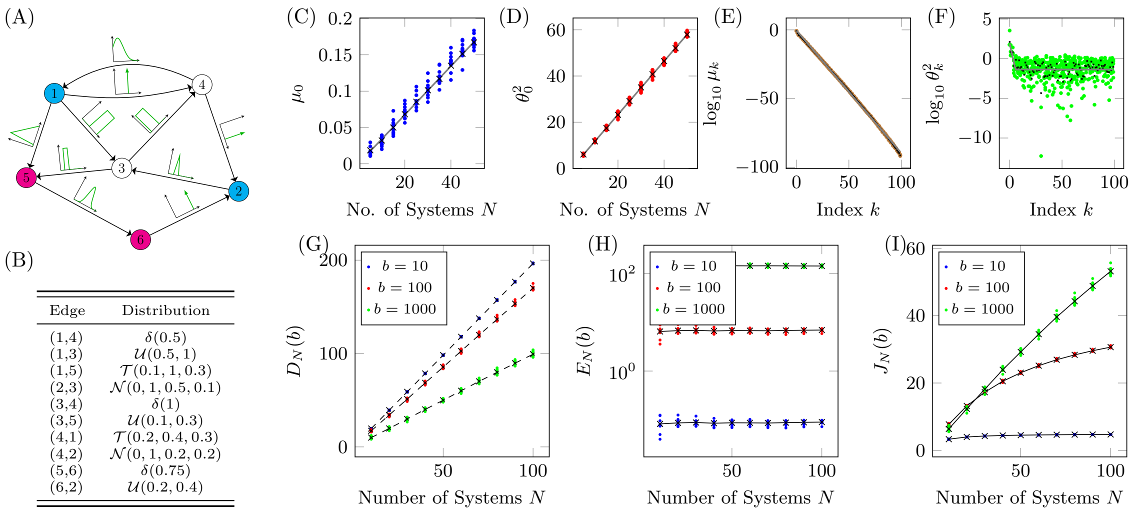


**Figure 3.10:** Control of an ensemble of unidirectional chain graphs of length $n = 4$. (A) A diagram of the composite network with one global input applied to each of the $N$ copies of node $v_0$. Each realization has uniform regulation parameter $p_k$ and uniform edge weight $s_k$, drawn from distributions. For the remaining simulations, we choose the target node set $\mathcal{T} = \{v_1\}$. The largest eigenvalue $\mu_0$ and corresponding inner product $\theta_0^2$ are shown in (B) and (C), respectively, where we see the hypothesized linear growth in the mean after averaging over 10 realizations. For $N = 150$, we see the eigenvalues decay exponentially in (D) while the inner products $\theta_k^2$ initially decay exponentially before saturating to approximately a constant value in (E). The three costs are shown as functions of both $N$ and $\alpha$ in (F), (G), and (H), where little change is seen along the $N$ axis, but there is clearly a dynamic range along the $b$ axis, outside of which each cost is seen to change very little with exponential growth of the parameter $b$.

chain graphs for $N5,\ldots,75$, are shown in Figs. 3.10(B) and 3.10(C), respectively, where the linear growth as a function of $N$ is seen in the average (the black marks). For $N = 150$, the eigenvalues are shown to decay exponentially in Fig. 3.10(D) while the inner products $\theta_k^2$ are seen to initially decay exponentially before saturating to something approximately constant in Fig. 3.10(E). With the results shown in Figs. 3.10(B)-(E), there is good evidence that assumptions in Eq. (3.4.8) hold and so we expect that the costs in Eq. (3.4.10) approach a constant value in the $N \to \infty$ limit. These costs are shown in Figs. 3.10(F)-(H) as a function of both $N$ and $b$. Along the $N$ axis, we see the colors are constant, while along the $b$ axis there is a dynamic range for the average deviation and the control energy where changes in $b$ correspond to proportional changes in the costs. Outside of this range though, exponential changes in $b$ lead to only subtle changes in these costs. The total cost in Fig. 3.10(H) though changes more uniformly than the other two costs.

The next model we consider is a small network with edge weights drawn from distributions whose shape is shown along each edge in the diagram in Fig. 3.11(A). The precise distributions for each edge are listed in the table in Fig. 3.11(B) where $\delta(a)$ represents those edge weights whose values are known, $\mathcal{U}(a,b)$ is a uniform distribution so each edge weight $a \leq s \leq b$, $\mathcal{T}(a,b,c)$ is a triangular distribution with peak at $c$ and edge weights $a \leq s \leq b$, and $\mathcal{U}(a,b,\mu,\sigma)$ is the truncated normal distribution with mean $\mu$, standard deviation $\sigma$, and finite support between $a$ and $b$. The driver nodes are chosen to be $\mathcal{D} = \{1,2\}$ while the target nodes are chosen to be $\mathcal{T} = \{5,6\}$. The largest eigenvalues, $\mu_0$, and corresponding values $\theta_0^2$, for $N = 2,\ldots,25$, are shown in Figs. 3.11(C) and 3.11(D), where in the average over 25 realizations, are seen to grow linearly. The eigenvalues, $\mu_k$ for $N = 50$ (so that $Nn_t = 100$), is shown in Fig. 3.11(E) where the exponential decay is apparent and the inner product values $(\boldsymbol{\beta}^T\boldsymbol{\xi}_k)^2 = \theta_k^2$ is seen to decay exponentially for the first few indices $k$ before saturating to a constant value. Again, the assumptions in Eq. (3.4.8) are shown to hold for this network and so we expect the average deviation, the control energy, and the total cost to all approach constant values in the limit of large $N$. The total deviation for three values of $b$ is



| Edge | Distribution |
|---|---|
| (1,4) | $\delta(0.5)$ |
| (1,3) | $\mathcal{U}(0.5, 1)$ |
| (1,5) | $\mathcal{T}(0.1, 1, 0.3)$ |
| (2,3) | $\mathcal{N}(0, 1, 0.5, 0.1)$ |
| (3,4) | $\delta(1)$ |
| (3,5) | $\mathcal{U}(0.1, 0.3)$ |
| (4,1) | $\mathcal{T}(0.2, 0.4, 0.3)$ |
| (4,2) | $\mathcal{N}(0, 1, 0.2, 0.2)$ |
| (5,6) | $\delta(0.75)$ |
| (6,2) | $\mathcal{U}(0.2, 0.4)$ |

**Figure 3.11:** Controlling a network with uncertain edge weights. (A) The network from Fig. 3.9 is shown again with nodes colored according to their role as driver nodes (blue) or target nodes (pink). (B) The distributions used to draw edge weights for each realization; (i) known value $\delta(a)$, uniform distribution $\mathcal{U}(a, b)$, triangular distribution $\mathcal{T}(a, b, c)$, or truncated normal distribution $\mathcal{N}(\mu, \sigma, a, b)$. (C) The largest eigenvalue $\mu_0$ and (D) the largest inner product $\theta_0^2$ where the black marks are the average taken over 25 realizations. for $N = 50$ (so $Nn_t = 100$), the eigevalues $\mu_k$ are seen to decay exponentially in (E) and the inner products $\theta_k^2$ are seen to at first decay exponentially before saturating to an approximately constant value. For three values of $b$, the three costs are plotted as a function of $N$ in (G), (H), and (I).

shown in Fig. 3.11(G) which all grow linearly with $N$ so that $D_N(b)/N$ is constant. As $b$ grows, the slopes of the curves decrease. In Fig. 3.11(H), the control energy, even for moderate values of $N$, for the three values of $b$ examined, the control energy is approximately constant. Finally, the total cost is shown in Fig. 3.11(I) which is also shown to approach a constant value, but more slowly as the multiplier in front of the total cost is proportional to $\alpha(b)$.

The third example samples state matrices $A$ that are linearizations of a nonlinear dynamical system when some of the parameters are unknown. The dynamical system examined is the autophagy model discussed in Sec. 2.4.3 with the system of differential equations written in Eq. (2.4.11). All of the parameters are chosen from Table A.1 except for the parameters $C_{Nu}$ and $C_{En}$ which are chosen from the uniform distribution $\mathcal{U}(0.1, 0.6)$. From Fig. 2.5(B), we see that the system in Eq. (2.4.11) has a single stable fixed point. The Jacobian of the dynamical system is evaluated about this stable fixed point for each choice of $C_{Nu}$ and $C_{En}$. The edge weights are the partial

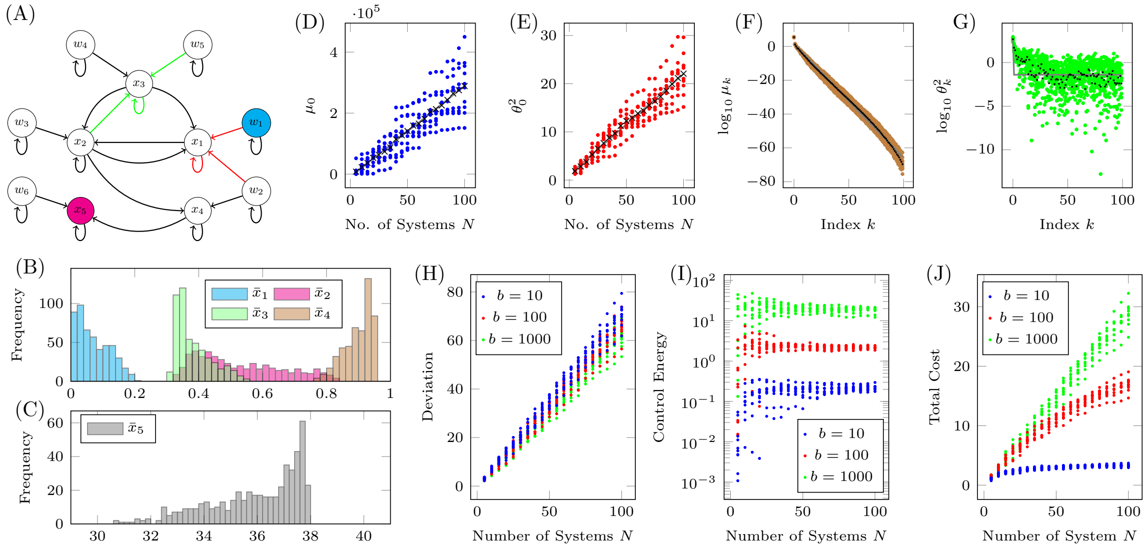


**Figure 3.12:** Control of the linearized autophagy system near its stable fixed point. (A) The diagram of its dynamical system with edges colored red if $C_{Nu}$ appears in their expressions explicitly or green if $C_{En}$ appears in their expression explicitly. We choose a single drug therapy, 1, so that only node $w_1$ is a driver node, and we are only attempting to control the number of AVs, represented by $x_5$, making it the only target node. (B) and (C) Distributions of the fixed point values of each of the 5 states when $C_{Nu}$ and $C_{En}$ are drawn from the uniform distribution $\mathcal{U}(0.1, 0.6)$. (D) The largest eigenvalues of $\mu_0$ of the Jacobian of the autophagy dynamics where the average is seen to grow linearly (albeit at a much faster rate, note the multiplier $10^5$) and the inner products $\theta_0^2$ are shown to grow linearly in (E). For $N = 100$, the eigenvalues are shown in (F) which decay exponentially and the inner products $\theta_k^2$ are shown in (G). For three values of $b$, the deviation is shown growing linearly in (H), the control energy is shown approaching constant values in (I) and the total cost is shown to also approach constants (although quite slowly for large $b$) in (J).

derivatives evaluated at the fixed point $\bar{\boldsymbol{x}}$ where $\boldsymbol{f}(\bar{\boldsymbol{x}}) = \mathbf{0}$.

$$w_{j,k} = \left.\frac{\partial f_j}{\partial x_k}\right|_{\boldsymbol{x}=\bar{\boldsymbol{x}}} \tag{3.4.12}$$

The dynamical system's network respresetation is shown in Fig. 3.12(A) where red edges are those whose equations described by Eq. (3.4.12) have $C_{Nu}$ appear explicitly, and likewise green edges are those which have $C_{En}$ appear explicitly. Black edges have neither $C_{Nu}$ nor $C_{En}$ appear explicitly, but likely are affected implicitly by different choices of $C_{Nu}$ and $C_{En}$. For 500 random pairs of $C_{En}$ and $C_{Nu}$, the fixed point $\bar{\boldsymbol{x}}$ is computed using a Newton method implemented in Kinsol [191]. The distributions of the steady states values of the five states that represent production of autophagic vesicles is shown in Figs. 3.12(B) and 3.12(C). By construction of the model, the first four states are distributed between 0 and 1, while the number of autophagic vesicles, shown in Fig. 3.12(C) is distributed with a single peak just below 38 and a tail to about 30. As in the previous two examples, the largest eigenvalues $\mu_0$, and associated inner product $\theta_0^2$, are shown in Figs. 3.12(D) and 3.12(E), repspectively, where in the average over 10 realizations for $N = 5, \ldots, 100$, they are seen to grow linearly. Note the much larger slope of $\mu_0$ compared to the previous two examples. The fact $c_1$ is orders of magnitude larger does not affect the validity of the bounding performed in Eq. (3.4.11). For 10 realizations of $N = 100$ selections of $C_{En}$ and $C_{Nu}$, the eigenvalues $\mu_k$ and inner products $\theta_k^2$ are shown in Figs. 3.12(F) and 3.12(G), respectively, where the exponential decay of $\mu_k$ and the transition between exponential decay to a constant value can be seen. Thus, with the numerical results in Figs. 3.12(D)-(G), there is good evidence that the assumptions in Eq. (3.4.8) hold, and we can be sure that $D_N(b)/N$, $E_N(b)$ and $J_N(b)$ all approach constant values in the $N \to \infty$ limit. The total deviation is shown in Fig. 3.12(H) which, for the three values of $b$ selected, is seen to grow linearly with $N$ so that $D_N(b)/N$ is a constant. In Fig. 3.12(I), the control energy is shown to be nearly constant for even moderate values of $N$. Similarly, the total cost in Fig. 3.12(J) is shown to approach constant values proportional to the form of $\alpha(b)$ in Eq. (3.4.9).

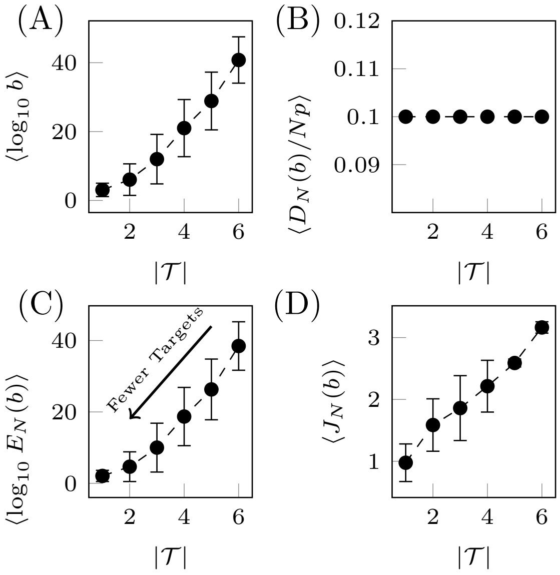


**Figure 3.13:** An example of the behavior of the costs as a function of the number of targets. For the network shown in Fig. 3.11(A), we leave $\mathcal{D} = \{1, 2\}$ but we examine all possible sets of target nodes, $\mathcal{T} \in 2^{\mathcal{V}}$. For each target set, 10 realizations of the network are taken and the values of weight $b$, average deviation $D_N(b)/Nn_t$, control energy $E_N(b)$, and total cost $J_N(b)$, are shown in panels (A), (B), (C), and (D) repsectively. Error bars represent one standard deviation.

As was demonstrated in Secs. 3.2 and 3.3, the control energy decays exponentially with linear reduction in the number of target nodes. This is investigated for the network ensemble case as well. In order to do this though, we would like to find $b$ such that $\frac{D_N(b)}{Nn_t} = \bar{D}$, a constant value, so the comparison will be fair. To find $b$, we use a bisection method as $D_N(b)$ is a monotonically decreasing function in $b$. The network examined to generate the numerical results in Fig. 3.13 is the same as the one shown in Fig. 3.11(A). The set of driver nodes remains the same, $\mathcal{D} = \{1, 2\}$, while each possible set of target nodes is examined in turn. For each target node set, 10 realizations of $N = 10$ networks are created and the value of $b$ is found that sets $\frac{D_N(b)}{Nn_t} = 0.1$ to within $10^{-16}$. The average values of $b$ over sets of target nodes of the same cardinality found from the bisection method are shown in Fig. 3.13(A) where the error bars represent one standard deviation. The average deviation in Fig. 3.13(B) which remains constant as designed. The control energy is shown in Fig. 3.13(C) and the total cost is shown in Fig. 3.13(D). As the number of target nodes decreases linearly, the control energy decreases exponentially.

The main objective of this project is to demonstrate that there exists a characteristic control energy of network ensembles that can be computed numerically. In the presence of uncertain edge weights in particular, which is common for experimental systems, we have seen that the assumptions in Eq. (3.4.8) held for all systems examined. These results provide insight into the relationship between optimality and uncertainty.

## 3.5 Locally Optimal Control Strategies

The minimum energy control framework for the case the final states are constrained has been correctly criticized for, in general, not being applicable to linearized systems as it suffers from *nonlocality*. A linearization of a nonlinear system is only *valid* for some finite region, $\mathcal{S}(\mathbf{x}_p) \subseteq \mathbb{R}^n$ where the linearization is taken about $\mathbf{x}_p$. As the state deviation, $||\mathbf{x}(t) - \mathbf{x}_p||_2$, grows, the linearization error grows as well (not necessarily monotonically). The state trajectory $\mathbf{x}(t)$ corresponding to the minimum energy control input,

$$\mathbf{x}(t) = e^{At}\mathbf{x}_0 - \int_0^t e^{A(t-\tau)} B B^T e^{A^T(t_f - \tau)} d\tau \boldsymbol{\lambda}_f \tag{3.5.1}$$

is generally nonlocal, by which we mean for ever $\epsilon > 0$, there exists a $\delta(\epsilon) > 0$ such that,

$$||\mathbf{x}_0 - \mathbf{x}_f|| = \epsilon \quad \Rightarrow \quad \int_0^{t_f} ||\mathbf{x}(t)||_2 dt = \delta(\epsilon) \tag{3.5.2}$$

for initial conition $\mathbf{x}_0$ and final condition $\mathbf{x}_f$ chosen uniformly at random from some closed set in $\mathbb{R}^n$.

A small example will show why this result is expected. Let $n = 2$ and let the system matrices be,

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{3.5.3}$$

We choose $\boldsymbol{x}_0 = [0.4, 1]^T$ and $\boldsymbol{x}_f = [0.4, 1 - \gamma]^T$ for $\gamma < 1$. From the definition of $\epsilon$ in Eq. (3.5.2), we see that $\epsilon = \gamma$ for this choice of initial and final conditions. The

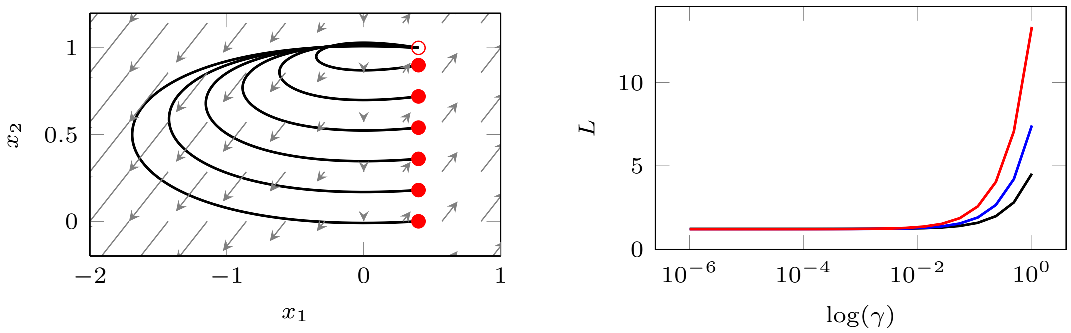


**Figure 3.14:** Example of nonlocality. (A) The minimum energy trajectories for the linear system described by the matrices in Eq. (3.5.3) where the initial condition is $\boldsymbol{x}(0) = [0.4, \quad 1]^T$ and the final condition is $\boldsymbol{x}(1) = [0.4, \quad 1-\gamma]^T$ for $\gamma > 0$. (B) The numerically computed length $L$ defined in Eq. (3.5.4) of the optimal trajectories for various values of $\gamma$ where it is clear that $\lim_{\gamma \to 0^+} L \neq 0$.

minimum energy state trajectory as written in Eq. (3.5.1) for this control maneuver for decreasing values of $\gamma$ are shown in Fig. 3.14(A) where the trajectory length appears to decrease to some shortest length. This trend is investigated for exponentially decreasing $\gamma$ in Fig. 3.14(B) where the trajectory length $L$ is seen to approach a constant positive value, the limiting value $\delta(\epsilon)$ defined in Eq. (3.5.2) as $\epsilon \to 0$ for this example. If one examines the equation for the trajectory length,

$$\begin{aligned} L &= \int_0^{t_f} \sqrt{\dot{\boldsymbol{x}}^T(t)\dot{\boldsymbol{x}}(t)}dt \\ &= \int_0^{t_f} \sqrt{\boldsymbol{x}_0^T M_1(t)\boldsymbol{x}_0 + \boldsymbol{\beta}^T M_2(t)\boldsymbol{x}_0 + \boldsymbol{\beta}^T M_3(t)\boldsymbol{\beta}}dt \\ &= \begin{bmatrix} \boldsymbol{x}_0^T & \boldsymbol{\beta}^T \end{bmatrix} \left[\int_0^{t_f} M(t)dt\right] \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{\beta} \end{bmatrix} \end{aligned} \tag{3.5.4}$$

Note that the matrix $M(t)$ does not depend on the initial and final condition. Let $\boldsymbol{x}_f \to \boldsymbol{x}_0$ so that $\boldsymbol{\beta} \approx \left(I_n - e^{At_f}\right)\boldsymbol{x}_0$. Thus, only if $[\boldsymbol{x}_0^T \quad (I_n - e^{At_f})\boldsymbol{x}_0^T]^T$ is in the null space of $\int_0^{t_f} M(t)dt$ will the minimum energy state trajectory go to zero or, if $\boldsymbol{x}_0 = \boldsymbol{0}$ and $A$ is Hurwitz. While [131] explained the phenomenon with respect to the condition of the controllability Gramian, they did not investigate remedies beyond suggesting that additional control inputs must be added.

Alternative methods for controlling general nonlinear systems have been suggested such as perturbing the initial state of a system to the basin of attraction of the desired attractor [192] or perturbing parameters of the system so as to change the shape and size of the system's attractor's basins of attraction [193, 194]. Other methods require overriding the dynamics of some states [195–197] in order to drive the remaining states to a desired attractor.

Many of the methods for controlling nonlinear systems require *brute-force* [194] by which it is meant some of the states' values must be overridden, such as the feedback vertex set method [195] or pinning control [198–200], which can be hard to implement in practice.

We approach this problem geometrically, rather than algebraically, to determine the quantitative properties of the nonlocality. Once again, let $\boldsymbol{z}_t$ denote the free evolution of the system at time $t$ and $\boldsymbol{x}_f$ be the desired final state. The control energy can be written,

$$E = (\boldsymbol{x}_f - \boldsymbol{z}(t_f))^T W^{-1} (\boldsymbol{x}_f - \boldsymbol{z}(t_f))$$

which is the equation of an ellipsoid. To be precise, let

$$\mathcal{S}(t_f, E) = \Big\{ \boldsymbol{x}_f \in \mathbb{R}^n | E = (\boldsymbol{x}_f - \boldsymbol{z}(t_f))^T \, W^{-1}(t_f) \, (\boldsymbol{x}_f - \boldsymbol{z}(t_f)) \Big\} \qquad (3.5.5)$$

be the ellipsoid which has principal axes along the directions of the eigenvectors of $W$, denoted $\boldsymbol{\xi}_k$, of widths $E\mu_k^2$, such that $W\boldsymbol{\xi}_k = \mu_k \boldsymbol{\xi}_k$. The ellipsoid is centered at $\boldsymbol{z}(t_f)$, the free evolution of the system. If one is interested in moving the states in a particular direction $\alpha\bar{\boldsymbol{x}}$, where $||\bar{\boldsymbol{x}}||_2 = 1$, we can determine the control energy required to move a distance $\alpha$ away. Examples of Eq. (3.5.5) for a two dimensional system are shown in Figs. 3.15(A) and 3.15(B) for the case that $\dot{x}_1(t) = x_2(t) + u(t)$ and $\dot{x}_2(t) = -x_1(t)$. In Fig. 3.15(A) we plot ellipsoids holding the control energy constant, $E = 1$, while letting $t_f$ vary. We see that as $t_f$ grows, the centroid of the ellipsoid moves with the free evolution of the system while the directions and lengths of the principal axes also change. In Fig. 3.15(B) we plot ellipsoids holding the time

$t_f = 1$ constant and we vary the control energy $E$. In this case, the centroid and the directions of the axes also remain constant while with increasing energy, the widths of the principal axes grow.

Let a generic nonlinear dynamical system that is affine in the controls be,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t)) + B\boldsymbol{u}(t)$$

The nonlinear system is linearized about an arbitrary point $\boldsymbol{x}_k$ using a Taylor series,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}_k + A_k\boldsymbol{x}(t) + B\boldsymbol{u}(t) + H.O.T.$$

where the matrix $A_k = \frac{d\boldsymbol{f}}{d\boldsymbol{x}}\big|_{\boldsymbol{x}=\boldsymbol{x}_k}$ is the Jacobian evaluated at $\boldsymbol{x}_k$ and the drift vector $\boldsymbol{f}_k = \boldsymbol{f}(\boldsymbol{x}_k) - A_k\boldsymbol{x}_k$. All higher order terms (quadratic, etc.) are collected in the $H.O.T.$ term. If we ignore the higher order terms, the dynamics are approximately linear when $\boldsymbol{x}(t)$ is *close* to $\boldsymbol{x}_k$, which is quantified as the *Valid Linear Region* (VLR).

$$\mathcal{N}_k = \{\boldsymbol{x} \in \mathbb{R}^n |\ ||\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}_k - A_k\boldsymbol{x}||_2 \leq \epsilon\} \tag{3.5.6}$$

We also restrict $\mathcal{N}_k$ to be closed and convex. The value of $\epsilon > 0$ is chosen to adjust the allowed tolerance for the linearization error.

The LOCS procedure can be outlined in the following set of steps.

1. Select a point, $\boldsymbol{x}_0$, to linearize about and construct $A_0 = \frac{d\boldsymbol{f}}{d\boldsymbol{x}}\big|_{\boldsymbol{x}=\boldsymbol{x}_0}$ and $\boldsymbol{f}_0 = f(\boldsymbol{x}_0) - A_0\boldsymbol{x}_0$.

2. Choose values of $t_f$ and $E$ such that $\mathcal{S}_0(t_f, E) \subseteq \mathcal{N}_0$.

3. Select a new point $\boldsymbol{x}_1 \in \mathcal{S}_0(t_f, E)$ as the terminus of the control action and repeat.

The above procedure can be used to iteratively construct a piecewise controller to move the set of states from an initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$ to some final set denoted $\mathcal{B} \subset \mathbb{R}^n$. Typically, $\mathcal{B}$ is chosen as a conservative approximation of the basin of attac-

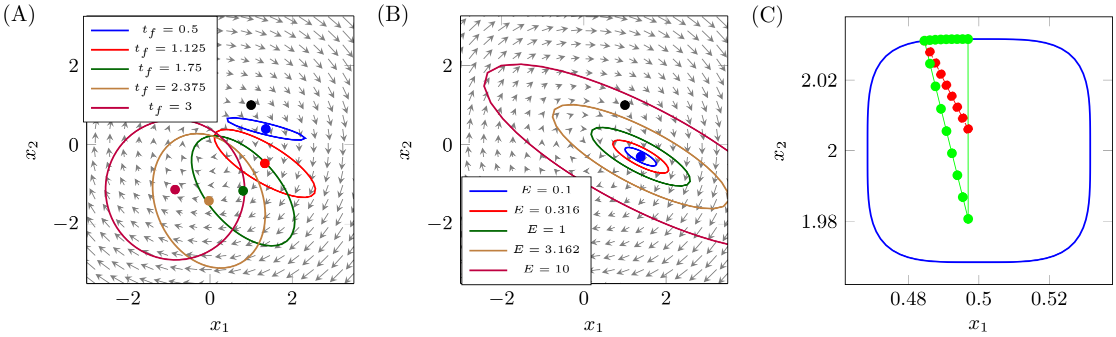


**Figure 3.15:** Examples of the LOCS ellipsoids. In (A) and (B) we plot the ellipsoids for the two dimensional controlled dynamical system $\dot{x}_1 = x_2 + u$ and $\dot{x}_2 = -x_1$. (A) The ellipsoids for $E = 1$ while letting $t_f$ vary. (B) The ellipsoids for $t_f = 1$ while letting the energy $E$ vary. (C) An example of the bounding procedure to ensure that the constant energy ellipsoid remains within the valid linear region. Adapted from [3].

tion of a desired attractor (fixed point, orbit, strange attractor, etc).

The controller requires a decision making mechanism to choose the next linearization point $\boldsymbol{x}_{k+1}$ given a starting point $\boldsymbol{x}_k$. There are three quantities which are used to judge the "fitness" of the next point.

$$F_k = w_E E_k + w_{t_f} t_k + w_x ||\boldsymbol{x}_k||_{\mathcal{B}} \tag{3.5.7}$$

The coefficients $w_E$, $w_{t_f}$, and $w_x$, are relative weights assigned to the control energy, the control horizon, and some metric that assigns a distance between a point and the set $\mathcal{B}$. The final time for the iteration is bounded between $0 < t_k \leq t_{k,\max}$ where $t_{k,\max}$ is the time at which the free evolution leaves the VLR defined in Eq. (3.5.6). For any value of $t_k$, there is an associated control energy which bounds the ellipsoid within the VLR. Finally, the choice of the metric that measures the distance between points within the ellipsoid $\mathcal{S}_k(t_k, E_k)$ can take many forms. Ideally, one would choose the value $\boldsymbol{x}_k$ that minimizes the fitness function in Eq. (3.5.7) over all possible choices. Unfortunately, minimizing the fitness function can be quite difficult due to the fact that determining the set of valid points $\boldsymbol{x}_k$ requires knowledge of the VLR for which typically there is no closed form and the eigendecomposition of the controllability Gramian for $0 < t_k < t_{k,\max}$ for which there is typically also no closed form when the dimension of the system $n > 3$.

To counter these difficulties in determining the optimal choice of $\boldsymbol{x}_k$, we use the following algorithm to select $\boldsymbol{x}_k$ from a finite set of feasible choices.

**Algorithm 1:** Description of LOCS

```
Data: Nonlinear function $\boldsymbol{f} : \mathbb{R}^n \mapsto \mathbb{R}^n$, input matrix $B \in \mathbb{R}^{n \times n_d}$,
      linearization tolerance $\epsilon > 0$, (possibly adaptive) weights $w_E$, $w_{t_f}$, $w_x$,
      initial condition $\boldsymbol{x}_0$, and basin of attraction $\mathcal{B}$
$k \leftarrow 0$;
$\boldsymbol{x}_k \leftarrow \boldsymbol{x}_0$;
while $\boldsymbol{x}_k \notin \mathcal{B}$ do
    $A_k \leftarrow \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}=\boldsymbol{x}_k}$;
    $\boldsymbol{f}_k \leftarrow \boldsymbol{f}(\boldsymbol{x}_k) - A_k \boldsymbol{x}_k$;
    $t_k^{\max} \leftarrow \begin{cases} \min & t \\ \text{s.t.} & ||\boldsymbol{f}(\boldsymbol{z}(t)) - \boldsymbol{f}_k - A_k \boldsymbol{z}(t)||_2 = \epsilon \\ & \dot{\boldsymbol{z}}(t) = \boldsymbol{f}(\boldsymbol{z}(t)), \quad \boldsymbol{z}(t_k) = \boldsymbol{x}_k \end{cases}$;
    Select $p$ times, $t_k^{(j)}$, $j = 1, \ldots, p$, such that $t_k < t_k^{(j)} < t_k^{\max}$;
    for $j = 1, \ldots, p$ do
        Integrate $\dot{W}(t) = AW(t) + W(t)A^T + BB^T$, $W(t_k) = O_n$ until $t = t_k^{(j)}$;
        Compute the eigendecomposition of $W(t_k^{(j)}$;
        for $\ell = 1, \ldots, n$ do
            $E_{+\ell}^{(j)} \leftarrow \begin{cases} \min & E \\ \text{s.t.} & ||\boldsymbol{f}(\boldsymbol{y}) - \boldsymbol{f}_k - A_k \boldsymbol{y}||_2 = \epsilon \\ & \boldsymbol{y} = \sqrt{\frac{E}{\boldsymbol{\xi}_\ell^{(j)T} W^{-1}(t_k^{(j)}) \boldsymbol{\xi}_\ell^{(j)}}} \boldsymbol{\xi}_\ell^{(j)} \end{cases}$;
            $E_{-\ell}^{(j)} \leftarrow \begin{cases} \min & E \\ \text{s.t.} & ||\boldsymbol{f}(\boldsymbol{y}) - \boldsymbol{f}_k - A_k \boldsymbol{y}||_2 = \epsilon \\ & \boldsymbol{y} = -\sqrt{\frac{E}{\boldsymbol{\xi}_\ell^{(j)T} W^{-1}(t_k^{(j)}) \boldsymbol{\xi}_\ell^{(j)}}} \boldsymbol{\xi}_\ell^{(j)} \end{cases}$;
        end
        $E_k^{(j)} \leftarrow \min_\ell E_{\pm\ell}^{(j)}$;
        for $\ell = 1, \ldots, n$ do
            $F_{\pm\ell}^{(j)} \leftarrow w_E E_k^{(j)} + w_{t_f} t_k^{(j)} \pm \sqrt{\frac{E_k^{(j)}}{\boldsymbol{\xi}_\ell^{(j)T} W^{-1}(t_k^{(j)}) \boldsymbol{\xi}_\ell^{(j)}}} \boldsymbol{\xi}_\ell^{(j)}$;
        end
    end
    $(\ell, j) \leftarrow \text{argmin } F_{\pm\ell}^{(j)}$;
    $\boldsymbol{x}_{k+1} \leftarrow \text{sign}(\ell) \sqrt{\frac{E_\ell^{(j)}}{\boldsymbol{\xi}_\ell^{(j)T} W^{-1}(t_k^{(j)}) \boldsymbol{\xi}_\ell^{(j)}}} \boldsymbol{\xi}_\ell^{(j)}$;
```

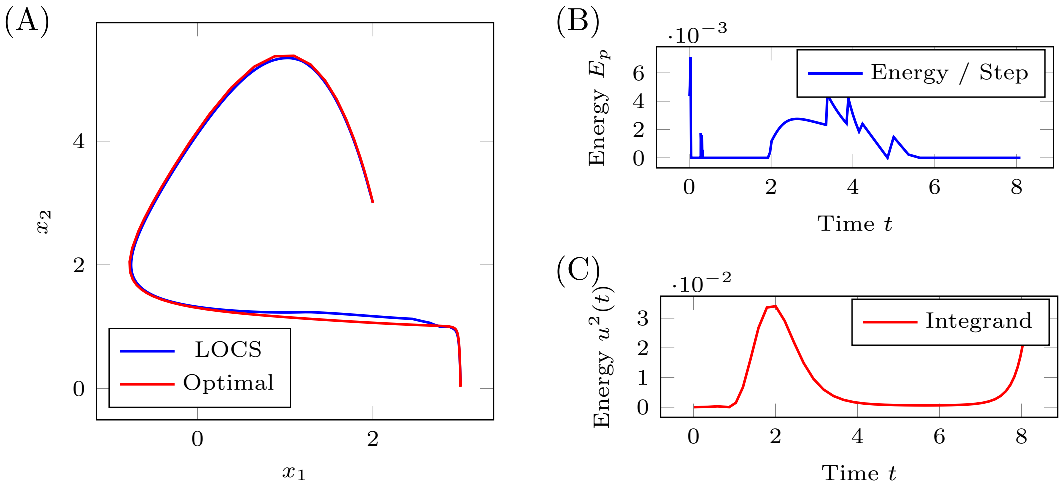


**Figure 3.16:** Small example of LOCS. (A) The optimal trajectory is shown in red which was found by numerically solving the optimal control problem using the pseudo-spectral optimal control method outlined in the previous chapter. The trajectory found using LOCS is shown in blue which is quite similar to the optimal trajectory. The control energy used per step of the LOCS procedure is shown in (B) and the instantaneous control energy of the optimal solution is shown in (C). Adapted from [3].

A two dimensional example of the output of the LOCS procedure is shown in Fig. 3.16 which is governed by the system of equations,

$$\dot{x}_1(t) = (x_1(t) - 3)(x_2(t) - 2)$$
$$\dot{x}_2(t) = x_2(t)(x_1(t) - 1)(x_2(t) - 1) + u(t)$$

with initial condition $x_1(0) = 2$ and $x_2(0) = 3$ and final condition $x_1(t_f) = 3$ and $x_2(t_f) = 0$. In Fig. 3.16(A) we show the optimal state trajectory found by solving the optimal control problem using the pseudospectral method outlined in the previous chapter. Also shown is the state trajectory computed using LOCS which is quite similar to the optimal solution. This was achieved by the proper choice of the fitness function in Eq. (3.5.7),

$$F = 4E_k + 0.8|x_1(t_k) - 3| + 0.2|x_2(t_f)|$$

Also shown is the energy used per step of the LOCS procedure in Fig. 3.16(B) and the instantaneous energy found from the optimal solution in Fig. 3.16(C). While the

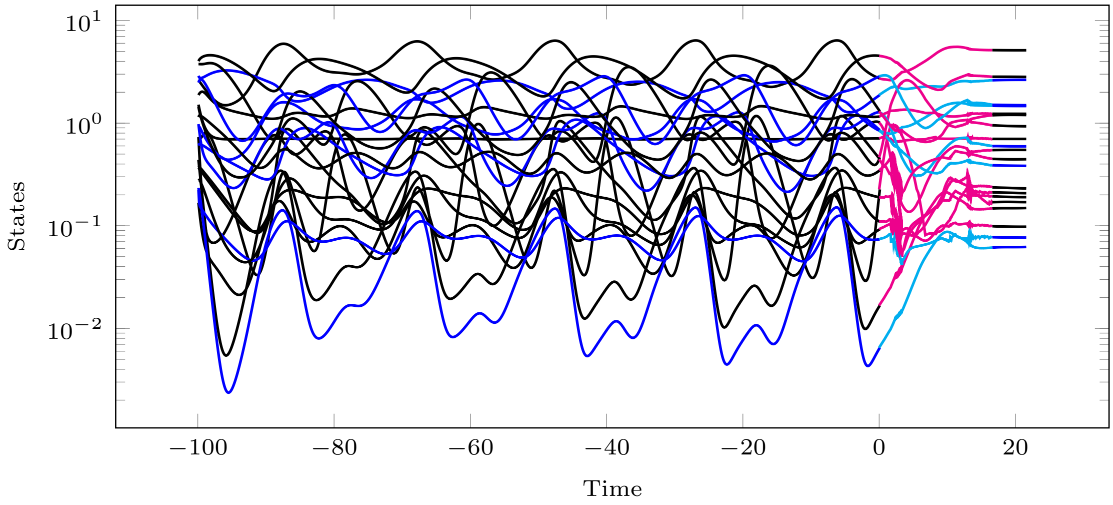


**Figure 3.17:** Application of LOCS to a model of the regulatory structure of the circadian clock in mice. The initial time trajectories of the states in dark colors are the initial condition of the system on a stable periodic orbit. The lighter colors in the second time period correspond to when the LOCS controller has been turned on. The last stretch of dark colors show the LOCS controller's success as the system is now at the desired fixed point. Adapted from [3].

state trajectories appear similar, the times at which energy is supplied to the system are clearly different.

For a larger example, we apply LOCS to a model of the regulatory structure of the intracellular circadian clock in mice [201]. This model was used as a demonstration of the *feedback vertex set method* [195, 196]. The feedback vertex set requires overriding the dynamics of the states of the nodes found to be in the minimal feedback vertex set, a brute-force method [194]. The initial condition of the system is set on a periodic orbit and the desired attractor is a stable fixed point. In Fig. 3.17, we show the initial evolution of the system for $t < 0$ on the periodic orbit, represented in the darker colors. Black lines correspond to non-driver nodes while blue lines correspond to driver nodes. The next segment of time, represented in lighter colors, plots the states while the LOCS controller is turned on and the system is driven towards the desired fixed point. The last segment of time, represented in darker colors again, shows the system now residing at the desired fixed point.

In this section, a connection between the minimum energy control of linear systems and control of nonlinear systems is presented. An example of its application to a nonlinear system is shown, but many more applications can be found in the supplementary information of [3] including the New England Power Grid [202], generalized Lotka-Volterra equations, discrete dynamics, in the presence of noise and parameter mismatch, systems with multiple timescales, and systems with riddled basins of attraction [203].

## 3.6 Optimal Control of Networks in the Presence of Attackers and Defenders [7]

An application of the minimum energy control is to provide counter measures if a dynamical network is attacked. In this case, the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is augmented with an additional $n_a$ nodes, $\mathcal{W}$, which represent the attackers and edges $\mathcal{F} \subset \mathcal{W} \times \mathcal{V}$ that represent which attackers are attached to which nodes. The dynamics of each of the attackers is assumed to be,

$$\dot{w}_j(t) = s_j w_j(t) + r_i, \quad j = 1, \dots, n_a \tag{3.6.1}$$

which can represent either constant valued attacks ($s_j = 0$ and $r_j = 0$), linearly increasing attacks ($s_j = 0$ and $r_j \neq 0$), or exponentially increasing attacks ($s_j > 0$). Let $S \in \mathbb{R}^{n_a \times n_a}$ be a diagonal matrix with elements $s_j$ that represent the attackers described by Eq. (3.6.1) and let $H \in \mathbb{R}^{n \times n_a}$ be the adjacency matrix associated with the edges in $\mathcal{F}$. The combined node state and attacker system can be written,

$$\begin{aligned} \begin{bmatrix} \dot{\boldsymbol{x}}(t) \\ \dot{\boldsymbol{w}}(t) \end{bmatrix} &= \begin{bmatrix} A & H \\ O_{n_a \times n} & S \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{w}(t) \end{bmatrix} + \begin{bmatrix} \mathbf{0}_n \\ \boldsymbol{r} \end{bmatrix} + \begin{bmatrix} B \\ O_{n_a \times n_d} \end{bmatrix} \boldsymbol{u}(t) \\ \boldsymbol{x}(t) &= \begin{bmatrix} I_n & O_{n \times n_a} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{w}(t) \end{bmatrix} \end{aligned} \tag{3.6.2}$$

which is a linear system with an additional drift vector $\boldsymbol{r}$. The composite system in Eq. (3.6.2) can more succinctly be written as,

$$\begin{aligned}\dot{\boldsymbol{\xi}}(t) &= \bar{A}\boldsymbol{\xi}(t) + \bar{\boldsymbol{r}} + \bar{B}\boldsymbol{u}(t)\\ \boldsymbol{x}(t) &= \bar{C}\boldsymbol{\xi}(t)\end{aligned}$$

The resulting optimal control problem we solve is the same as was addressed in Sec. 3.2, or more specifically, as written in Eq. (3.2.1) with the sole modification that there is an additional drift vector in the linear dynamical equation. The solution of the optimal control problem with this modification is nearly identical except that the control maneuver, $\boldsymbol{\beta}$, is only slightly different.

$$\boldsymbol{\beta} = \bar{C}e^{\bar{A}t_f}\boldsymbol{\xi}_0 + C\int_0^{t_f} e^{\bar{A}\tau}d\tau\bar{\boldsymbol{r}} - \boldsymbol{x}_f \tag{3.6.3}$$

where $\boldsymbol{x}_f$ is the desired final values of the states of the network. The control energy is still $E = \boldsymbol{\beta}^T\bar{W}\boldsymbol{\beta}$ where $\boldsymbol{\beta}$ is defined in Eq. (3.6.3) and the output controllability Gramian is $\bar{W} = \int_0^{t_f} e^{A\tau}BB^Te^{A^T\tau}d\tau$, or more specifically, the attackers' dynamics do not affect it. In [7], the affect of the number of attackers and the placement of attackers is examined methodically for simple graph topologies such as chain graphs, ring graphs, and star graphs. Additionally, for a graph constructed using the Barábasi-Albert preferential attachment method with $n = 100$ nodes, we select 10% of the nodes to be drivers and choose each node to be attacked one by one. We find that there is a linear relationship between the degree of the attacked node and the control energy necessary to counter the unstable dynamics.

## 3.7 Conclusion

This chapter covered results concerning properties of the solutions of the optimal control problem presented in Eq. (3.2.1) which is derived in Appendix B.1. We have seen that for the *final constraint* case in Sec. 3.2, the control energy increases exponentially with the number of target nodes $|\mathcal{T}| = n_t$. On the other hand, if we relax the constraints so they are included as an additional term in the cost function, the *final cost* case in Sec. 3.3, we can exponentially decrease the control energy while sacrificing the *accuracy*. Combining these results provides a framework by which one may balance accuracy and control energy for a desired control action on a complex network.

Additionally, in Sec. 3.4 we derived a control input, and the control energy, when the adjacency matrix is not known exactly, but instead is known to be one of a finite set of possible realizations. In this framework, with the proper choice of weighting parameter, $\alpha$, it is proven that the average deviation, the control energy, and the total cost all approach constant values in the limit $N \to \infty$. This result suggests that there exists control inputs capable of approximately driving an entire network ensemble to some desired output.

We use the fact the control energy expression can be expressed as an ellipsoid to derive a control for nonlinear systems which we call *locally optimal control strategy* in Sec. 3.5. Its utility is demonstrated for a selection of nonlinear systems. The minimum energy control framework is also applied to systems when there are malicious actors attacking the networked system in Sec. 3.6. In particular, we apply this optimal control to linearized power systems.

While much of the analysis concerning the scaling of the control energy is predictable in the mean over all choices of sets of target nodes, we have seen that the variance can be extremely large which we have not yet explored. This large variance is explained in terms of distances, measured as the length of the shortest path, between driver nodes and target nodes in Chapter 5 for simplified models of networks.

# Chapter 4

# Symmetry in Graphs

## 4.1 Introduction

It has been shown that symmetries in graphs can have both beneficial and detrimental effects on dynamical systems. For example, it has been shown that symmetries in graphs increase the number of independent control signals required to ensure a system is controllable [75]. On the other hand, symmetries in graphs can be used to reduce the dimension of the dynamics (by examining the quotient graph) [204–206]. Symmetries can also induce consensus and synchronization in dynamical networks [204, 207, 208].

Three topics are covered in this chapter; (i) a method one may construct graphs with pre-defined symmetry structure, (ii) an analysis of stable synchronization despite unstable dynamics, and (iii) a proposed method to include edge weights in a new definition of *approximate symmetries*. Details of more extensive proofs are contained in Appendix C.

In section 4.2, some definitions and lemmas about the automorphism group of a graph are stated for use later. The results of [8, 9] are described in Sections 4.3 and 4.4 covering the concept of *feasible quotient graphs* and generating graph's with symmetries. The problem of stable cluster synchronization corresponding to properties of the automorphism group of a graph is discussed in Section 4.7 as described in [12]. The current state of ongoing work on the problem of approximate symmetries is discussed in Section 4.8 and the current difficulties faced. Work that the author contributed to on symmetries in multi-layer and multiplex networks is briefly covered in Section 4.6. Finally, concluding remarks are made in Section 4.9.

## 4.2 Graph Symmetries and the Automorphism Group

This section expands on some of the definitions introduced in Chapter 2. As a brief reminder of the notation, let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph and let $\text{Aut}(\mathcal{G})$ be the automorphism group of this graph defined in Def. 2.2.23. The automorphism group induces an equitable partition of the nodes (see Def. 2.2.18 for partitions and Def. 2.2.20 for equitable partitions).

**Lemma 4.2.1** (Automorphism Group Induced Equitable Partition: Orbits)**.** *The automorphism group, $Aut(\mathcal{G})$, induces an equitable partition of the nodes in the graph, denoted as $\mathcal{O}$, in which the clusters are commonly referred to as* orbits*. Two nodes, $v_j$ and $v_k$, are in the same orbit if there exists a permutation $\pi \in Aut(\mathcal{G})$ such that $\pi(v_j) = v_k$.*

For the derivations in the following sections, we will assume that $\mathcal{G}$ is *undirected* (see Def. 2.2.1), but extensions to *directed* graphs does not require too many modifications. An equitable partition of the nodes in a graph can be used to compress a graph into a *quotient graph.*

**Definition 4.2.1** (Quotient Graphs)**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph and let $\mathcal{C}$ be an equitable partition of its nodes. The quotient graph is defined as a structure with 5 elements: $\mathcal{Q} = \{\mathcal{C}, \mathcal{F}, \boldsymbol{n}, \boldsymbol{s}, \boldsymbol{w}\}$*

- *$\mathcal{C}$: The nodes of the quotient graph are the clusters of nodes in the equitable partition.*

- *$\mathcal{F}$: The edges of the quotient graph. If there exists an edge $(v_j, v_k) \in \mathcal{E}$ and $v_j \in \mathcal{C}_a$ and $v_k \in \mathcal{C}_b$ then there is a quotient edge $(\mathcal{C}_a, \mathcal{C}_b)$ in the quotient graph.*

- *$\boldsymbol{n} \in \mathcal{Z}^{|\mathcal{V}|}$: The populations of the original cluster, $n_k = |\mathcal{C}_k|$.*

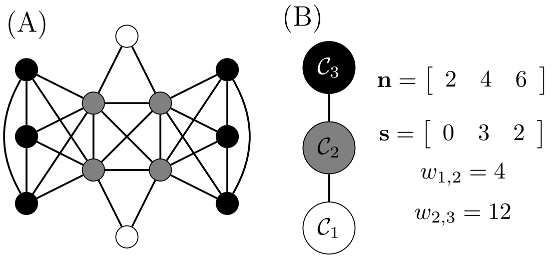


**Figure 4.1:** An example of a quotient graph of an undirected graph. (A) An undirected graph with nodes colored according to an equitable partition. (B) The resulting structure of the quotient graph along with the properties $\boldsymbol{n}$, $\boldsymbol{s}$, and $\boldsymbol{w}$.

- $\boldsymbol{s} \in \mathcal{Z}^{|\mathcal{V}|}$: *The intra-cluster degrees, that is, if a node* $v_j \in \mathcal{C}_k$, *then* $s_k = |\mathcal{N}_{j,k}|$.
- $\boldsymbol{w} \in \mathcal{Z}^{|\mathcal{E}|}$: *The quotient edge weights which is equal to the number of edges passing between all nodes in the two connected clusters. If the k'th quotient edge is* $(\mathcal{C}_i, \mathcal{C}_j)$, *then* $\boldsymbol{w}_k$ *is the total number of edges passing between nodes in these two clusters.*

An example of an undirected graph, with nodes colored according to an equitable partition, is shown in Fig. 4.1(A). There are three clusters in the partitioned, the white cluster denoted $\mathcal{C}_1$, the gray cluster denoted $\mathcal{C}_2$, and the black cluster denoted $\mathcal{C}_3$. The resulting quotient graph structure and additional information of the graph in Fig. 4.1(A) is shown in Fig. 4.1(B). The original graph has 12 nodes and 28 edges while the quotient graph has 3 quotient nodes and 2 quotient edges along with the data required by $\boldsymbol{n}$, $\boldsymbol{s}$, and $\boldsymbol{w}$.

Determining the automorphism group of a graph and representing its orbits as a quotient graph is a famously difficult problem [33]. A number of software packages have been developed to find the automorphism group of a graph such as nauty [37], traces [209] , bliss [210], saucy [211], and conauto [212]. The reverse problem, of constructing an originating graph with symmetries given a quotient graph, has, as far as the author and his collaborators can tell, not been investigated. To construct an originating graph from a quotient graph, the following questions must be answered.

- Does an originating graph exist? Or in other words, is the quotient graph *feasible*?

- What types of symmetries will exist in the resulting automorphism group?

These two questions are answered in sections 4.3 and 4.4, respectively.

## 4.3 Feasible Quotient Graphs

To answer the first question listed above, we derive the conditions for which a quotient graph can represent some original graph, that is, whether the quotient graph is *feasible.* While the structure of a quotient graph has been defined in Def. 4.2.1 as a graph and a set of $2|\mathcal{C}| + |\mathcal{F}|$ integers, the reverse is not necessarily true, that is not all graphs along with appropriate sets of integers are quotient graphs. The conditions which $\mathcal{Q}$ must satisfy are addressed individually using the following theorems.

**Theorem 4.3.1** (Erdős-Gallai [213]). *Let $a = (a_1, a_2, \ldots, a_n)$ be a non-increasing sequence of non-negative integers. The sequence is realizable as the degree sequence of an undirected simple graph (no self-loops or multi-edges) if and only if*

1. $\sum_{i=1}^{n} a_i$ *is even, and*

2. *for* $1 \leq k \leq n$,
$$\sum_{i=1}^{k} a_i \leq k(k-1) + \sum_{i=k+1}^{n} \min\{a_i, k\}$$

**Lemma 4.3.1** (Intra-Cluster Degree Condition). *Using Thm. 4.3.1 for the special case that the sequence consists of a single integer, say $s$, of length $n$, then the two conditions become,*

1. *The product $ns$ must be even, and*

2. $s + 1 \leq n$

*Proof.* The first condition, that $ns$ is even, can be seen from $\sum_{i=1}^{n} s = ns$ in the first condition of Thm. 4.3.1. The second condition comes from simplifying the second condition of Thm. 4.3.1 for the specific case $a_i = s$,

$$ks = k(k-1) + (n-k)\min\{s, k\}, \quad 1 \leq k \leq n$$

If $k \leq s$, then the condition simplifies to $ks \leq (n-1)k$ which can be re-written $s+1 \leq n$. If $k > s$, the inequality is trivially satisfied. □

**Theorem 4.3.2** (Gale-Ryser [214]). *Define the following sequences,*

$$a = (a_1, a_2, \ldots, a_{n_1}) \quad \textit{and} \quad b = (b_1, b_2, \ldots, b_{n_2}),$$

*to be of length $n_1$ and $n_2$, respectively. The sequences $a$ and $b$ can be realizable as the degree sequence of a bipartite graph if and only if,*

$$\sum_{i=1}^{k} a_i \leq \sum_{i=1}^{n_2} \min\{b_i, k\}, \quad 1 \leq k \leq n_1 \tag{4.3.1}$$

*or, equivalently,*

$$\sum_{i=1}^{k} b_i \leq \sum_{i=1}^{n_1} \min\{a_i, k\}, \quad 1 \leq k \leq n_2 \tag{4.3.2}$$

**Lemma 4.3.2** (Inter-Cluster Degree Condition). *Let $a$ and $b$ be two sequences of constant value $r_1$ and $r_2$ of lengths $n_1$ and $n_2$, respectively. Taking the necessary and sufficient conditions in Thm. 4.3.2 and applying the constant sequences yield the three conditions,*

$$r_1 \leq n_2, \quad r_2 \leq n_1, \quad r_1 n_1 = r_2 n_2$$

*Proof.* Four conditions must be checked, Eq. (4.3.1) when $k \leq r_2$ and when $k > r_2$ and Eq. (4.3.2) when $k \leq r_1$ and when $k > r_1$. The two conditions for the specific constant sequences described can be simplified to,

$$kr_1 \leq \sum_{i=1}^{n_2} \min\{r_2, k\} = n_2 \min\{r_2, k\}, \quad 1 \leq k \leq n_1$$
$$kr_2 \leq \sum_{i=1}^{n_1} \min\{r_1, k\} = n_1 \min\{r_1, k\}, \quad 1 \leq k \leq n_2$$

Taking the first condition for $k \leq r_2$ yields,

$$kr_1 \leq kn_2 \quad \Rightarrow \quad r_1 \leq n_2$$

which implies the first condition in the Lemma. For $r_2 < k \leq n_1$,

$$kr_1 \leq n_2 r_2 \quad \Rightarrow \quad n_1 r_1 \leq n_2 r_2 \tag{4.3.3}$$

Taking the second condition for $k \leq r_1$ yields,

$$kr_2 \leq kn_1 \quad \Rightarrow \quad r_2 \leq n_1$$

which implies the second condition in the Lemma. For $r_1 < k \leq n_2$,

$$kr_2 \leq n_1 r_1 \quad \Rightarrow \quad n_2 r_2 \leq n_1 r_2 \tag{4.3.4}$$

For Eqs. (4.3.3) and (4.3.4) to both be satisfied simultaneously,

$$r_1 n_1 = r_2 n_2$$

□

Combining the results of Lemmas 4.3.2 and 4.3.1 yields the definition of a feasible quotient graph.

**Definition 4.3.1** (Feasible Quotient Graphs)**.** *Let $\mathcal{Q} = \{\mathcal{C}, \mathcal{F}, \boldsymbol{s}, \boldsymbol{n}, \boldsymbol{w}\}$ be a quotient graph as defined in Def. 4.2.1. Said quotient graph is also* feasible *if,*

*1.* $s_k + 1 \leq n_k$, $\forall \mathcal{C}_k \in \mathcal{C}$

*2.* $n_k s_k \bmod 2 = 0$, $\forall \mathcal{C}_k \in \mathcal{C}$

*3.* $w_{k,\ell} \leq n_k n_\ell$, $\forall (\mathcal{C}_k, \mathcal{C}_\ell) \in \mathcal{F}$

*4.* $w_{k,\ell} \bmod n_k = 0$ *and* $w_{k,\ell} \bmod n_\ell = 0$, $\forall (\mathcal{C}_k, \mathcal{C}_\ell) \in \mathcal{F}$

### 4.3.1 Creating Feasible Quotient Graphs

The set of conditions laid forth in Def. 4.3.1 are not trivially satisfiable for arbitrary connectivity pattern as denoted by the set of quotient edges, $\mathcal{F}$. In what follows, we assume that $\mathcal{C}$ and $\mathcal{F}$ have been chosen *a priori* and that we must determine the integers in $\boldsymbol{n}$, $\boldsymbol{s}$, and $\boldsymbol{w}$. The most general framework assigns a cost to every unknown, $c_k^p$ for each $p_k$, $c_k^s$ for each $s_k$, and $c_{k,\ell}^w$ for each quotient edge $(\mathcal{C}_k, \mathcal{C}_\ell) \in \mathcal{F}$. Condition 2 in Def. 4.3.1 can be rewritten introducing extra variables as,

$$\begin{aligned}
&n_k + y_k = 2z_k, \quad y_k \in \{0,1\}, \quad z_k \geq 1 \\
&s_k + a_k = 2b_k, \quad a_k \in \{0,1\}, \quad b_k \geq 0 \\
&y_k + a_k \leq 1
\end{aligned}$$

The interpretation of the above set of equation is that $y_k = 1$ if $n_k$ is odd while $y_k = 0$ if $n_k$ is even. The variables $s_k$ and $a_k$ have the same relationship. Then, requiring the product $n_k s_k$ to be even is equivalent to requiring that either $n_k$ or $s_k$ be even, which is ensured by requiring $y_k + a_k \leq 1$.

Unfortunately, linearizing conditions 3 and 4 in Def. 4.3.1 is not possible without introducing many auxiliary variables. Thus, to find a feasible quotient graph without pre-specifying any of the values requires solving a quadratic integer program.

Rather than facing the daunting task of the quadratic integer programming problem and the unclear relationship between the cost function coefficients and desired properties of the resulting graph, alternative methods are presented here where one defines *part* of the quotient graph and then compute the remaining parts ensuring feasibility. The first method allows one to design the connectivity, that is, the intra-cluster degrees $s_k$ $\forall \mathcal{C}_k \in \mathcal{C}$ and the quotient edge weights $w_{a,b}$ $\forall (\mathcal{C}_a, \mathcal{C}_b) \in \mathcal{F}$ and then compute cluster populations $n_k$ to ensure feasibility. The second method instead allows one to design the cluster populations and intra-cluster degrees and then computes the necessary quotient edge weights, $w_{a,b}$, to ensure feasibility.

### 4.3.2 Forcing Connectivity [8]

In this framework, the number of clusters $|\mathcal{C}|$, the set of quotient edges $\mathcal{F}$, the intra-cluster degrees, $s_k$, and the quotient edge weights $W_{a,b}$, are all designed with some restrictions which will be discussed. Importantly, note that rather than computing the quotient edge weight directly, we instead look for two values, denoted,

$$\frac{w_{a,b}}{n_a} = W_{a,b} \quad \text{and} \quad \frac{w_{a,b}}{n_b} = W_{b,a} \tag{4.3.5}$$

which is equal to the cluster degree from cluster $b$ to cluster $a$ (see Def. 2.2.19 for more information). An integer linear program (ILP) is constructed to find the unknown cluster populations, $|\mathcal{C}_k| = n_k$. For notational simplicity, the constraints for those clusters with even and odd intra-cluster degree are slightly different and so define,

$$m_k = s_k \bmod 2$$

The variables in the ILP, $x_k$, are defined as,

$$(1 + m_k)x_k = n_k \tag{4.3.6}$$

to satisfy the constraint in Lemma 4.3.1. A lower bound is assigned to each variable from the remaining constraints,

$$x_k \geq \left(1 - \frac{m_k}{2}\right) \max\left\{\max_{(\mathcal{C}_j,\mathcal{C}_k)\in\mathcal{F}} W_{j,k}, s_k + 1\right\} = x_k^{LB}$$

The remaining constraints, $W_{j,k}n_k = W_{k,j}n_j$, $\forall(\mathcal{C}_j, \mathcal{C}_k) \in \mathcal{F}$, are represented in matrix form as,

$$W_{j,k}(1 + m_k)x_k - W_{k,j}(1 + m_j)x_j = 0$$

which can be composed into the system of linear equations,

$$A\boldsymbol{x} = \boldsymbol{0}$$

where the matrix $A \in \mathbb{R}^{|\mathcal{F}| \times |\mathcal{C}|}$ has rows corresponding to each quotient edge. If row $\ell$ corresponds to the quotient edge $(\mathcal{C}_j, \mathcal{C}_k) \in \mathcal{F}$, then matrix $A$ has elements,

$$\begin{aligned} A_{\ell,j} &= (1 + m_k) W_{j,k} \\ A_{\ell,k} &= -(1 + m_j) W_{k,j} \\ A_{\ell,a} &= 0, \quad a \neq j, k \end{aligned} \tag{4.3.7}$$

The ILP is composed with weights assigned to each population, $c_k$, $\forall \mathcal{C}_k \in \mathcal{C}$, in the vector $\boldsymbol{c}$.

$$\begin{aligned} \min \quad & \boldsymbol{c}^T \boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} = \boldsymbol{0} \\ & \boldsymbol{x} \geq \boldsymbol{x}^{LB} \end{aligned} \tag{4.3.8}$$

A caveat of this formulation arises due to the constraints which require that the matrix $A$ as defined in Eq. (4.3.7) has a non-empty null space. This is not a trivial requirement as typically $|\mathcal{F}| \geq |\mathcal{C}|$ and so the choice of the quotient edge weights $W(\mathcal{C}_j, \mathcal{C}_k)$ is not free. Instead, one may choose the quotient edge weights for the first $|\mathcal{C}| - 1$ quotient edges freely. For the remaining $|\mathcal{F}| - |\mathcal{C}| + 1$ quotient edges, the quotient edge weights must be chosen such that the corresponding row of the matrix $A$ is a linear combination of the previous rows.

Due to the difficulty of choosing quotient edge weights to ensure that the matrix $A$ has a non-empty null space, this method is mostly suited for acyclic quotient graphs, i.e., if $|\mathcal{F}| < |\mathcal{C}|$. An example of the ILP used to compute the cluster populations is shown in Fig. 4.2. In Fig. 4.2(A) a three node quotient graph is shown with intra-cluster degrees $s_k$ and quotient edge weights $W_{j,k}$ assigned. The quotient graph's adjacency matrix is shown as the matrix $Q$, and the constraint matrix for the ILP is shown as the matrix $A$. The complete ILP is written out in Fig. 4.2(B).

The solution to this ILP is $x_1 = 1$, $x_2 = 4$, and $x_3 = 1$ which leads to populations $n_1 = 2$, $n_2 = 4$ and $n_3 = 1$. Once a solution is found, $\boldsymbol{x}$, that lies in the null space of $A$, one can scale $\boldsymbol{x}$ by any integer, $r$, to tune the size of the resulting graph before using Eq. (4.3.6). To create the graph in Fig. 4.2(D), $r = 2$ to double the size of

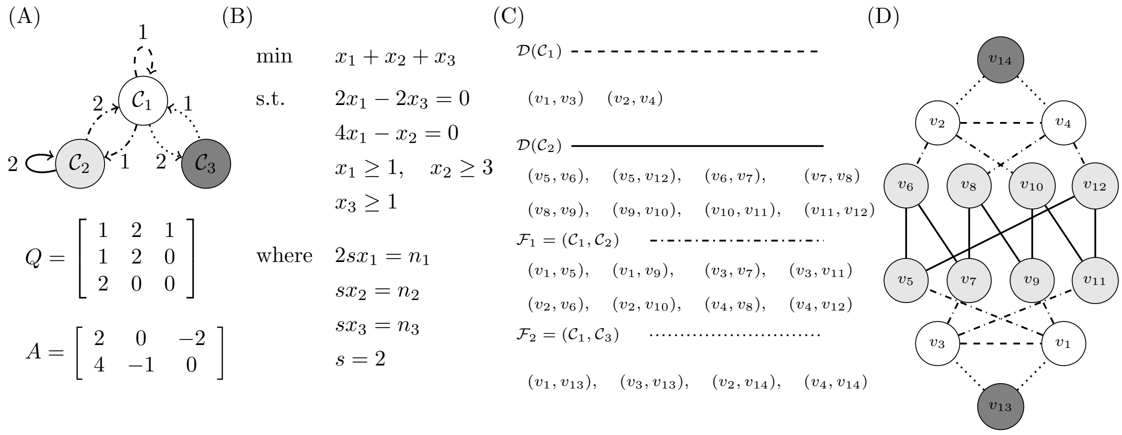

(A)
1
2
$\mathcal{C}_1$
1
2
$\mathcal{C}_2$
1
2
$\mathcal{C}_3$
$Q = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 2 & 0 \\ 2 & 0 & 0 \end{bmatrix}$
$A = \begin{bmatrix} 2 & 0 & -2 \\ 4 & -1 & 0 \end{bmatrix}$
(B)
min $x_1 + x_2 + x_3$
s.t. $2x_1 - 2x_3 = 0$
$4x_1 - x_2 = 0$
$x_1 \geq 1, \quad x_2 \geq 3$
$x_3 \geq 1$
where $2sx_1 = n_1$
$sx_2 = n_2$
$sx_3 = n_3$
$s = 2$
(C)
$\mathcal{D}(\mathcal{C}_1)$
$(v_1, v_3)$ $(v_2, v_4)$
$\mathcal{D}(\mathcal{C}_2)$
$(v_5, v_6)$, $(v_5, v_{12})$, $(v_6, v_7)$, $(v_7, v_8)$
$(v_8, v_9)$, $(v_9, v_{10})$, $(v_{10}, v_{11})$, $(v_{11}, v_{12})$
$\mathcal{F}_1 = (\mathcal{C}_1, \mathcal{C}_2)$
$(v_1, v_5)$, $(v_1, v_9)$, $(v_3, v_7)$, $(v_3, v_{11})$
$(v_2, v_6)$, $(v_2, v_{10})$, $(v_4, v_8)$, $(v_4, v_{12})$
$\mathcal{F}_2 = (\mathcal{C}_1, \mathcal{C}_3)$
$(v_1, v_{13})$, $(v_3, v_{13})$, $(v_2, v_{14})$, $(v_4, v_{14})$
(D)
$v_{14}$
$v_2$
$v_4$
$v_6$
$v_8$
$v_{10}$
$v_{12}$
$v_5$
$v_7$
$v_9$
$v_{11}$
$v_3$
$v_1$
$v_{13}$

**Figure 4.2:** (A) A quotient graph with three quotient nodes and two quotient edges. This quotient graph's adjacency matrix, $Q$, is shown below the quotient graph, and the matrix $A$ that appears in Eq. (4.3.8) is shown below it. (B) The ILP that must be solved to determine the population that each quotient node represents. (C) Each quotient edge and quotient loop in (A) is shown by its pattern, and the created edges for the full graph are listed underneath. (D) The resulting graph with the desired symmetries as dictated by the original quotient graph in (A).

each cluster. Further details on how the set of edges in Fig. 4.2(C) were found are contained in Sec. 4.4.

### 4.3.3 Forcing Populations [9]

With the acknowledged difficulty of construcing quotient edge weights that ensure the constraint matrix in the ILP in Eq. (4.3.8) is not full rank, an alternative method to construct feasible quotient graphs was conceived. In this framework, we instead design the intra-cluster degrees $s_k$ and the cluster populations $n_k$ for all $\mathcal{C}_k \in \mathcal{C}$. Rearranging the fourth constraint for a feasible quotient graph in Def. 4.3.1 using the alternative definition of the quotient edge weights in Eq. (4.3.5), the requirement for the quotient edge weights can be rewritten,

$$\frac{n_j}{n_k} = \frac{W_{k,j}}{W_{j,k}}, \quad n_j \geq W_{k,j}, \quad n_k \geq W_{j,k}, \quad \forall (\mathcal{C}_j, \mathcal{C}_k) \in \mathcal{F} \tag{4.3.9}$$

Let $c_{j,k} = \gcd(n_j, n_k)$ be the greatest common denominator between the two cluster populations and let $d_{j,k}$ be any factor of $c_{j,k}$. The quotient edge weight relation in Eq.

(4.3.9) can be written as,

$$W_{j,k} = \frac{n_k}{d_{j,k}} \quad \text{and} \quad W_{k,j} = \frac{n_j}{d_{j,k}} \tag{4.3.10}$$

Note that $d_{j,k} = 1$ is always a valid choice for Eq. (4.3.10). The resulting number of edges between the nodes in clusters $\mathcal{C}_j$ and $\mathcal{C}_k$ is $w_{j,k} = n_j W_{j,k} = n_k W_{k,j} = \frac{n_j n_k}{d_{j,k}}$. The density of the edges between nodes in $\mathcal{C}_j$ and $\mathcal{C}_k$ can be written,

$$\rho_{j,k} = \frac{w_{j,k}}{n_j n_k} = \frac{1}{d_{j,k}} \tag{4.3.11}$$

The result of Eq. (4.3.11) provides a way to choose the values of $d_{j,k}$ as to adjust the density of the resulting graph. The benefit of this method is that one may determine the values $W_{j,k}$ and $W_{k,j}$ independently of each other, so one does not need to solve the system of diophantine equations like in the previous subsection.

## 4.4 Generating Symmetric Graphs

At this point, we assume $\mathcal{Q}$ is a feasible quotient graph constructed using either of the methods described above or some other way. The graph to create is denoted $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where,

$$\mathcal{V} = \bigtimes_{k=1}^{n_c} \mathcal{C}_k, \tag{4.4.1}$$

is the collection of the nodes that make up each cluster and the edges,

$$\mathcal{E} = \left( \bigtimes_{k=1}^{n_c} \mathcal{E}_k \right) \times \left( \bigtimes_{(\mathcal{C}_j, \mathcal{C}_k) \in \mathcal{F}} \mathcal{E}_{j,k} \right). \tag{4.4.2}$$

are composed of both the intra-cluster edges and the inter-cluster edges. The nodes in defined by Eq. (4.4.1) are labeled according to their originating cluster so that,

$$\mathcal{C}_k = \left\{ v_\ell^k | \ell = 0, \ldots, p_k - 1 \right\}, \quad |\mathcal{C}_k| = p_k \tag{4.4.3}$$

The intra-cluster edges and inter-cluster edges sets, as defined in Eq. (4.4.2), each consist of only edges,

$$\begin{aligned} &\mathcal{E}_k \subseteq \mathcal{C}_k \times \mathcal{C}_k, \quad |\mathcal{E}_k| = s_k, \qquad k = 0, \dots, n_c - 1 \\ &\mathcal{E}_{j,k} \subseteq \mathcal{C}_j \times \mathcal{C}_k, \quad |\mathcal{E}_{j,k}| = w_{j,k}, \quad \forall(\mathcal{C}_j, \mathcal{C}_k) \in \mathcal{F} \end{aligned} \tag{4.4.4}$$

respectively. To simplify the following notation in the theorems, apply the notation in Eqs. (4.4.3) and (4.4.4) and let edges be equivalently denoted in terms of the nodes' cluster indices, that is, in the proper context,

$$(v^j_\ell, v^k_{\ell'}) = (\ell, \ell'), \quad \ell = 0, \dots, n_j - 1, \quad \ell' = 0, \dots, n_k - 1 \tag{4.4.5}$$

The intra-cluster edges are constructed using the following theorems with the notation in Eq. (4.4.5).

**Theorem 4.4.1** (Intra-Cluster Edges). *Let there be $n_k$ nodes, $v^k_\ell$, $\ell = 0, \dots, n_k - 1$, and let $0 \leq s_k < n_k$ be the intra-cluster degree of the nodes in $\mathcal{C}_k$. Let $b$ be a set of integers between $1$ and $\frac{n_k - n_k \bmod 2}{2}$ without repetition. Then, the edges $(\ell, \ell') \in \mathcal{E}_k$, are,*

$$\begin{matrix} (\ell \bmod n_k, (\ell + b_j) \bmod n_k) \\ (\ell \bmod n_k, (\ell - b_j) \bmod n_k) \end{matrix} \quad j = 1, \dots \frac{s_k - s_k \bmod 2}{2} \tag{4.4.6}$$

*If $s_k$ is odd, then we additionally add one more edge,*

$$(\ell \bmod n_k, (\ell + n_k/2) \bmod n_k)$$

The proof is contained in Appendix C.1.

**Theorem 4.4.2** (Inter-Cluster Edges). *Let $(\mathcal{C}_j, \mathcal{C}_k) \in \mathcal{F}$ be a quotient edge. Define $c = gcd(n_j, n_k)$ to be the greatest common denominator between the populations of clusters $\mathcal{C}_j$ and $\mathcal{C}_k$. Define two integers $d_j$ and $d_k$ such that,*

$$n_j = d_j c \quad n_k = d_k c$$

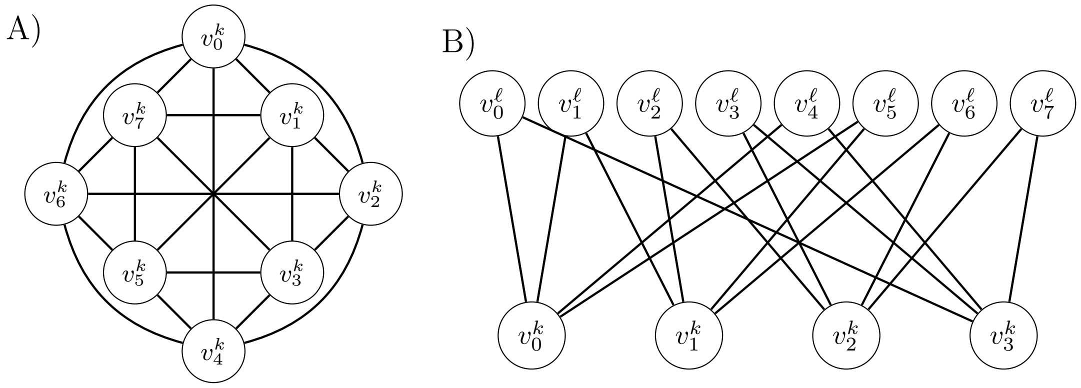


**Figure 4.3:** Examples of the intra-cluster wiring and the inter-cluster wiring. In (A), an example of intra-cluster wiring is shown for $n_k = 8$ and $s_k = 3$. The sequence $b = (1, 2)$. In (B), an example of inter-cluster wiring is shown for $n_k = 4$ and $n_\ell = 8$ with quotient edge weights $W_{k,\ell} = 2$ and $W_{\ell,k} = 4$. The values $c = \gcd(4, 8) = 4$ and $m = 2$ with sequence $b = (1, 3)$.

*Let* $m = \frac{W_{j,k}c}{n_k} = \frac{W_{k,j}c}{n_j}$ *and define a sequence of positive integers* $b = \{b_1, \dots, b_m\}$ *such that,*

$$\sum_{\ell=1}^{m} b_j = c$$

*Then, the set of edges can be written in two equivalent ways, either from* $\mathcal{C}_j$ *to* $\mathcal{C}_k$ *as,*

$$\left(\ell \bmod p_j, \left(\ell + r_1 c + \sum_{a=1}^{r_2} b_a\right) \bmod p_k\right), \qquad \begin{array}{l} r_1 = 0, \dots, d_k - 1 \\ r_2 = 1, \dots, m \end{array} \tag{4.4.7}$$

*or from* $\mathcal{C}_k$ *to* $\mathcal{C}_j$ *as,*

$$\left(\ell' \bmod p_k, \left(\ell' + r_3 c + \sum_{a=1}^{r_4} b_{m-a+1}\right) \bmod n_j\right), \qquad \begin{array}{l} r_3 = 0, \dots, d_j - 1 \\ r_4 = 1, \dots, m \end{array} \tag{4.4.8}$$

The proof is contained in Appendix C.2. An example of the intra-cluster wiring procedure is shown in Fig. 4.3(A) where $pn_k = 8$ and $s_k = 3$. The sequence used is $b = (1, 2)$ and node $s_k$ is odd so each node, $v_\ell^k$ has neighbors

$$\mathcal{N}_\ell = \left\{ \begin{array}{c} (\ell+1) \bmod 8, \quad (\ell-1) \bmod 8, \quad (\ell+2) \bmod 8 \\ (\ell-2) \bmod 8, \quad (\ell+4) \bmod 8 \end{array} \right\}.$$

An example of the inter-cluster wiring procedure is shown in Fig. 4.3(B) where $n_k = 4$

and $n_\ell = 8$ with quotient edge weights $W_{k,\ell} = 2$ and $W_{\ell,k} = 4$. The greatest common denominator is $c = \gcd(4, 8) = 4$ so that $m = 2$. The sequence chosen is $b = (1, 3)$ which can be seen as node $v_0^k$ has neighbors $v_1^\ell$, $v_4^\ell$, $v_5^\ell$ and $v_0^\ell$.

The final step is to prove that the resulting graph has the desired orbits of the automorphism group.

**Theorem 4.4.3** (Orbits of the Automorphism Graph)**.** *The graph constructed using Thms. 4.4.1 and 4.4.2 has symmetries,*

$$\pi(v_\ell^k) = v^k_{(\ell+z) \bmod n_k}, \quad \forall \ell = 0, \ldots, n_k - 1, \quad \forall \mathcal{C}_k \in \mathcal{C} \tag{4.4.9}$$

*for any integer $z \in \mathbb{Z}$. The permutation in Eq.* (4.4.9) *is a* group cyclic permutation. *The set of all symmetries of the form in Eq.* (4.4.9) *make up the automorphism group of the graph with orbits corresponding to those nodes originating from the same cluster.*

*Proof.* From the expression for the intra-cluster edges in Eq. (4.4.6), a group cyclic permutation does not affect the set of edges. Similarly, the inter-cluster edges in Eq. (4.4.7) or in Eq. (4.4.8) are unaffected by the group cyclic permutation. Thus, as the edges are invariant under the group cyclic permutation applied to the nodes, these permutations must be symmetries. The orbits of the automorphism group correspond to the nodes originating from the clusters which can be seen that any two nodes $v_\ell^k$ and $v_{\ell'}^k$ from cluster $\mathcal{C}_k$ there exists a group cyclic permutation with $z = (\ell' - \ell) \bmod n_k$ that maps $\pi(v_\ell^k) = v_{\ell'}^k$. □

The process described in the past couple of sections constructs graphs with the desired orbits of the automorphism group, and whose automorphism group consists of group cyclic permutations. An interesting future extension of this work would be to impose a particular automorphism group along with the desired orbits. To do this, additional constraints must be placed on the properties of the quotient graph as well as the edge wiring process would be less straightforward.

Two examples of this process are shown in Figs. 4.2 and 4.4. In Fig. 4.2(C) each

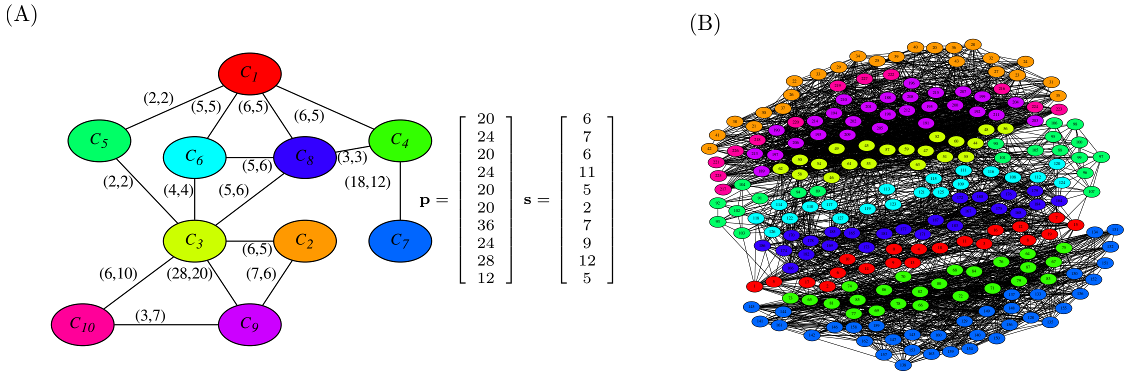


**Figure 4.4:** An example of creating a graph with desired orbits. The feasible quotient graph is shown in (A) with quotient edge weights written along each edge and quotient nodes colored uniquely. The node populations and the intra-cluster degrees are shown as the vectors $\boldsymbol{n}$ and $\boldsymbol{s}$. The resulting graph is shown in (B) with nodes colored according to their originating cluster.

quotient edge is shown with the originating pattern (dashed, solid, dotted) and the resulting edges in the graph below it. The graph is shown in Fig. 4.2(D) with edges patterned according to the originating quotient edges. A larger example is shown in Fig. 4.4 with the quotient graph shown in Fig. 4.4(A) where each quotient node is colored uniquely and the quotient edge weights written along each quotient edge. The cluster populations and intra-cluster degrees are shown as the vectors $\boldsymbol{n}$ and $\boldsymbol{s}$, respectively. The resulting graph is shown in Fig. 4.4(B) with nodes colored according to their originating quotient node.

## 4.5 Orbits vs. Minimal Equitable Partition

It has previously been demonstrated that the orbits of the automorphism group (OAG) represent an equitable partition of the nodes. It is also known [205, 215, 216] that the orbits of the automorphism group is not necessarily the minimal equitable partition (MEP). If the quotient graph is *irreducable*, then the graph constructed using the method described in the previous sections have coinciding orbits of the automorphism group (OAG) and minimal equitable partition (MEP).

To numerically investigate how often the OAG and MBC align, we perform the following procedure.

1. With a feasible quotient graph with cluster populations $\boldsymbol{n}$, we scale the populations with an integer $t$.

2. Construct a graph using the wiring procedure described above.

3. Perform *cluster degree preserving randomization* which is a process by which we choose two edges from the same set of intra-cluster edges $\mathcal{E}_k$ or inter-cluster edges $\mathcal{E}_{j,k}$ and swap one end of the two edges. This procedure preserves the cluster degree of each node but may eliminate some of the symmetries from the automorphism group and thus refines the OAG.

An example is shown in Fig. 4.5(A) and Fig. 4.5(B) where the same graph is shown but the nodes are colored according to its MEP in panel (A) where $|\mathcal{C}| = 2$ and its OAG in panel (B) where $|\mathcal{O}| = 10$, i.e., the orbits are all *trivial.* Let $\mathcal{O}$ be the OAG and let $\mathcal{C}$ be the MEP, and define the function,

$$f(\mathcal{O}) = \frac{n - |\mathcal{O}|}{n - |\mathcal{C}|} \tag{4.5.1}$$

where $f(\mathcal{O}) = 1$ if $\mathcal{O} = \mathcal{C}$ and $f(\mathcal{O}) = 0$ if $\mathcal{O}$ consists of only trivial orbits, that is, an orbit $\mathcal{O}_k \in \mathcal{O}$ such that $|\mathcal{O}_k| = 1$. An example of this process is shown in Fig. 4.5(C) for the two quotient node quotient graph shown in the inset. For each value of $t$, we generate 1000 graphs and compute their MEP and OAG and compute $f(\mathcal{O})$. The black curve plots the mean value of $f(\mathcal{O})$ and the error bars represent one standard deviation. The blue curve is the minimum value of $f(\mathcal{O})$ and the red curve is the maximum value of $f(\mathcal{O})$. We see that for $t = 1$, the MEP and OAG coincide and so $f(\mathcal{O}) = 1$. As $t$ increases, the average value of $f(\mathcal{O})$ and the maximum value of $f(\mathcal{O})$ over all 1000 realizations decrease rapidly. This means that for $t > 1$, less than 0.1% of graphs with the two node quotient graph shown in the inset will have their OAG and MEP coincide. This example, typical of all other quotient graphs examined, provides evidence that the OAG and the MEP rarely coincide rather than the expected result that they usually coincide.

In the next section, we show that the result above can, by extending the definition

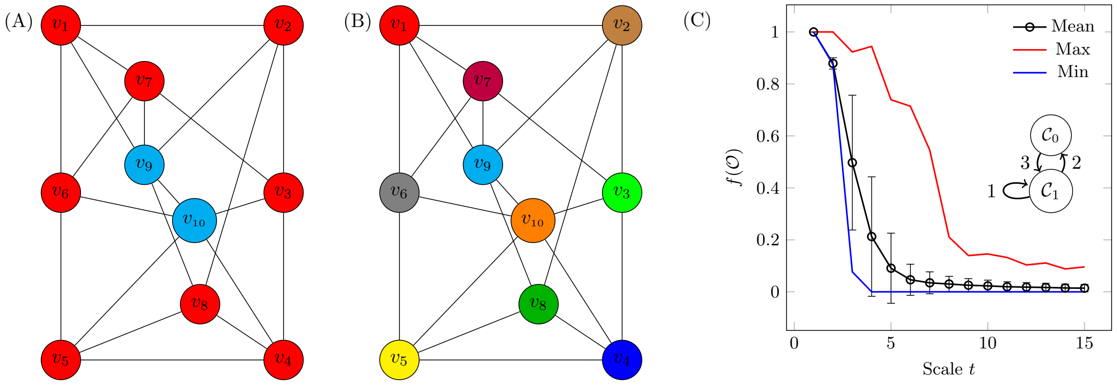


**Figure 4.5:** Examples of cases when the OAG and the MBC are not the same. (A) A 10 node graph with nodes colored according to its minimum equitable partition which consists of two orbits. (B) The same graph but with nodes colored according to the orbits of its automorphism group. There is no orbit with more than a single node in it, i.e., there is only one symmetry, the identity permutation. (C) A numerical example for the relative sizes of the OAG and the MEP using Eq. (4.5.1). The quotient graph in the inset is used to generate graphs with two orbits such that the populations of the two orbits is scaled by $t$. The red curve is the maximum value of $f(\mathcal{O})$ found over 1000 realizations and the blue curve is the minimum.

of a quotient graph, be directly applied to multi-layer networks, which may be more useful to describe certain types of systems.

## 4.6 Symmetries in Multi-Layer Networks [15]

Most generally, a multi-layer network can be described using the following definition.

**Definition 4.6.1** (Multi-Layer Graphs). *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a* multi-layer graph *by which it is meant there is a partition of the nodes into $n_\ell$ layers, denoted,*

$$\mathcal{V} = \bigtimes_{k=1}^{n_\ell} \mathcal{V}_\ell$$

*Additionally, let the set of edges be similarly split,*

$$\mathcal{E} = \bigtimes_{j,k=1}^{n_\ell} \mathcal{E}_{j,k}$$

*where an edge $(v_a, v_b) \in \mathcal{E}_{j,k}$ if $v_a \in \mathcal{V}_j$ and $v_b \in \mathcal{V}_k$. Each set of edges may be further*

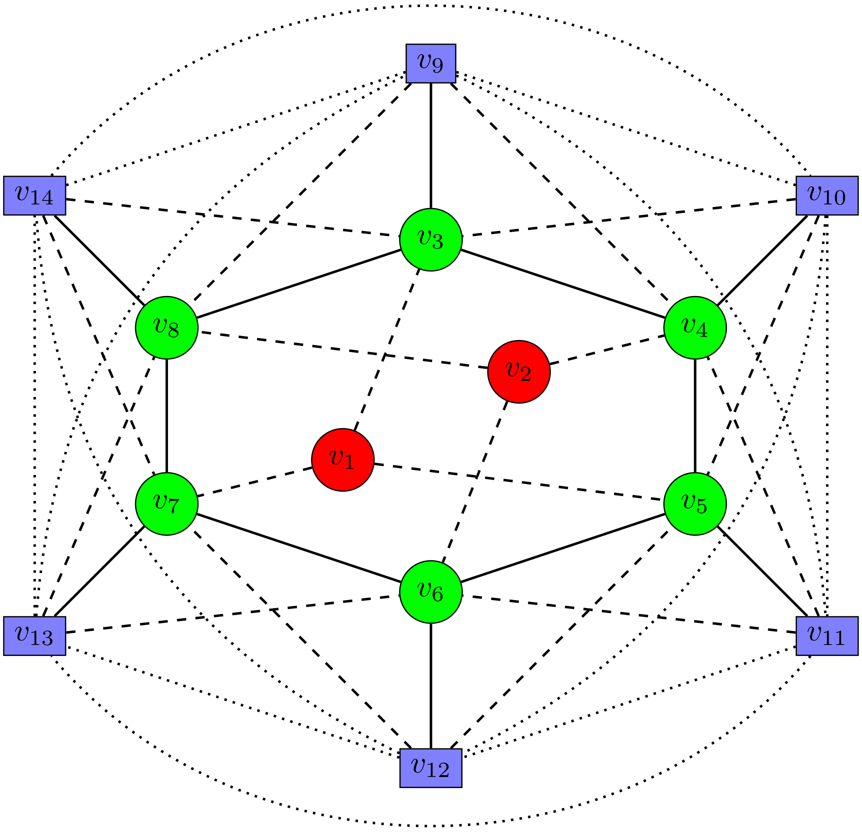


**Figure 4.6:** A multi-layer graph with $n_\ell = 2$ layers (denoted as circles and rectangles) with $m = 3$ edge types, but with $m_\ell = 2$ (denoted as solid, dashed, or dotted). The nodes are colored according to its automorphism group.

*refined to $m_{j,k}$ different types so that,*

$$\mathcal{E}_{j,k} = \bigtimes_{i=1}^{m_{j,k}} \mathcal{E}_{j,k}^{(i)}$$

*The effective number of edge types is denoted $m_\ell = \max m_{j,k}$, i.e., the largest number of edge types between any pair of node types.*

An example of a multi-layer graph is shown in Fig. 4.6 with two types of nodes, $n_\ell = 2$, where node types are denoted by the shape. There are three edge types, denoted by solid, dashed, or dotted lines, but at most there are only $m_\ell = 2$ edge types at most between any two types of nodes. The quotient graph of this multi-layer graph is shown in Fig. 4.7 where the quotient nodes are colored and have shape corresponding to their originating nodes and the quotient edges have pattern

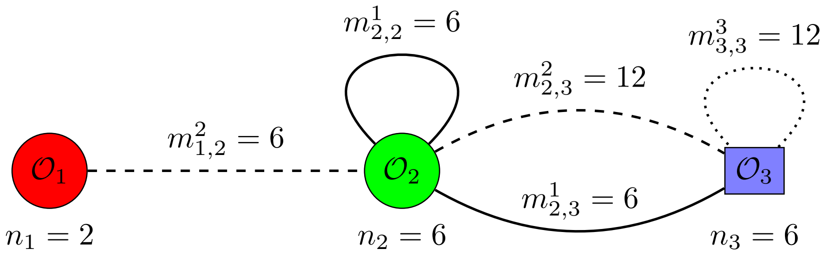


**Figure 4.7:** The quotient graph of the multi-layer graph in Fig. 4.6. Nodes are colored and have shape according to their originating orbit. Edges have pattern according to their originating edge types as well.

corresponding to their originating edges.

The definition of a quotient graph in Def. 4.2.1 can be directly applied to the multi layer networks with the caveat that each quotient node in $\mathcal{C}$ now has a node type associated with it and each edge $(\mathcal{C}_j, \mathcal{C}_k)$ likewise has an edge type associated with it. The procedure to create feasible quotient graphs remains the same except that now there may be more than one quotient edge connecting two quotient nodes of different types.

In the following section, we turn to an application of graphs with symmetries, namely, the consensus problem. This problem addresses how linear dynamics applied to graphs with symmetry can exhibit interesting consensus states, even if the system is unstable.

## 4.7 Symmetry Induced Group Consensus [12]

As an application of the use of graphs with symmetry, we briefly discuss *consensus problems* and in particular, *group consensus.* In consensus problems, we assign to each node in a graph $v_j \in \mathcal{V}$, a time-varying state vector, $\boldsymbol{x}_j(t)$, and consensus is achieved if,

$$\lim_{t\to\infty} ||\boldsymbol{x}_j(t) - \boldsymbol{x}_k(t)|| = 0, \quad \forall v_j, v_k \in \mathcal{V} \tag{4.7.1}$$

Consensus type problems as described by Eq. (4.7.1) find applications in vehicle coordination [217], opinion dynamics [218], sensor networks [219], and communication [220]. More recently, cluster consensus and group consensus (sometimes used interchangeably) have been investigated which is defined for some partition of the nodes, $\mathcal{C}$, as,

$$\lim_{t\to\infty} ||\boldsymbol{x}_j(t) - \boldsymbol{x}_k(t)|| = 0, \quad \forall v_j, v_k \in \mathcal{C}_\ell, \quad \forall \mathcal{C}_\ell \in \mathcal{C} \tag{4.7.2}$$

Group consensus, as defined in Eq. (4.7.2), for undirected and directed graphs with and without topology switching is investigated in [221–223]. The intra-group coupling is used as the mechanism for determining whether group consensus will occur in [224–226]. Most of the current work on group consensus [227–230] assumes that the adjacency matrix is *balanced*, that is, the inter-group coupling for each node sums to zero, that is,

$$\sum_{v_k \in \mathcal{C}_b} A_{j,k} = 0, \quad v_j \in \mathcal{C}_a, \quad a \neq b$$

The works referenced above do not exploit intrinsic properties of the graph structure which is different from our work which uses the automorphism group of the graph. The only work on consensus, or synchronization, using the automorphism group of a graph, applies contraction theory [231, 232] which does not provide insight into the role control gains may have in whether or not group consensus can be achieved.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with adjacency matrix $A$ and $n$ nodes. We assume the edge weights $w(v_j, v_k) = 1 \ \forall (v_j, v_k) \in \mathcal{E}$. Let $\text{Aut}(\mathcal{G})$ be the automorphism group of the graph and let $\mathcal{O}$ be the partition of the nodes induced by the orbits of the automorphim group of the graph (as described in Lemma 4.2.1). We assume that there are $|\mathcal{O}| = q$ orbits. To determine the orbit of each node, we use the notation,

$$\bar{k} = \ell \quad \text{if} \quad v_k \in \mathcal{O}_\ell \tag{4.7.3}$$

We are interested in linear dynamics described by the equation,

$$\dot{\boldsymbol{x}}_j(t) = F_{\bar{j}} \boldsymbol{x}_j(t) + \sum_{k=1}^{N} A_{j,k} H \boldsymbol{x}_k(t). \tag{4.7.4}$$

where each node's state vector is composed of $N$ variables. Let $\boldsymbol{x}(t)$ be the composite vector of all states of the nodes in the graph which has length $Nn$, then the evolution of the states for all systems governed by Eq. (4.7.4) can be written,

$$\dot{\boldsymbol{x}}(t) = \left[\sum_{k=1}^{q} J_k \otimes F_k + A \otimes H\right] \boldsymbol{x}(t) \tag{4.7.5}$$

where $J_k$ is a diagonal matrix with diagonal elements $\ell, \ell$ equal to 1 if $\bar{\ell} = k$ as defined in Eq. (4.7.3) and equal to 0 otherwise. The symbol $\otimes$ denotes the Kronecker product.

**Definition 4.7.1** (Group Consensus Manifold)**.** *The set of states $\boldsymbol{x}_j(t) = \boldsymbol{x}_k(t)$ for all $\bar{j} = \bar{k}$ defines an invariant manifold of the dynamics in Eq.* (4.7.5)*.*

If the system lies on the group consensus manifold, then the evolution of the system can equivalently be described by the quotient graph dynamics,

$$\dot{\boldsymbol{q}}_k(t) = F_k \boldsymbol{q}_k(t) + \sum_{\ell=1}^{q} Q_{k,\ell} H \boldsymbol{q}_\ell(t), \quad k = 1, \ldots, q \tag{4.7.6}$$

where the $Q \in \mathbb{R}^{q \times q}$ is the adjacency matrix of the quotient graph. The matrix $Q$ can be constructed as,

$$Q = E^\dagger A E$$

where $E \in \{0,1\}^{n \times q}$ is the orbit indicator matrix with elements $E_{k,\bar{k}} = 1$ for $k = 1, \ldots, n$ and all other elements equal to 0. The symbol † represents the Moore-Penrose pseudo-inverse defined as,

$$E^\dagger = (E^T E)^{-1} E^T$$

The quotient dynamics in Eq. (4.7.6) is *stable* if the matrix,

$$(F_1 \oplus F_2 \oplus \ldots \oplus F_q) + Q \otimes H$$

is Hurwitz.

We separate four possible cases of interest:

1. The system in Eq. (4.7.5) is unstable and group consensus is not achieved.

2. The system in Eq. (4.7.5) is unstable and group consensus is achieved.

3. The system in Eq. (4.7.5) is stable and group consensus is achieved.

4. The quotient dynamics in Eq. (4.7.6) is stable but the group consensus manifold is not.

To determine which case will occur for a particular choice of matrices $F_k$, $k = 1, \dots, q$, adjacency matrix $A$, and coupling matrix $H$, we use the automorphism group block diagonalizing orthogonal transformation [204], $T \in \mathbb{R}^{n \times n}$. The transformation is determined from the irreducible representation (IRR) of the automorphism group of the network. The first $q$ rows of $T$ consists of elements,

$$T_{k,\ell} = \begin{cases} \sqrt{\frac{1}{n_k}} & \text{if } \bar{k} = \ell \\ 0 & \text{otherwise} \end{cases}$$

These first $q$ rows describe the behavior along the group consensus manifold. The remaining $n - q$ rows describe motion *orthogonal* to the group consensus manifold and all have row-sum equal to zero. Each row of $T$ has non-zero values associated with columns corresponding to a single orbit. The transformation applied to the adjacency matrix yields the matrix $TAT^T = B$ and the transformation applied to $J_k$ is invariant, i.e., $TJ_kT^T = J_k$. Define the transformed states of the system as $\tilde{\boldsymbol{x}}(t) = (T \otimes I_n)\boldsymbol{x}(t)$ so that the transformed version of Eq. (4.7.5) is,

$$\dot{\tilde{\boldsymbol{x}}}(t) = \left[ \sum_{k=1}^{q} J_k \otimes F_k + B \otimes H \right] \tilde{\boldsymbol{x}}(t) = \hat{B}\tilde{\boldsymbol{x}}(t) \tag{4.7.7}$$

This transformation has important consequences to determine the stability of group consensus. The first $q$ transformed states in Eq. (4.7.7) are denoted $\tilde{\boldsymbol{x}}_{para}(t)$ and represent motion along the group consensus manifold and the remaining $n-q$ transformed states are denoted $\tilde{\boldsymbol{x}}_{orth}(t)$ and represent motion orthogonal to the group consensus manifold. The behavior of these two components have been decoupled, that is,

$$\begin{bmatrix} \dot{\tilde{\boldsymbol{x}}}_{para}(t) \\ \dot{\tilde{\boldsymbol{x}}}_{orth}(t) \end{bmatrix} = \begin{bmatrix} \hat{B}_{para} & O_{nq \times n(N-q)} \\ O_{n(N-q) \times nq} & \hat{B}_{orth} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{x}}_{para}(t) \\ \tilde{\boldsymbol{x}}_{orth}(t) \end{bmatrix} \tag{4.7.8}$$

We can now determine the stability of the group consensus manifold independent of the stability of the global system. Let $\lambda_{para}^{\max}$ be the largest real part of an eigenvalue of the matrix $\hat{B}_{para}$ and let $\lambda_{orth}^{\max}$ be the largest real part of an eigenvalue of the matrix $\hat{B}_{orth}$ that appear in Eq. (4.7.8). If $\lambda_{para}^{\max} < 0$ ($> 0$), then the motion along the group consensus manifold is stable (not stable) while if $\lambda_{orth}^{\max} < 0$ ($> 0$), then perturbations from the group consensus manifold will vanish with increasing time (will grow with increasing time).

An example of the automorphism group consensus problem for the twelve node graph shown in Fig. 4.8(A) is described in depth. Each node is assigned $n = 4$ states. The quotient graph is shown in Fig. 4.8(B) with associated matrices,

$$F_1 = \begin{bmatrix} O_2 & I_2 \\ -2I_2 & -2I_2 \end{bmatrix} \quad F_2 = \begin{bmatrix} O_2 & I_2 \\ -4I_2 & -4I_2 \end{bmatrix} \quad F_3 = \begin{bmatrix} O_2 & I_2 \\ -6I_2 & -6I_2 \end{bmatrix}$$

and the coupling matrix has two control parameters $\alpha$ and $\beta$,

$$H = \begin{bmatrix} O_2 & O_2 \\ \alpha I_2 & \beta \mathbf{1}_2 \mathbf{1}_2^T \end{bmatrix} \tag{4.7.9}$$

The adjacency matrix with rows colored according to each node's orbits is shown in Fig. 4.9. The orthogonal block diagonalizing transformation $T$ is shown in Fig. 4.10 and the resulting block diagonal matrix $B$ is shown in Fig. 4.11. Note that the upper left hand corner represents the motion along the group consensus manifold while the lower right hand corner represents the blocks describing motion orthogonal to the group consensus manifold. Using the control parameters $\alpha$ and $\beta$ in Eq. (4.7.9), we examine the stability of the system and whether or not group consensus is achieved which is colored in Fig. 4.8(C). We see all four possibilities occur for different values of the control parameters where red corresponds to stable global dynamics and group consensus occurs, the purple region corresponds to unstable dynamics but group consensus is achieved, the green region corresponds to unstable dynamics and group consensus is not achieved, and the blue strip corresponds to stable dynamics

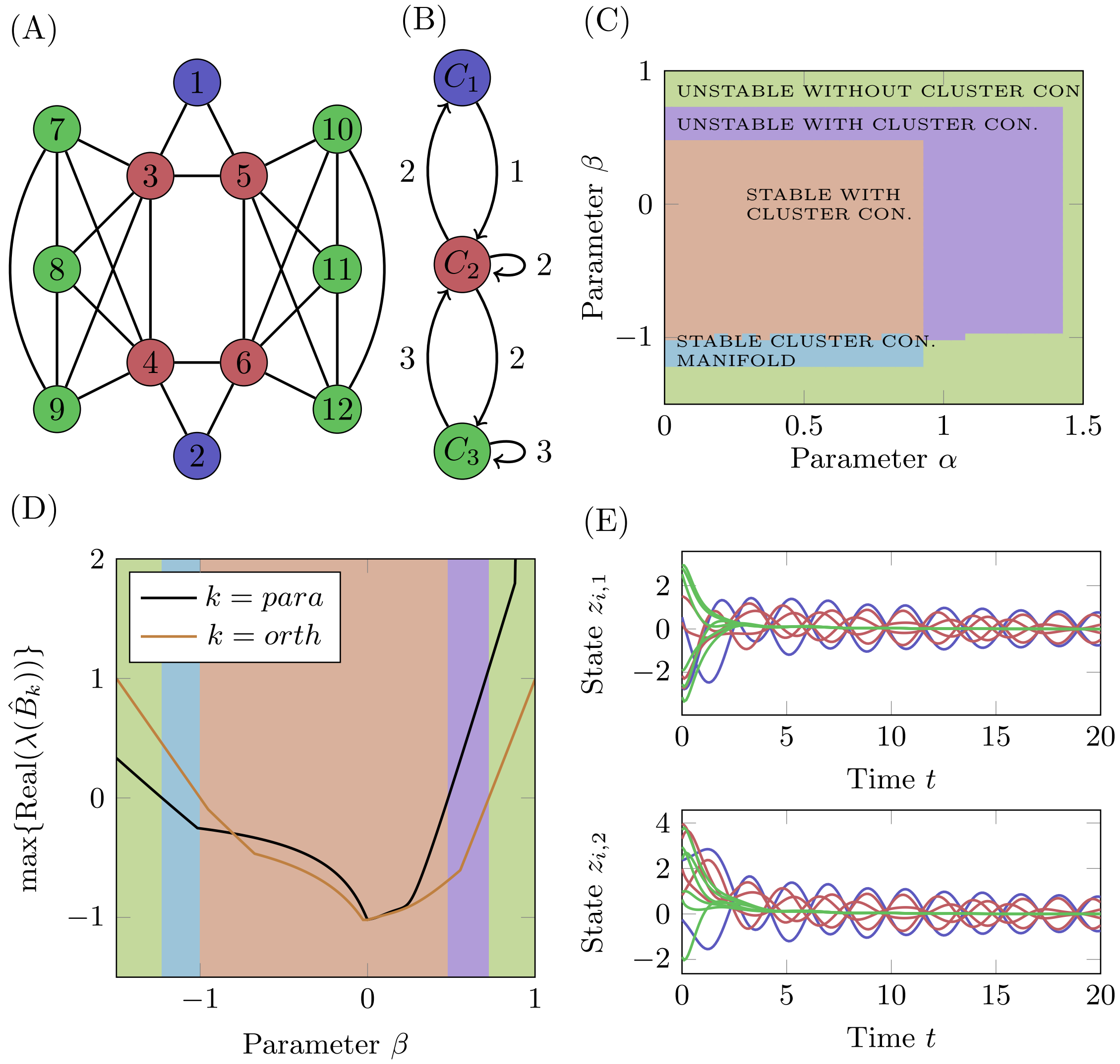


**Figure 4.8:** An example of the different regimes which may exist in the group consensus problem. (A) A twelve node graph with three orbits in its automorphism group, with nodes colored according to their orbits. (B) The quotient graph associated with the orbits of the automorphism group of the graph in (A). (C) The stability of the system in control parameter space ($\alpha$,$\beta$) where the red background represents stable dynamics and group consensus achieved, the purple background represents unstable dynamics but group consensus achieved, the green background represents unstable dynamics and group consensus is not achieved, and the blue background represents stable dynamics but group consensus is not achieved. (D) The largest real part of the eigenvalues of the matrices $\hat{B}_{para}$ and $\hat{B}_{orth}$ for $\alpha = 0.2$ and varying balues of $\beta$. The background is colored according to the regions in (C). (E) An example of the time trajectories in the blue region where $\hat{B}_{para}$ is negative definite but $\hat{B}_{orth}$ is marginally stable so the system is marginally stable but group consensus is not achieved.

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

**Figure 4.9:** The adjacency matrix of the graph in Fig. 4.8(A) with rows colored according to each node's orbit.

$$T = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1/\sqrt{6} & 1/\sqrt{6} & 1/\sqrt{6} & 1/\sqrt{6} & 1/\sqrt{6} & 1/\sqrt{6} \\ 0 & 0 & 0.5 & 0.5 & 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/\sqrt{2} & 1/\sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2/\sqrt{6} & 1/\sqrt{6} & 1/\sqrt{6} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2/\sqrt{6} & 1/\sqrt{6} & 1/\sqrt{6} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & -1/\sqrt{6} & -1/\sqrt{6} & -1/\sqrt{6} & 1/\sqrt{6} & 1/\sqrt{6} & 1/\sqrt{6} \\ 0 & 0 & -0.5 & -0.5 & 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.5 & 0.5 & 0.5 & -0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.5 & 0.5 & -0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

**Figure 4.10:** The block diagonalizing transformation matrix found from the IRR of the graph in Fig. 4.8(A). The rows are colored according to the associated orbit.

but group consensus is not achieved. Values of $\lambda_{para}^{\max}$ and $\lambda_{orth}^{\max}$ are shown in Fig. 4.8(D) for $\alpha = 0.2$ and varying $\beta$. The regions are colored according to the same criteria as in Fig. 4.8. In Fig. 4.8(E) time trajectories of the states are shown for $\alpha = 0.2$ and $\beta = -1$ so that the system is marginally stable but group consensus is not achieved. This example, different from the previous work on group consensus, demonstrates that it is possible for a dynamical system to be unstable, yet still achieve group consensus.

This work focuses on the problem of whether or not group consensus may occur whether or not the dynamics of the overall system is stable or not. Note that the block diagonal matrix $\hat{B}_{orth}$ though is further decomposed into smaller blocks representing different orthogonal motion from the group consensus manifold. This suggests it is possible to determine if some orbits achieve consensus while others may not in a more granular fashion. This focus will guide future work in this area.

$$B = \begin{pmatrix}
2 & \sqrt{6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\sqrt{6} & 2 & \sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & \sqrt{6} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{6} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{2} & 0
\end{pmatrix}$$

**Figure 4.11:** The block diagonal matrix $B = TAT^T$ where $A$ and $T$ are shown in Figs. 4.9 and 4.10, respectively. The upper block is the matrix $B_{para}$ which describes motion parallel to the group consensus manifold while the remaining blocks describe motion orthogonal to the group consensus manifold.

## 4.8 Approximate Equitable Partitions

In the previous sections, our discussion concerned symmetries of graphs, a structural property of a graph (that is, independent of edge weights). By ignoring the edge weights though, information about a network is lost and so the orbits of the automorphism group may not represent redundancies in the network as described in Sec. 4.2 or may not capture those nodes with similar behavior as described in Sec. 4.7. The obvious question though is how best to include edge weights. To discuss the various options, we repeat some definitions. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph with edge weights $w : \mathcal{E} \mapsto \mathbb{R}^+$. Also, let $\mathcal{C}$ be a partition of the nodes. We differentiate two quantities, the *cluster degree* of a node and the *cluster strength* of a node.

**Definition 4.8.1** (Cluster Degree and Cluster Strength)**.** *Let $n_c = |\mathcal{C}|$ be the number of clusters in the partition $\mathcal{C}$. The cluster neighbors of node $v_j$ is,*

$$\mathcal{N}_{j,k} = \{v_\ell \in \mathcal{C}_k | (v_j, v_\ell) \in \mathcal{E}\}$$

*The cluster degree of node $v_j$, denoted $\boldsymbol{\kappa}_j \in \mathbb{Z}^{n_c}$, has elements equal to,*

$$\boldsymbol{\kappa}_{j,k} = |\mathcal{N}_{j,k}|$$

*and the cluster strength of a node $v_j$, denote $\boldsymbol{\sigma}_j \in \mathbb{R}^{n_c}$, has elements equal to,*

$$\boldsymbol{\sigma}_{j,k} = \sum_{v_\ell \in \mathcal{N}_{j,k}} w_{j,\ell}$$

An equitable partition, as defined in Def. 2.2.20, has the property that, for every pair of nodes $v_j, v_k \in \mathcal{C}_\ell$, $\boldsymbol{\kappa}_j = \boldsymbol{\kappa}_k$, but it is not necessary for $\boldsymbol{\sigma}_j = \boldsymbol{\sigma}_k$. Some possible extensions of equitable partitions and symmetries to weighted graphs are now discussed.

1. One can shoehorn the definition of an equitable partition to the cluster strength vectors such that $\mathcal{C}$ is an equitable partition if for every pair of nodes $v_j, v_k \in \mathcal{C}_\ell$, then $\boldsymbol{\sigma}_j = \boldsymbol{\sigma}_k$. Note that this definition will only give non-trivial results if the edge weights are drawn from some relatively small finite set of values. If the edge weights are drawn from some continuous distribution, then the probability that any sum of a subset of the edge weights is equal to the sum of any other subset of the edges is essentially zero.

2. If the edges can be classified into $n_e$ sets, that is, $\mathcal{E} = \bigcup_{\ell=1}^{n_e} \mathcal{E}_k$, then one can compute an *edge-colored equitable partition.* Define cluster degrees associated with each set of edges, $\boldsymbol{\kappa}_j^{(\ell)}$, $\forall v_j \in \mathcal{V}$, $\forall \mathcal{E}_\ell \in \mathcal{E}$. The partition $\mathcal{C}$ is an edge-colored equitable partition if,

$$\boldsymbol{\kappa}_j^{(\ell)} = \boldsymbol{\kappa}_k^{(\ell)}, \quad \forall v_j, v_k \in \mathcal{C}_a, \quad \forall \mathcal{C}_a \in \mathcal{C}, \quad \forall \mathcal{E}_\ell \in \mathcal{E}$$

   This definition again ignores any edge weights, but may be applicable for networks with edges that perform different tasks, such as communication links and information links.

Rather than these possibilities, we relax the strict equality demanded by the previous definitions and instead introduce a tolerance.

**Definition 4.8.2** (Approximate Equitable Partition)**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph with weights $w : \mathcal{E} \mapsto \mathbb{R}$, which we assume are drawn from a continuous* *distribution. A partition of the nodes, $\mathcal{C}$, is approximately equitable if for every pair of nodes $v_j, v_k \in \mathcal{C}_\ell$, their cluster strengths satisfy,*

$$||\boldsymbol{\sigma}_j - \boldsymbol{\sigma}_k|| \leq \epsilon \tag{4.8.1}$$

*where $||\cdot||$ is some norm chosen based on the application.*

Of particular interest is the minimum approximate equitable partition, that is, the partition $\mathcal{C}$ that satisfies Def. 4.8.2 that is of minimum size, $|\mathcal{C}| = n_c$.

$$\begin{aligned} \min \quad & |\mathcal{C}| \\ \text{s.t.} \quad & ||\boldsymbol{\sigma}_j - \boldsymbol{\sigma}_k|| \leq \epsilon, \quad \forall v_j, v_k \in \mathcal{C}_\ell, \quad \forall \mathcal{C}_\ell \in \mathcal{C} \end{aligned} \tag{4.8.2}$$

The minimum approximate equitable partition depends on the chosen value of $\epsilon$, and so we denote it as $\mathcal{C}(\epsilon)$. Note that $\mathcal{C}(\epsilon)$, the minimum approximate equitable partition for a specific value of $\epsilon$, is not necessarily unique. There exists $n$ values of $\epsilon$, denoted $\epsilon_k$, $k = 1, \ldots, n$ of particular interest which are defined as

$$\begin{aligned} |\mathcal{C}(\epsilon)| > k \quad & \text{if } \epsilon < \epsilon_k \\ |\mathcal{C}(\epsilon)| \leq k \quad & \text{if } \epsilon \geq \epsilon_k \end{aligned} \tag{4.8.3}$$

These values of $\epsilon_k$ are the switching values when the MAEP increases in cardinality. For the lower bound, $\epsilon_n = 0$, can be trivially proven using Eq. (4.8.3). At the other end, one can compute $\epsilon_1$ using Def. 4.8.2 directly. Compute the strengths of every node, $\sigma_j$, and then,

$$\epsilon^{\max} = \max_{v_j, v_k \in \mathcal{V}} |\sigma_j - \sigma_k| \tag{4.8.4}$$

If the graph is small enough, one can enumerate all possible partitions. To determine when this is possible, we define some results from the field of combinatorics.

**Definition 4.8.3** (Stirling Numbers of the Second Kind [233]). *The Stirling numbers of the second kind, denoted $\left\{ n \atop k \right\}$, are equal to the number of ways a set of $n$ items can* *be partitioned into $k$ subsets. An explicit formular to compute them is,*

$$\left\{ {n \atop k} \right\} = \frac{1}{k!} \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} j^n$$

Even for larger values of $n$, one can enumerate all partitions for the first few values of $k$ as the Stirling numbers scale approximately as $k^n$. This exercise can be useful to find the first few values of $\epsilon_k$, $k = 2, 3, \ldots$.

**Definition 4.8.4** (Bell Numbers)**.** *The Bell numbers, denoted $B_n$, are equal to the number of ways a set of $n$ items can be partitioned into any number of sets, i.e.,*

$$B_n = \sum_{k=0}^{n} \left\{ {n \atop k} \right\}$$

*that is, a sum of the Stirling numbers of the second kind defined in Def. 4.8.3.*

For $n > 14$, $B_n > 10^9$, after which enumerating all partitions may become computationally infeasible. A small example demonstrates the above concepts.

A star graph with one central node, $v_1$, and 10 leaves, $v_k$, $k = 2, \ldots, 11$, is constructed as shown in Fig. 4.12(A). For this example, the MAEP is computed for different values of $\epsilon$ which is shown in Fig. 4.12(B). The discrete behavior of $\epsilon_k$ is seen as the steps. For instance, $\epsilon_{10} = 0.01$, which is the value of $\epsilon$ that, if $\epsilon < \epsilon_{10}$, every node is placed in its own cluster in the MAEP. The unweighted minimum equitable partition is recovered for $\epsilon_3 < \epsilon \leq \epsilon_2$ where $\epsilon_3 = 0.62$ and $\epsilon_2 = 10.59$.

An additional difficult of this problem is that it does not obey *transitivity*. Specifically, if nodes $v_j$ and $v_k$ obey Eq. (4.8.1) and nodes $v_k$ and $v_\ell$ obey Eq. (4.8.1), it may be the case that nodes $v_j$ and $v_\ell$ do not. As an example, two partitions that are optimal solutions to Eq. (4.8.2) for the choice of $\epsilon - 0.5$ are shown in Fig. 4.12(C). For this value of $\epsilon$, nodes $v_2$, $v_4$, and $v_{10}$ cannot be in the same cluster as node $v_{11}$. On the other hand, all of the remaining leaf nodes may be assigned to either cluster indiscriminately, meaning that there are $\binom{6}{2} = 15$ possible, equally optimal, partitions.

To better visualize the change in the partitions, we define a metric on the space

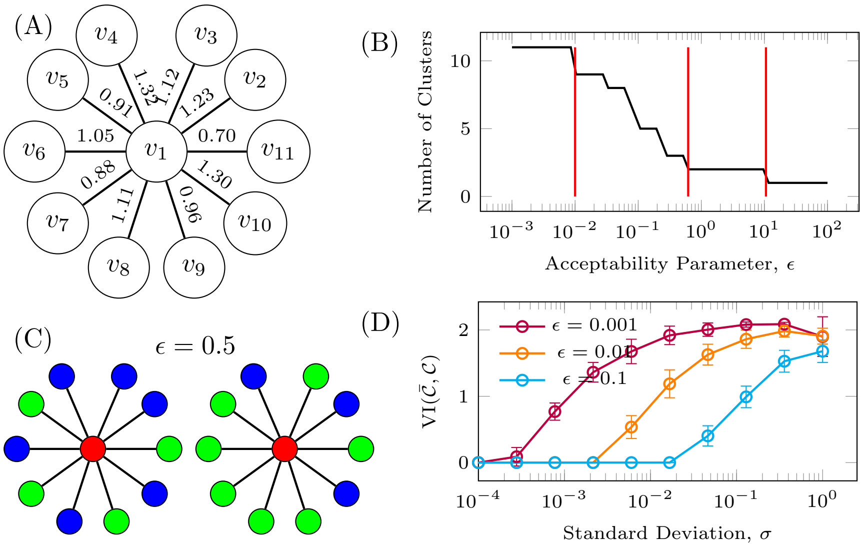


**Figure 4.12:** An example of the definition used for minimum approximate equitable partitions. (A) A star graph with 10 leaf nodes with edge weights applied to each edge. (B) The number of clusters in the MAEP as a function of $\epsilon$. The discrete values of $\epsilon_k$ are seen as the steps in the number of clusters. (C) An example of the non-uniqueness of solutions for the graph in (A) when $\epsilon = 0.5$. (D) The role of $\epsilon$ when the edge weights are drawn from more or less narrow distributions represented by $\sigma$, the standard deviation of a normal distribution.

of partitions called the *Variation of Information.*

**Definition 4.8.5** (Variation of Information [234])**.** *Let $\mathcal{C}$ be a partition of a set of $n$ items into $n_c$ clusters labeled $\mathcal{C}_j$, $j = 1, \ldots, n_c$. The entropy of a partition is defined as,*

$$H(\mathcal{C}) = -\sum_{j=1}^{n_c} p_j \log p_j \tag{4.8.5}$$

*where $p_j = \frac{|\mathcal{C}_j|}{n_c}$ is the probability an item is in the $j$'th cluster. Note that the entropy in Eq.* (4.8.5) *is bounded by,*

$$0 \leq H(\mathcal{C}) \leq \log n$$

*The mutual information shared between two clusterings is defined as,*

$$I(\mathcal{C}, \mathcal{C}') = \sum_{j=1}^{n_c} \sum_{k=1}^{n'_c} p_{j,k} \log \frac{p_{j,k}}{p_j p_k} \tag{4.8.6}$$

*where $p_{j,k} = |\mathcal{C}_j \cap \mathcal{C}_k|$ and the logarithm function is modified to satisfy* $\log(0) = 0$. *If $\mathcal{C} = \mathcal{C}'$ it is easy to show that $I(\mathcal{C}, \mathcal{C}) = 2H(\mathcal{C})$. The Variation of Information between* *two partitions of the same set can be written in terms Eqs.* (4.8.5) *and* (4.8.6).

$$VI(\mathcal{C},\mathcal{C}') = H(\mathcal{C}) + H(\mathcal{C}') - 2I(\mathcal{C},\mathcal{C}')$$

To move beyond a single set of edge weights as shown in Fig. 4.12(A), we instead select edge weights from a normal distribution with mean $\mu = 1$ and standard deviation $\sigma > 0$. A value of interest is $\epsilon_3$ which is the value at which point the unweighted minimum equitable partition is no longer the minimum approximate equitable partition. To determine this value of $\epsilon$, we apply Eq. (4.8.1) for the star graph for the minimum equitable partiton yields $|d_j - d_k| \leq \epsilon$ for $j, k = 2, \dots, 11$. The distribution of the absolute value of the difference of random variates drawn from normal distribution $\mathcal{N}(\mu, \sigma)$ is the half normal distribution which has CDF $\text{erf}\left(\frac{x}{\sqrt{2}\sigma}\right)$. Thus, the probability that a pair of edges satisfies the requirement to be in the same cluster is,

$$P(|d_j - d_k| \leq \epsilon) = \text{erf}\left(\frac{\epsilon}{\sqrt{2}\sigma}\right) \tag{4.8.7}$$

For a star graph with $n$ leaves, every pair of leaf nodes must satisfy Eq. (4.8.7), for which we must check $(n-1)!$ pairs. While not strictly independent, a rough estimate assumes each pair $(d_j, d_k)$ is independent so that we may use the joint probability formula,

$$\text{erf}^{(n-1)!}\left(\frac{\epsilon}{\sqrt{2}\sigma_c}\right) = 0.995 \tag{4.8.8}$$

so that 99.5% of realizations will have the minimum equitable partition as its MAEP. Solving Eq. (4.8.8) for the critical value of $\sigma$ holding $\epsilon$ constant, we find that,

$$\sigma_c = \frac{\epsilon}{\sqrt{2}} \left[\text{erf}^{-1}(0.995^{1/(n-1)!})\right]^{-1} \tag{4.8.9}$$

In Fig. 4.12(D), we plot the Variation of Information, $VI(\mathcal{C},\bar{\mathcal{C}})$ from Def. 4.8.5, where $\bar{\mathcal{C}}$ is the minimum equitable partition of the unweighted graph for $\epsilon = 0.001$, $\epsilon = 0.01$, and $\epsilon = 0.1$. For each value of $\sigma$, 50 realizations of edges are drawn from the corresponding distribution and the minimum approximate equitable partition is

| $\epsilon$ | $\sigma_c$ |
|---|---|
| 0.1 | 0.0176 |
| 0.01 | 0.00176 |
| 0.001 | 0.000176 |

**Table 4.1:** Critical values of $\sigma_c$ for $n = 10$.

computed. For $n = 10$ as used to compute the results in Fig. 4.12(D), we can compute the critical value of $\sigma_c$, The results in Table 4.1 agree with the curves in Fig. 4.12(D) despite the inapropriate assumption of independent variables discussed above.

Unfortunately, for complex networks, the simple analysis of star graphs cannot be applied. Instead, other methods must be developed in order to search over the set of partitions more efficiently. A first approach is to use an ILP formulation of the problem which is amenable to general purpose solvers.

### 4.8.1 ILP Formulation of MAEP

The minimum approximate equitable partition can be formulated as an Integer Linear Program (ILP). In this section, for the ILP formulation, the norm used in Eq. (4.8.2) is chosen to be the inf-norm.

$$||\boldsymbol{\sigma}_j - \boldsymbol{\sigma}_k||_\infty = \max |\sigma_{j,\ell} - \sigma_{k,\ell}| \leq \epsilon$$

This choice is made because it can be constructed as a set of linear constraints.

$$\sigma_{j,\ell} - \sigma_{k,\ell} \leq \epsilon, \quad \ell = 1, \ldots, |\mathcal{C}|$$

Let $n$ be the number of nodes and let $n_c \leq n$ be the maximum allowed number of clusters. The variables $y_{j,k} \in \{0, 1\}$, $j = 1, \ldots, n$, $k = 1, \ldots, n_c$, is equal to 1 if node $v_j$ is in cluster $\mathcal{C}_k$ and is equal to 0 otherwise. We introduce additional variables $z_{i,j} \in \{0, 1\}$, $1 \leq i < j \leq n$, which is equal to 1 if nodes $v_i$ and $v_j$ are in the same cluster and 0 otherwise. To find the values of $\epsilon_{n_c}$ for a particular value of $n_c$,

$$\begin{aligned}
\min \quad & \epsilon \\
\text{s.t.} \quad & \sum_{k=1}^{n_c} y_{j,k} = 1, \quad \forall v_j \in \mathcal{V} \\
& \sum_{v_\ell \in \mathcal{N}_i} w_{i,\ell} y_{\ell,k} - \sum_{v_\ell \in \mathcal{N}_j} w_{j,\ell} y_{\ell,k} \le M(1 - z_{i,j}) + \epsilon, \quad \begin{cases} \forall v_i, v_j \in \mathcal{V} \\ i \neq j \\ k = 1, \ldots, n_c \end{cases} \\
& z_{i,j} + 1 \ge y_{i,k} + y_{j,k}, \quad i,j = 1, \ldots, n, \quad k = 1, \ldots, n_c
\end{aligned} \tag{4.8.10}$$

To determine the size of this ILP, we count the number of variables and constraints. The number of variables is clearly $n_{var} = nn_c + n^2$ while the number of constraints is $n_{con} = n + (n^2 - n)n_c + n^2 n_c$. In its current formulation, the ILP in Eq. (4.8.10) suffers from a few serious computational issues which can be, at least partially, addressed in the following ways.

1. The variables $z_{i,j}$ and $z_{j,i}$ must be equal and the diagonal values, $z_{i,i}$, are unnecessary. Thus, a little more than half of the variables are redundant. Instead, we reduce the set of variables $z_{i,j}$ to just $1 \le i < j \le n$, so that the number of $z$ variables is now $\frac{n^2-n}{2}$.

2. It is difficult for most ILP software to prove optimality when there are many equivalent optimal solutions. This problem arises in the formulation in Eq. (4.8.10) as the solution is a clustering $\mathcal{C}$ which is only unique up to a permutation of the labelling of the clusters. To remedy this, we assign nodes to clusters of index less than or equal to the node's index, which reduces the number of variables $y_{j,k}$, $j = 1, \ldots, n$ and $k = 1, \ldots, \min\{j, n_c\}$. Combining the new number of $z$ variables with the new number of $y$ variables, the number of variables has been reduced to,

$$\begin{aligned}
n_{var} &= \frac{n^2 - n}{2} + \frac{n_c^2 + n_c}{2} + (n - n_c)n_c \\
&= \frac{1}{2}\left(n^2 - n - n_c^2 + n_c + 2nn_c\right)
\end{aligned}$$

For large $n$ and $n_c \ll n$, these eliminations of redundancy cut the number of variables in half.

3. Additionally, the first set of constraints, CON1, has been reduced from $nn_c$ to $\frac{n_c^2+n_c}{2}+(n-n_c)n_c$. The number of the third set of constraints has been reduced from $n^2 n_c$ to $\frac{3n^2n_c-3nn_c^2+n_c^3-n_c}{6}$. The second set of constraints is unaffected yet.

4. The second set of constraints is affected by the change in variables, just in a more complex fashion. If two nodes are in the same cluster, then we must check that the their cluster strengths satisfy $||\boldsymbol{d}_j - \boldsymbol{d}_k||_\infty \leq \epsilon$. If the node of maximum index of the neighbors of nodes $v_i$ and $v_j$ is $v_\ell$ where $\ell < n_c$, then we only need to include constraints from family CON2 for $k = 1, \ldots, \ell$.

Let $\ell_j$ denote the maximum index of a neighbor of node $v_j$, i.e.,

$$\ell_j = \max_{v_\ell \in \mathcal{N}_j} \ell$$

From this analysis, there are two competing, not necessarily exclusive, goals to help make the ILP more tractable. On the one hand, we want to ensure there are as few equivalent solutions as possible, i.e., the MAEP is unique even accounting for allowed relabeling of the clusters. To ensure this, it requires finding a labeling of the nodes such that the first $n_c$ nodes are unlikely to be in any of the same clusters. If at least two nodes, $v_i$ and $v_j$ such that $i, j \leq n_c$, are in the same cluster in the MAEP, then there exists at least two clusters in the MAEP consisting of all nodes of index larger than $n_c$. The clustering where these two clusters have their labels swapped is also an MAEP and so the MAEP is not unique which makes optimality harder to prove for an ILP solver. On the other hand, we also want to relabel the nodes such that the maximum index of each node's neighbor is as small as possible to reduce the number of constraints in family CON2. The decision about which relabelling is more beneficial for an ILP solver is not clear.

The first relabelling can be performed by clustering the nodes' strengths into $n_c$ classes and choosing representatives from each cluster to have the first $n_c$ indices. The second relabelling can be performed by labelling the nodes of according to decreasing degree. A mix of the two schemes can be performed by first clustering the nodes according to strength, and then labeling each representative node from the $n_c$ clusters according to decreasing degree.

The last task to improve the performance of the ILP is to choose proper values for the big $M$ constraints. If nodes $v_i$ and $v_j$ are not in the same cluster, i.e. $z_{i,j} = 0$, then $M$ should be chosen such that this constraint is satisfied for any value of the LHS without being larger than necessary. Assuming positive edge weights, we can bound the difference

$$-s_j \leq \sum_{v_\ell \in \mathcal{N}_i} w_{i,\ell} y_{\ell,k} - \sum_{v_\ell \in \mathcal{N}_j} w_{j,\ell} y_{\ell,k} \leq s_i$$

As $\epsilon > 0$, then we can choose $M_{i,j} = s_i$. Rearranging the second constraint we can find a helpful lower bound,

$$-\sigma_j + \sigma_i z_{i,j} - \epsilon \leq \sum_{v_\ell \in \mathcal{N}_i} w_{i,\ell} y_{\ell,k} - \sum_{v_\ell \in \mathcal{N}_j} w_{j,\ell} y_{\ell,k} + \sigma_i z_{i,j} - \epsilon \leq \sigma_i$$

The lower limit of the inequality can be lower bounded by $-\sigma_j - \epsilon^{\max}$, the upper limit of $\epsilon$.

With the above discussions in mind, and using $\epsilon_{\max}$ from Eq. (4.8.4), the nodes of the graph are relabeled as described above, and the more tractable version of the

ILP is,

$$
\begin{aligned}
\min\quad & \epsilon \\
\text{s.t.}\quad & \sum_{k=1}^{\min\{j,n_c\}} y_{j,k} = 1, \quad j = 1,\dots,n \\
& -\sigma_j - \epsilon^{\max} \leq \sum_{v_\ell \in \mathcal{N}_i} w_{i,\ell} y_{\ell,k} - \sum_{v_\ell \in \mathcal{N}_j} w_{j,\ell} y_{\ell,k} \\
& \qquad\qquad + \sigma_i z_{\min\{i,j\},\max\{i,j\}} - \epsilon \leq \sigma_i \\
& \qquad\qquad \begin{cases} i,j = 1,\dots,n \\ i \neq j \\ k = 1,\dots,\min\{n_c, \max\{\ell_i,\ell_j\}\} \end{cases} \\
& -1 \leq y_{i,k} + y_{j,k} - z_{i,j} \leq 1, \quad \begin{cases} 1 \leq i < j \leq n \\ k = 1,\dots,\min\{i,n_c\} \end{cases}
\end{aligned}
\tag{4.8.11}
$$

which is in a form amenable to any ILP solver.

Nonetheless, it has been noticed that using ILP formulations to solve partitioning problems, even the formulation in Eq. (4.8.11), tends to be quite difficult, by which it is meant even small problems require long computational times. Thus, even with the symmetry breaking and constraint reduction, the problem in Eq. (4.8.11) is incapable of handling even moderate sized problems. In fact, one of the author's collaborators found that for graphs with just $n = 60$ nodes, UNM's Wheeler super computer was unable to find the MAEP within allotted times. This result does not bode well for the current formulation.

### 4.8.2 Heuristic for MAEP

With the difficulty of approaching this problem using an ILP formulation, a heuristic is proposed instead. This is inline with many other partitioning problems, such as graph coloring, where most often solutions are found using a heuristic that is,

1. very fast, and

2. capable of random initialization that leads to different sub-optimal solutions.

These two properties are important so that we may run the heuristic many times, potentially in parallel, with different initial conditions that yield different results, and then we simply select whichever run returns the best solution as measured by the cost function in Eq. (4.8.2). The heuristic developed is a modification of the polynomial time algorithm for finding the minimum equitable partition of an unweighted graph presented in [34]. To do this, some definitions are first presented.

**Definition 4.8.6** (Graph Coloring)**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph. A partition of the nodes $\mathcal{C}$ is a* valid coloring *if*

$$(v_j, v_k) \in \mathcal{E} \quad \wedge \quad v_j \in \mathcal{C}_\ell \quad \Leftrightarrow \quad v_k \notin \mathcal{C}_\ell$$

Finding the minimum graph coloring of a general graph is an NP-hard problem (but there exist substitutions for some types of graphs such as planar graphs) which means there exists no polynomial time algorithm solve this problem. Instead, many heuristics are used to solve the graph coloring problem, usually based on a greedy update.

**Definition 4.8.7** (Clique)**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph. A subgraph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ is a clique if*

$$\exists (v_j, v_k) \in \mathcal{E}, \quad \forall v_j, v_k \in \mathcal{V}'$$

*or in other words, $\mathcal{G}'$ is a* complete graph*.*

A *clique partition* of a graph is a partition such that each subgraph induced by a cluster is a clique.

**Lemma 4.8.1** (Clique Partition)**.** *A clique partition of a graph is also a graph coloring of its complement.*

*Proof.* Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph and let $\bar{\mathcal{G}} = (\mathcal{V}, \bar{\mathcal{E}})$ be its complement.

Let $\mathcal{C}$ be a clique partition of the graph $\mathcal{G}$ so that by Def. 4.8.7,

$$(v_j, v_k) \in \mathcal{E}, \quad \forall v_j, v_k \in \mathcal{C}_\ell, \quad \forall \mathcal{C}_\ell \in \mathcal{C}$$

As a graph's complement has the property $(v_j, v_k) \in \mathcal{E} \Leftrightarrow (v_j, v_k) \notin \bar{\mathcal{E}}$ then,

$$(v_j, v_k) \notin \bar{\mathcal{E}}, \quad \forall v_j, v_k \in \mathcal{C}_\ell, \quad \forall \mathcal{C}_\ell \in \mathcal{C}$$

which is the definition of a graph coloring as stated in Def. 4.8.6. □

An immediate extension of Lemma 4.8.1 is that the minimum graph coloring of a graph is also the minimum clique partition of its complement.

Given a weighted undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a partition of its nodes $\mathcal{C}$, construct a new unweighted undirected graph $\mathcal{H} = (\mathcal{V}, \mathcal{F})$ on the same set of nodes but with edges $(v_j, v_k) \in \mathcal{F}$ if and only if they satisfy Eq. (4.8.1) and $v_j, v_k \in \mathcal{C}_\ell$. If $\mathcal{H}$ consists of only disjoint cliques then $\mathcal{C}$ is an approximate equitable partition. On the other hand, if $\mathcal{H}$ does not consist of only cliques, then a refinement of the clustering $\mathcal{C}$ is made such that the new clustering $\mathcal{C}'$ is a clique partition. Using $\mathcal{C}'$, a new induced graph is created $\mathcal{H}'$ and the process is repeated until the induced graph is a clique partition.

**Algorithm 2:** Heuristic for Minimum Approximate Equitable Partitioning

```
Data: Undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and initial clustering of its nodes $\mathcal{C}^{(0)}$
      with $n_c^{(0)}$ clusters.
$k \leftarrow 0$;
$\mathcal{H}^{(0)} = (\mathcal{V}, \mathcal{F}^{(0)}) \leftarrow$ Induced Unweighted Graph $\mathcal{G}$, $\mathcal{C}^{(0)}$;
while $\mathcal{H}^{(k)}$ is not a set of disjoint cliques do
    $k \leftarrow k + 1$;
    Compute graph coloring of $\bar{\mathcal{H}}^{(k-1)}$ and call it $\mathcal{C}^{(k)}$.;
    $\mathcal{H}^{(k)} = (\mathcal{V}, \mathcal{F}^{(k)}) \leftarrow$ Induced Unweighted Graph $\mathcal{G}$, $\mathcal{C}^{(k)}$;
end
```

The initial clustering $\mathcal{C}^{(0)}$ affects the approximate equitable partition found by this heuristic so many initial choices of $\mathcal{C}^{(0)}$ are used to find a best approximate equitable

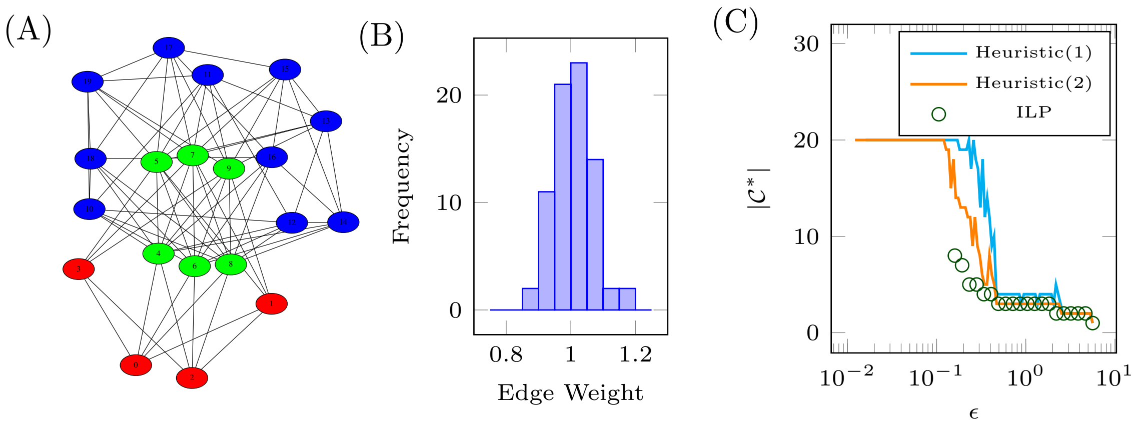


**Figure 4.13:** An example of using the minimum approximate equitable partition. (A) An undirected graph with $n = 20$ nodes. The nodes are colored according to the graph's minimum equitable partition. (B) The edge weights are drawn from a normal distribution centered around 1. (C) The comparison of three methods for computing the minimum approximate equitable partition. The first heuristic, in light blue, initializes the method with $\mathcal{C}^{(0)} = \mathcal{V}$. The second heuristic, in orange, chooses initial $\mathcal{C}^{(0)}$ with sizes $n_c^{(0)} \in \{1, 2, 3\}$. The green marks are the result returned by the ILP for values of $\epsilon$ that terminated after the allotted time provided on the Wheeler Machine at UNM's CARC.

partition. The obvious choice is to set $\mathcal{C}^{(0)} = \mathcal{V}$, that is, every node is in the same cluster. Other choice may create $\mathcal{C}^{(0)}$ by partitioning $\mathcal{V}$ into two or more clusters randomly or according to some rule. Future work on approximate symmetries will require a modification to the definition

## 4.9 Conclusion

This chapter investigates symmetries in graphs, and in particular, the problem of constructing graphs with desired orbits of its automorphism group. We describe in detail the requirements for a graph to be a *feasible quotient graph* showing that not all graphs can represent an equitable partition of some original graph. With a feasible quotient graph, we describe the procedure by which one can wire a graph such that its automorphism group is described by the original quotient graph. With the increased interest in graph symmetries as a way to induce consensus or synchronization, the ability to create systems with desirable symmetry patterns is an important contribution. Extensions to multi-layer graphs is also demonstrated.

Also included is an application of how graphs with symmetries can achieve *group* *consensus* independent of whether or not the dynamical system is stable.

# Chapter 5

# Control of Lattice Graphs

## 5.1 Introduction

This chapter explores in detail the exponential scaling demonstrated in Chapter 3 using both analytical and numerical methods for simple models. In particular, we demonstrated that the control energy of the target control problem scales exponentially, that is, holding $\mathcal{G}$ and $|\mathcal{D}|$ constant, the mean control energy scales exponentially with the cardinality of the set of target nodes $|\mathcal{T}|$ [1].

$$\langle \log E_{\mathcal{T}} \rangle_{|\mathcal{T}|=n_t} \sim \eta \frac{n_t}{n} \tag{5.1.1}$$

While the exponential scaling in Eq. (5.1.1) appeared to hold quite well, we see that the variance of the log control energy can span many orders of magnitude (see Figs. 3.3 and 3.4). This large variance makes the result in Eq. (5.1.1) somewhat less useful as its ability to predict even the order of magnitude of the control energy is dubious for any particular target node set $\mathcal{T}$. Let us consider the simplest case, that is, $|\mathcal{D}| = n_d = 1$ and $|\mathcal{T}| = n_t = 1$, which we call the *single driver single target problem.* In this case, for $\mathcal{T} = \{v_k\}$, and $\beta_k(t_f) = 1$, the control energy can be written as,

$$E_{\{v_j\}} = \frac{1}{W_{k,k}(t_f)} \tag{5.1.2}$$

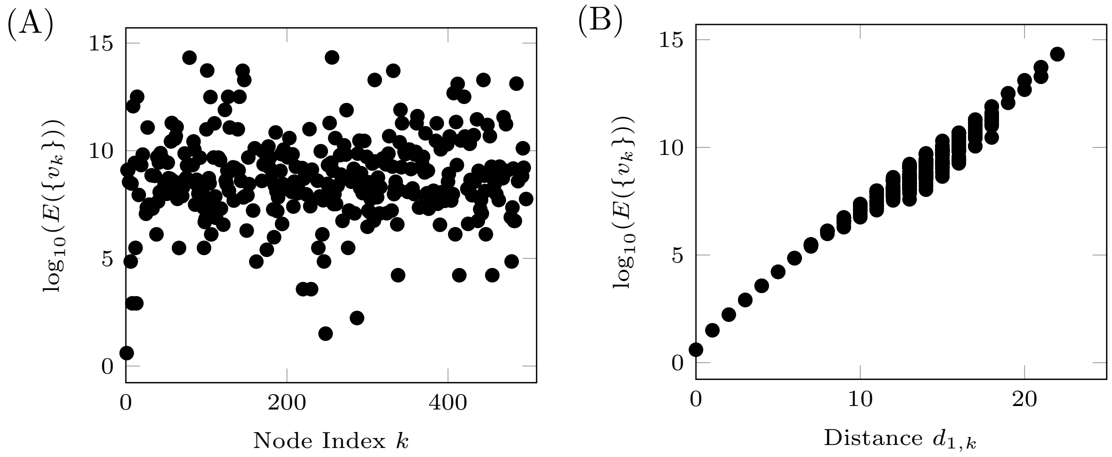


**Figure 5.1:** The control energy for the single driver single target problem for an Erdős-Rényi graph with $n = 500$ nodes and edge probability $p = 0.3\%$. The single driver is chosen to be $\mathcal{D} = \{v_1\}$ and each mark represents the control energy in Eq. (5.1.2) for $\mathcal{T} = \{v_k\}$, $k = 1, \ldots, n$. (A) The pairwise control energy is plotted as a function of the target node index $k$. (B) The control energies of the same graph as shown in (A) except, rather than plotting the control energy as a function of the node index $k$, we set the $x$-axis to be the distance between the driver node $v_1$ and the various target nodes $v_k$.

i.e., the control energy is just the inverse diagonal element of the controllability Gramian defined in Eqs. (2.3.7) and (2.3.8) corresponding to the single target node. As an example of the difficulty of predicting the control energy for the single driver single target problem, Fig. 5.1(A) plots the single driver single target control energy in Eq. (5.1.2) for an Erdős-Rényi graph (see Def. 2.2.15) with $n = 500$ nodes, edge probability $p = 0.003$, and driver node set $\mathcal{D} = \{v_1\}$. We see that the control energy spans multiple orders of magnitude, where the minimum value is clearly when $\mathcal{T} = \{v_1\}$ and the maximum is over $\log_{10}(E) > 10$.

As a first attempt to explain this large variation, we plot the control energy as a function of the length of the shortest path, or *distance* $d_{1,k}$ (see Def. 2.2.11), between the driver node $v_1$ and the various target nodes $v_k$, $k = 1, \ldots, n$ in Fig. 5.1(B). The cloud of control energies shown in Fig. 5.1(A) are now sorted nearly into a linearly increasing curve in Fig. 5.1(B), suggesting that the distance between driver node and target node must play a significant role.

This chapter investigates, in detail, the properties of the exponential scaling in Fig. 5.1(B). The derivations and more involved proofs of theorems are contained in

Appendix D. We first numerically solve the controllability Gramian for simple models which capture the important quantities of *distance* and *redundancy* in Sec. 5.2. With the numerical evidence gained in Sec. 5.2, we derive the analytic expressions for the controllability Gramian for path graphs and ring graphs in Sec. 5.3 using a variety of transforms in Appendices D.1.1 through D.4. Besides control distance, we also investigate the role of redunant paths in Sec. 5.4 which provides a corrective factor to the scaling laws derived in Sec. 5.3. The controllability Gramian of hypercubic lattices is derived in Sec. 5.5 and for general lattices is derived in Sec. 5.6. Finally, we use our analyses to propose a minimum energy driver node selection algorithm in Sec. 5.7 that, unlike other methods currently published, is not based on evaluating the controllability Gramian.

## 5.2 Control Energy and Control Distance [4]

With the above motivation, let us turn to some simple models of graphs shown in Fig. 5.2(A) and Fig. 5.2(C) which we call the *finite bidirectional path graph* and the *finite bidirectional balloon graph*, respectively. In this section, we assume all edges are undirected and have uniform edge weight equal to $s > 0$. Also, every node is assumed to have a uniform self-loop $-p$ such that $p > 0$ and the resulting adjacency matrix $A$ is *Hurwitz* so that we may compute the steady state controllability Gramian using the algebraic Lyapunov equation. The single driver single target control energy for the finite path graph is shown in Fig. 5.2(B) where each curve corresponds to a different value of the regulation parameter $p$. Each point represents the case for $\mathcal{D} = \{v_1\}$ and the target node $\mathcal{T} = \{v_{d+1}\}$. We see that as $p$ increases, the slope of the energy curve, $\eta(p)$, also increases, so that the magnitude of the regulation plays a role in the *rate of energy increase*,

$$\log E \sim \eta(p)d. \tag{5.2.1}$$

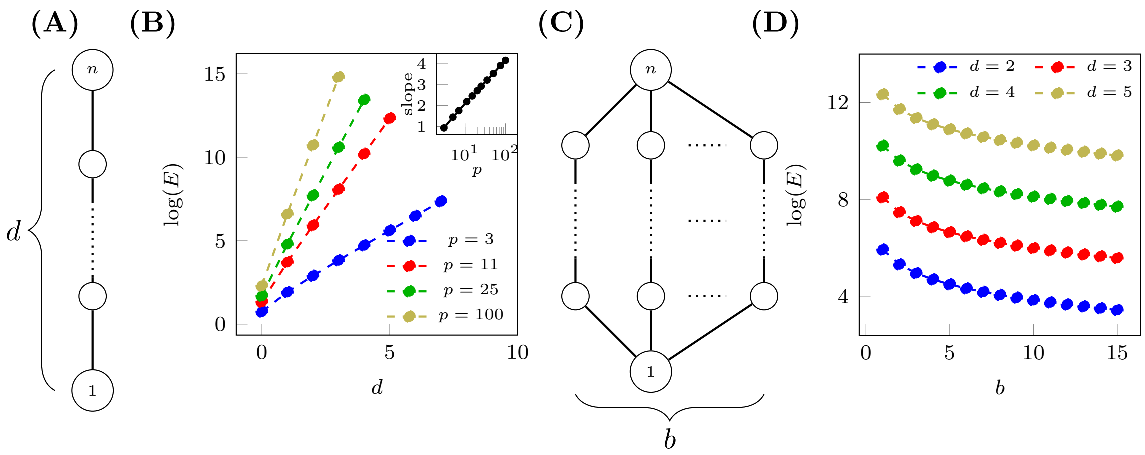


**Figure 5.2:** Control energy of simple models of graphs. (A) The finite path graph of length $n$ consists of $n$ nodes and undirected edges $(v_k, v_{k+1}) \in \mathcal{E}$ for $k = 1, \ldots, n-1$. Each edge has weight $s = 1$ and self-loop of weight $-p$ for $p > 2$ at each node. (B) The single driver single target control energy where the driver node set is $\mathcal{D} = \{v_1\}$ and the target set $\mathcal{T} = \{v_d\}$ so that the distance between the driver node and target node is $d - 1$. Each curve corresponds to different values of the self-loop weight $p$. The slopes, $\eta(p)$, of the energy curves are plotted as a function of $p$ in the inset. (C) The balloon graph is a simple model of redundant paths where the distance between the two terminal nodes is $d = n - 1$ and there are $b$ identical *branches*. (D) The control energy reduction as a function of the number of redundant branches for a selection of distances $d$ between the terminal nodes.

The inset in Fig. 5.2(B) shows the slopes of the energy curves, $\eta(p)$ in Eq. (5.2.1), as a function of $\log p$. The slope $\eta(p)$ can be approximated as,

$$\eta(p) \approx 2 \log p. \tag{5.2.2}$$

Combining Eqs. (5.2.1) and (5.2.2), the single driver single target control energy for the finite path graph can be approximated as,

$$\log E \sim 2d \log p \tag{5.2.3}$$

which captures the exponential increase in the control energy as the distance between the driver node and the target node increases linearly as seen in our motivating example in Fig. 5.1(B). The expression in Eq. (5.2.3) will be shown to be exact in the asymptotic limit of infinite path graphs in the following section.

While the finite path graph provides a relationship between control distance and control energy, control distance is not the only important parameter. The graph used to generate 5.1(B) is very *sparse* and the exponential growth of the control energy with control distance is very clear. The pure exponential increase no longer holds quite as tightly when the graph is more *dense*. In Fig. 5.3(A), we plot the single driver single target control energy as a function of the distance between the single driver node $\mathcal{D} = \{v_1\}$ and a single target node $\mathcal{T} = \{v_j\}$. This graph is also an Erdős-Rényi graph with $n = 500$ nodes as in Fig. 5.2 except that now the edge probability $p = 0.015$ so that this graph is five times more dense. Also shown in Fig. 5.3(A) is the mean log control energy for each distance $d$, $\langle \log E \rangle_d$, as red marks and the green dashed line is the control energy estimate in Eq. (5.2.3) shifted upwards by $-\log(W_{1,1})$. We see that for the first few values of $d$, our estimate is quite tight while for $d > 2$, Eq. (5.2.3) begins to overestimate the control energy significantly. We hypothesize that this is due to the *redundancy* of shortest paths in the graph.

**Definition 5.2.1** (Redundancy). *Let $\mathcal{S}_{j,k} \subseteq \mathcal{V}$ be the set of all nodes that lie along any shortest path between nodes $v_j$ and $v_k$, i.e., if $v_\ell \in \mathcal{S}_{j,k}$ then $d_{j,k} = d_{j,\ell} + d_{\ell,k}$. The redundancy of the ordered pair $v_j, v_k$ is then,*

$$r_{j,k} = \frac{|\mathcal{S}_{j,k}| - 2}{d_{j,k} - 1} \tag{5.2.4}$$

*which is defined for $d_{j,k} > 1$. If $d_{j,k} = 1$, then there can only be a single unique shortest path between nodes $v_j$ and $v_k$ as we have prohibited multi-edges. If $d_{j,k} = 0$ then $j = k$ and the concept of redundancy is not applicable.*

The redundancy for any pair of nodes is lower bounded by 1 which represents the case there is a unique shortest path between a pair of nodes. The control energy for all target nodes that satisfy $d_{1,j} = 4$ is shown in Fig. 5.3(B) as a function of $4r_{1,j}$. We see a decreasing trend of the control energy as $r_{1,j}$ increases.

A simple model of redundancy is the *balloon graph* shown in Fig. 5.2(C) which is characterized by the distance between the two terminal nodes, $v_1$ and $v_n$, denoted $d$, and the number of redundant branches, denoted $b$. The single driver single target control energy for balloon graphs with driver node $\mathcal{D} = \{v_1\}$ and target node $\mathcal{T} = \{v_n\}$

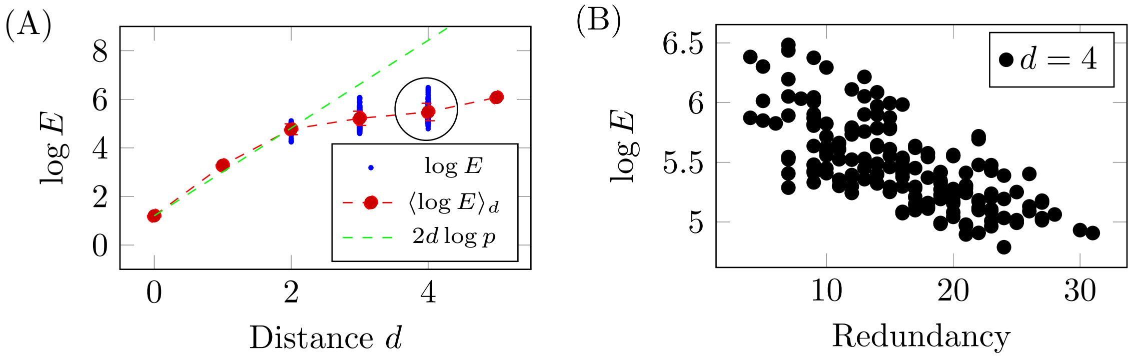


**Figure 5.3:** Control energy scaling of a denser graph. (A) The control energy of an Erdős-Rényi graph of $n = 500$ nodes and edge probability $p = 0.015$ with a single driver node $\mathcal{D} = \{v_1\}$ and a single target node. Also shown is the mean log control energy for each value of $d$ and the estimate in Eq. (5.2.3). (B) For all target nodes $v_j$ such that the distance $d_{1,j} = 4$, we compute the redundancy using Eq. (5.2.4) in Def. 5.2.1. We see there is a trend that as the number of nodes that lie along the shortest path increases, the control energy decreases.

as a function of the number of branches $b$ for some values of $d$ is shown in Fig. 5.2(D). The control energy decreases with increasing number of branches where the redundancy of a balloon graph is equal to $b$. The relationship between the density of the graph and the number of redundant paths is complex where the density of a graph is defined as,

$$\rho = 2\frac{|\mathcal{E}|}{n(n-1)} \tag{5.2.5}$$

which is the number of edges over the maximum possible number of edges in an undirected graph. If we know the distribution of shortest path lengths, $p_d$, $d = 1, 2, \ldots$, in a graph, which depends on the density of the graph among other factors, we can compute the probability of the number of nodes that lie along shortest paths between pairs of nodes a certain distance apart. Let $s = |\mathcal{S}_{j,k}|$ be defined as in Def. 5.2.1.

$$P(s|d) = \binom{n}{s} \left( \sum_{k=0}^{d} p_k p_{d-k} \right)^s \left( 1 - \sum_{k=0}^{d} p_k p_{d-k} \right)^{n-s} \tag{5.2.6}$$

From Eq. (5.2.6), in extremely dense graphs, the probability that a pair of nodes is a large distance FINISH DISCUSSION. The mean single driver single target control energy, $\langle \log E \rangle_d$, for Erdős-Rényi graphs of increasing density, proportional to $|\mathcal{E}|$, is shown in Fig. 5.4 where each graph has $n = 400$ nodes and number of edges $|\mathcal{E}|$.

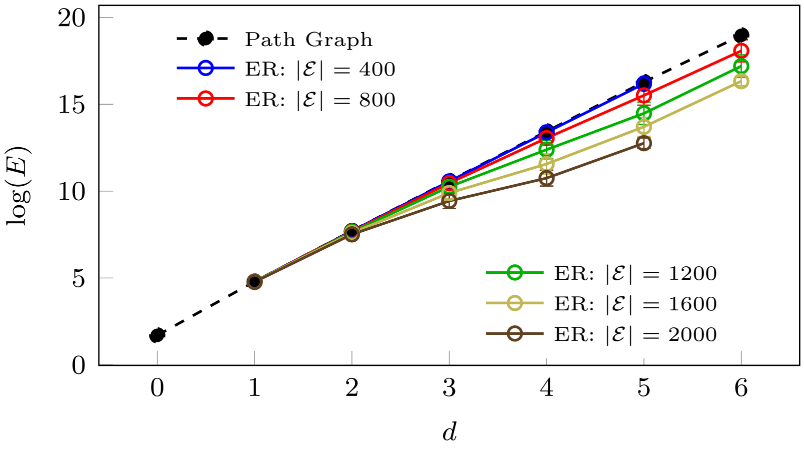


**Figure 5.4:** The deviation of the control energy from pure exponential scaling as a function of distance $d$ and density, characterized by the number of edges $|\mathcal{E}|$ (which is proportional to the density as seen in Eq. (5.2.5)). We compute the single driver single target control energy for every ordered pair of nodes $v_j, v_k$ where the first node is set to be the driver node and the second is the target node. Each graph is an Erdős-Rényi graph with $n = 400$ nodes and number of edges $|\mathcal{E}|$. The black dashed line is the control energy of a finite path graph.

As we have assumed there are no multi-edges, there can exist at most a single path between pairs of nodes of distance $(v_j, v_k) = 1$ and so the finite path graph is able to predict the control energy quite well when $d = 1$. As $d$ grows though, the finite path graph over-estimates the control energy more significantly as the graph density grows. For $|\mathcal{E}| = 400$, the average degree $< \kappa >= 2$ which is the average degree of a finite path graph in the large $n$ limit and so the blue curve is well approximated by the finite path graph control energy. As $\mathcal{E}$ grows, the deviation from pure exponential growth, shown by the black dashed line, becomes more pronounced.

The analysis so far has only numerically computed the controllability Gramian for the two simple models shown in Fig. 5.2. We determined the approximate scaling in Eq. (5.2.3) from curve fitting the numerical results in Fig. 5.2. We also demonstrated that redundancy can explain deviations from the pure exponential scaling that arises due to the distance. While providing hints at the cause of the large variance in the control energy over random sets of target nodes of equal cardinality, the underlying mechanism that causes the exponential increase remains unknown. An analytic investigation of the control energy is explored in the next section.

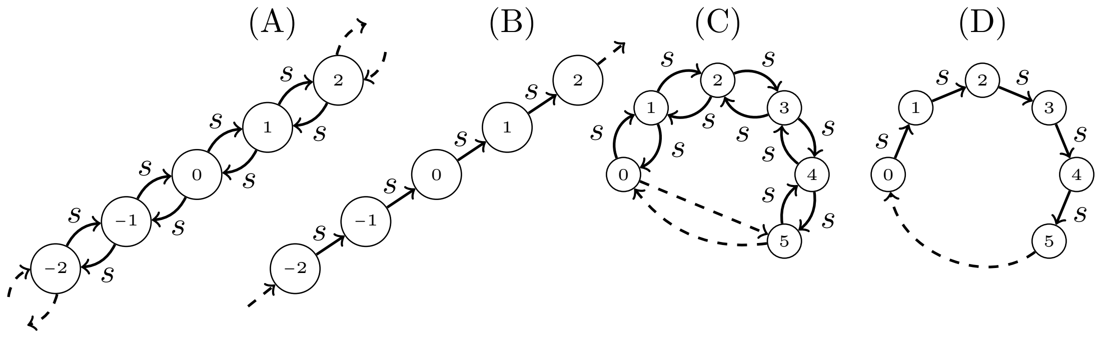


**Figure 5.5:** Diagrams of the simple graph models examined in this section. (A) The bidirectional path graph, (B) the unidirectional path graph, (C) the bidirectional ring, and (D) the unidirectional rings graph.

## 5.3 Control Energy of Paths and Rings [5]

To investigate the control energy analytically, we examine in detail a selection of simple models; bidirectional and unidirectional paths and rings. The four model graphs investigated in this section are shown in Fig. 5.5.

1. The bidirectional path graph is shown in Fig. 5.5(A) which has nodes $\mathcal{V} = \{v_k|\forall k \in \mathbb{Z}\}$ and undirected edges $\mathcal{E} = \{(v_k, v_{k+1})|\forall k \in \mathbb{Z}\}$.

2. The unidirectional path graph is shown in Fig. 5.5(B) which has nodes $\mathcal{V} = \{v_k|\forall k \in \mathbb{Z}\}$ and directed edges $\mathcal{E} = \{(v_k, v_{k+1})|\forall k \in \mathbb{Z}\}$.

3. The bidirectional ring graph is shown in Fig. 5.5(C) which has nodes $\mathcal{V} = \{v_k|k = 0, \dots, n-1\}$ and undirected edges $\mathcal{E} = \{(v_k, v_{(k+1) \bmod n}|k = 0, \dots, n-1\}$.

4. The unidirectional ring graph is shown in Fig. 5.5(D) which has nodes $\mathcal{V} = \{v_k|k = 0, \dots, n-1\}$ and directed edges $\mathcal{E} = \{(v_k, v_{(k+1) \bmod n}|k = 0, \dots, n-1\}$.

This section concerns itself with deriving closed form expressions for the elements of the controllability Gramian for each of these graphs. Also demonstrated is that, for even a moderate value of $n$, the controllability Gramians of the path graphs in Figs. 5.5(A) and 5.5(B) can approximate the controllability Gramians of the ring graphs in Figs. 5.5(C) and 5.5(D), respectively. This fact will be useful in the following sections where we relate the controllability Gramian of subgraphs of a graph to the controllability Gramian of the original graph.

### 5.3.1 The Controllability Gramian of the Bidirectional Path Graph

The bidirectional path graph is the undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where,

$$\mathcal{V} = \{v_k | \forall k \in \mathbb{Z}\} \quad \text{and} \quad \mathcal{E} = \{(v_k, v_{k+1}) | \forall k \in \mathbb{Z}\} \tag{5.3.1}$$

Each edge weight $w(v_k, v_{k+1}) = s$ and each node has self-loop of value $-p$. The differential Lyapunov equation for the infinite path graph in Eq. (5.3.1) with finite set of driver nodes $\mathcal{D} \subset \mathcal{V}$ can be written element-wise as,

$$\begin{aligned}\dot{W}_{j,k}(t) &= -2pW_{j,k}(t) + sW_{j+1,k} + sW_{j-1,k} \\ &\quad + sW_{j,k+1} + sW_{j,k-1} + \sum_{v_a \in \mathcal{D}} \delta_{j,a}\delta_{k,a}, \quad W_{j,k}(0) = 0\end{aligned} \tag{5.3.2}$$

for all $j, k \in \mathbb{Z}$. The solution of Eq. (5.3.2) is derived in Appendix D.1.1 using a Laplace transform for the temporal component and a discrete time Fourier transform (DTFT) for the spatial component resulting in the integral expression,

$$W_{j,k}(t) = \sum_{v_a \in \mathcal{D}} \int_0^t e^{-2p\tau} I_{j-a}(2s\tau) I_{k-a}(2s\tau) d\tau, \tag{5.3.3}$$

where the functions $I_n(z)$ is the modified Bessel function of the first kind (MBFFK) of integer order (see 9.6.19 in [235])

$$I_n(z) = \frac{1}{\pi} \int_0^\pi e^{z\cos\theta} \cos(n\theta) d\theta. \tag{5.3.4}$$

Some important properties of the MBFFK of integer order in Eq. (5.3.4) are that

1. $I_n(z) > 0$ for $z > 0$,

2. $I_n(0) = \delta_{n,0}$,

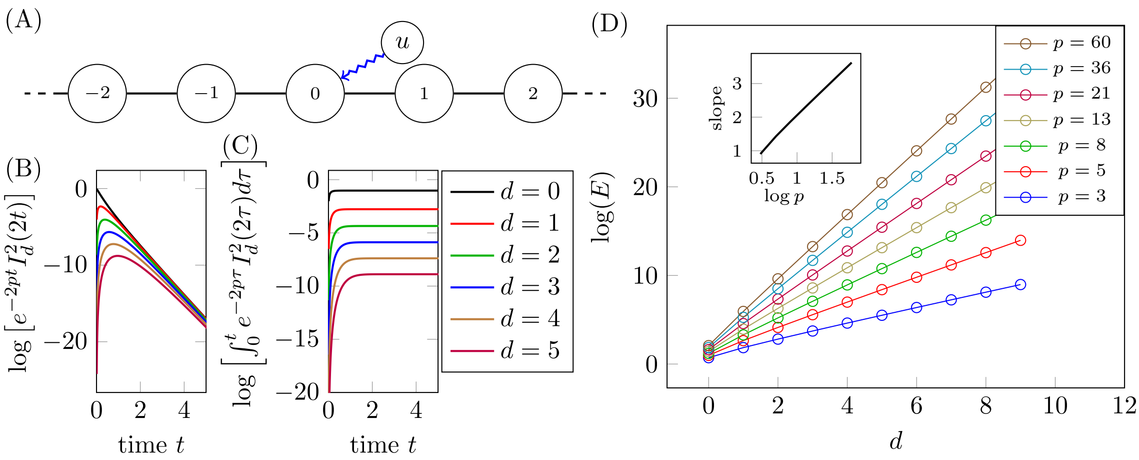


**Figure 5.6:** Control energy of the infinite path graph. A diagram of the infinite path is shown in (A) with a single driver node $\mathcal{D}=\{v_0\}$. The logarithm of the integrand in Eq. (5.3.3) is shown in (B) with edge weights $s=1$ and $p=7$. We see that as $p>2s$, the integrands each have a maximum value at some time $t\geq 0$ before exponentially decreasing towards zero with increasing time. The logarithm of the controllability Gramian in Eq. (5.3.3) is plotted in (C) for different values of $d$. (D) The control energy as a function of the distance $d$ is plotted holding $s=1$ constant and letting $p$ increase. The inset shows the slope of the curves as a function of $\log p$.

3. $\frac{d}{dz}I_n(z)>0$,

4. and $I_n(z)=I_{-n}(z)$.

Returning to the single driver single target problem, we set $\mathcal{D}=\{v_0\}$ and the target set $\mathcal{T}=\{v_d\}$ for $d\geq 0$ (as the controllability Gramian has symmetry $W_{j,k}(t)=W_{-j,-k}(t)$). The control energy of the single driver single target problem on the infinite path graph is,

$$E_d(t)=\left[\int_0^t e^{-2p\tau}I_d^2(2s\tau)d\tau\right]^{-1} \tag{5.3.5}$$

A diagram of the infinite path is shown in Fig. 5.6(A). Plots of the integrand are shown in Fig. 5.6(B) for $s=1$ and $p=7$ where we see that, as $p>2s$, the integrand has a maximum value for some time $t\geq 0$ before decreasing exponentially as $t$ grows. Plots of diagonal elements of the controllability Gramian are shown in Fig. 5.6(C) where we see they approach a constant as $t$ grows beyond the maximum value of the integrand. The control energy is shown as a function of distance $d$ and regulation $p$ where as $p$ increases, the slope of each energy curve also increases in Fig. 5.6(D). The slopes of these curves are shown as a function of $\log p$ in the inset in Fig. 5.6 where

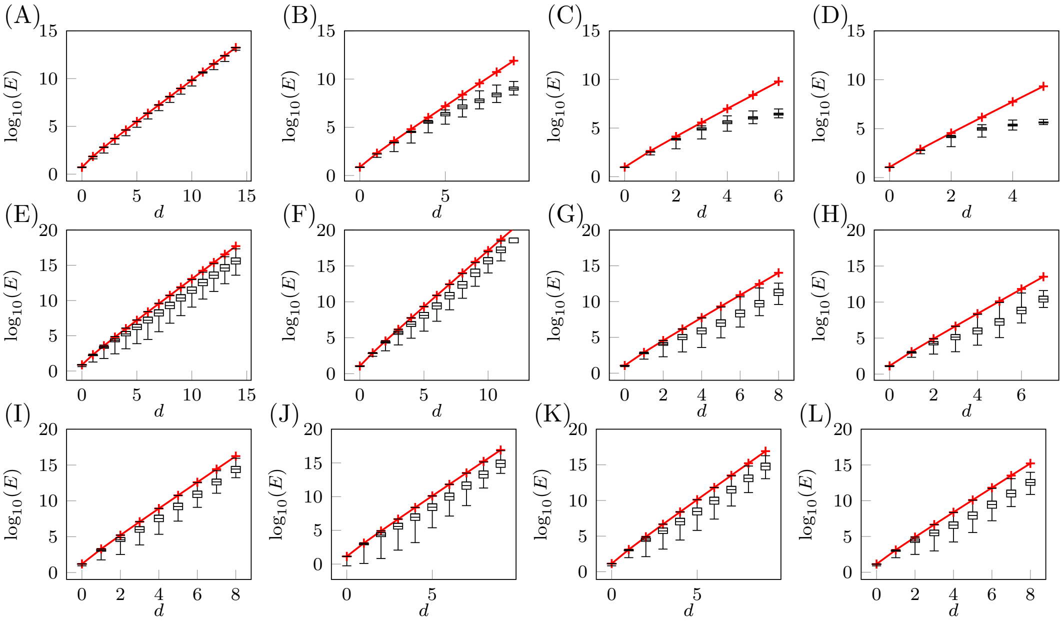


**Figure 5.7:** Control energy of the infinite path graph and single driver single target control energy in model networks. In each panel, all control energies for ordered pairs $(v_j, v_k)$ are computed for 100 realizations of graphs with $n = 100$ nodes each. The first row are k-regular graphs with (A) $k = 2$, (B) $k = 3$, (C) $k = 4$, and (D) $k = 5$. The second row are Erdős-Rényi graphs with average degrees (E) $\langle\kappa\rangle = 2$, (F) $\langle\kappa\rangle = 3$, (G) $\langle\kappa\rangle = 4$, and (H) $\langle\kappa\rangle = 5$. The third row are scale-free graphs constructed with the static model with parameters (I) $\langle\kappa\rangle = 2$ and $\gamma = 2.1$, (J) $\langle\kappa\rangle = 2.5$ and $\gamma = 2.5$, (K) $\langle\kappa\rangle = 3$ and $\gamma = 3$, and (L) $\langle\kappa\rangle = 4$ and $\gamma = 4$. Each boxplot has a center line equal to the median control energy, upper and lower edges of the box equal to the first and third quartile, and the whiskers show the maximum and minimum control energies. The red line is the corresponding control energy using Eq. (5.3.5).

again the slope is approximately $2\log p$ which agrees with the numerical results for the finite path graph depicted in Fig. 5.2(B).

To show the utility of Eq. (5.3.5), we compute the single driver single target control energy for all ordered pairs $(v_j, v_k)$ in a selection of 100 realizations of model networks of $n = 100$ nodes each shown as boxplots in Fig. 5.7. Each boxplot is centered at the median of the control energy over all pairs of nodes distance $d$ apart for a particular model network, upper and lower box edges represent the third and first quartiles, respectively, and the upper whisker and lower whiskey represent the maximum and minimum control energies, respectively. The red lines in each panel in Fig. 5.7 are found using Eq. (5.3.5) as a function of the distance $d$. The first row in Fig. 5.7(A)-(D) are k-regular graphs with $k = 2$, $k = 3$, $k = 4$, and $k = 5$. For $k = 2$, the k-regular graph consists of disjoint rings which, if large enough, behave similarly

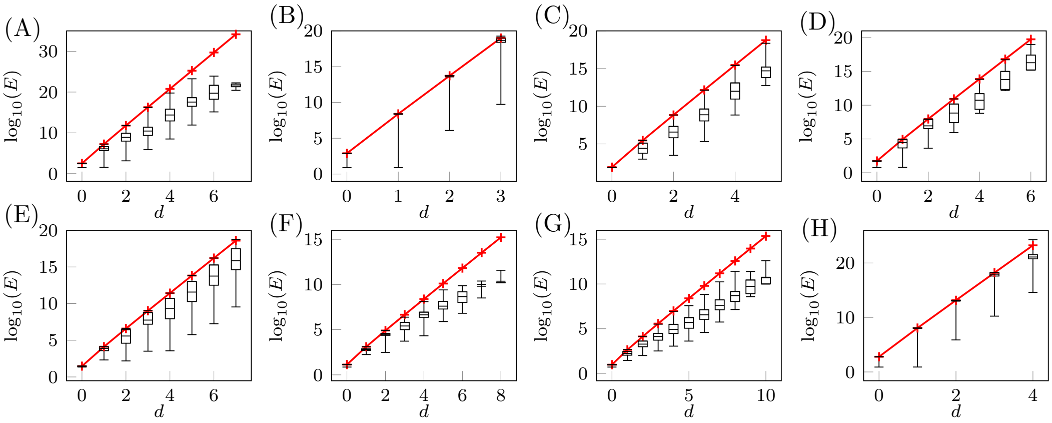


**Figure 5.8:** The single driver single target control energies for graphs derived from datasets from the literature. (A) the c. Elegans neural network [236], (B) the diseasome network [237], (C) the US air network in 1997 [180], (D) the Zachary karate club network [238], (E) a gene network made with functional associations in worms [239], (F) the visual cortex network in a mouse [240], the Klazar bibliography network [173], and (H) the collaboration network among authors who write about network science [241]. All edge weights are set to $s = 1$ and the regulation value $p$ is set so that the result adjacency matrix is Hurwitz.

to the infinite path graph and so the control energy of the infinite path well approximated the control energy of the 2-regular graphs. As $k$ increases though, the infinite path graph control energy begins to greatly over-estimate the control energy. The second row in Fig. 5.7(E)-(H) are Erdős-Rényi graphs with varying average degree $\langle \kappa \rangle$. As $\langle \kappa \rangle$ increases, the median control energy decreases, but the upper whisker remains quite close to the infinite path graph energy for those ordered pairs with no redundancy of shortest paths. The third row in Fig. 5.7(I)-(L) are scale-free graphs with various average degree $\langle \kappa \rangle$ and power-law exponent $\gamma$. We see similar results as for the Erdős-Rényi graphs with the upper whiskers nearly equal to the infinite path graph control energy. The same analysis done for the model networks in Fig. 5.7 is done for a selection of graphs drawn from the scientific literature in Fig. 5.8. The sources of each dataset is listed in the caption. We see the diseasome network in (B) and the collaboration network in (H) are very *path-like* as nearly the entire box lies on the infinite path graph energy curve. On the other hand the c. elegans neural network in (A) and the bibliography network in (G) are much more dense as the control energy with increasing distance $d$ is significantly lower than the infinite path graph.

The integral in Eq. (5.3.3) does not have a simple (expressable in terms of elementary functions) closed form solution for finite $t$. Instead, when $p > 2s$, we can compute the *steady state controllability Gramian* in the limit $t \to \infty$. One approach is to apply the final value theorem to Eq. (5.3.3) in the frequency domain (as done in Appendix D.1.2) which yields the double integral,

$$W_{j,k} = \frac{1}{2\pi^2} \sum_{v_a \in \mathcal{D}} \int_0^\pi \int_0^\pi \frac{\cos((a-j)\hat{j}) \cos(a-k)\hat{k})}{p - s\cos\hat{j} - s\cos\hat{k}} d\hat{j} d\hat{k} \tag{5.3.6}$$

Alternatively, the MBFFKs in Eq. (5.3.3) can be expanded as a power series and the integral is computed term by term in Appendix D.1.1 which yields the summation form of the controllability Gramian (setting $\mathcal{D} = \{v_0\}$),

$$W_{j,k}(t) = \frac{1}{2p} \sum_{\ell=0}^\infty \left(\frac{s}{2p}\right)^{j+k+2\ell} \binom{j+k+2\ell}{\ell} \binom{j+k+2\ell}{k+\ell} \left[1 - e^{-2pt} \sum_{a=0}^{j+k+2\ell} \frac{(2pt)^a}{a!}\right] \tag{5.3.7}$$

The steady state controllability Gramian for the bidirectional path graph is derived from Eq. (5.3.7) in terms of hypergeometric functions in Appendix D.1.2 where the result is,

$$\begin{aligned} W_{j,k} = &\frac{1}{2p} \left(\frac{s}{2p}\right)^{j+k} \binom{j+k}{j} \\ &\times {}_4F_3 \left[ \begin{matrix} \frac{1}{2}(j+k+1), & \frac{1}{2}(j+k+1), & \frac{1}{2}(j+k+2), & \frac{1}{2}(j+k+2) \\ & j+1, \quad k+1, \quad j+k+1 & \end{matrix} ; 16\left(\frac{s}{2p}\right)^2 \right] \end{aligned} \tag{5.3.8}$$

where ${}_4F_3$ is the generalized hypergeometric function (see Chapter 16 in [242])

$${}_4F_3 \left[ \begin{matrix} a_1, & a_2, & a_3, & a_4 \\ b_1, & b_2, & b_3 & \end{matrix} ; z \right] = \sum_{\ell=0}^\infty \frac{(a_1)_\ell (a_2)_\ell (a_3)_\ell (a_4)_\ell}{(b_1)_\ell (b_2)_\ell (b_3)_\ell} \frac{z^\ell}{\ell!} \tag{5.3.9}$$

and $(a)_\ell = \prod_{k=0}^{\ell-1}(a+k)$ is the rising factorial. Considering a diagonal term, $j = k$, the generalized hypergeometric term in Eq. (5.3.9) simplifies to Gauss's hypergeometric function (see Chapter 15 in [242]).

$$W_{k,k} = \frac{1}{2p}\left(\frac{s}{2p}\right)^{2k}\binom{2k}{k}{}_2F_1\left[\begin{matrix} k+\frac{1}{2}, & k+\frac{1}{2} \\ \multicolumn{2}{c}{2k+1} \end{matrix}; 16\left(\frac{s}{2p}\right)^2\right] \tag{5.3.10}$$

As there are computational libraries capable of accurately computing hypergeometric functions, Eqs. (5.3.8) and (5.3.10) are better choices for computing the steady state controllability Gramian of the bidirectional path graph rather than numerically integrating the improper integral in Eq. (5.3.3) in the $t \to \infty$ limit.

To derive the exponential increase of the control energy for the single driver single target problem in the bidirectional path graph as a function of distance between a driver node and a target node, we can derive a three term recurrence relation for the diagonal elements of the controllability Gramian of the bidirectional path graph. The integral form of the controllability Gramian in Eq. (5.3.3) is solved in the $t \to \infty$ limit in Appendix C.3 in terms of Legendre functions of the second kind (see identity 8.1.3 in [235]) which satisfy a recurrence relation (see identity 8.5.3 in [235]). After further simplification, the recurrence relation of the diagonal elements is,

$$W_{k,k} = \frac{4k-4}{2k1}\left(\frac{p^2}{2s^2}-1\right)W_{k-1,k-1} - \frac{2k-3}{2k-1}W_{k-2,k-2}, \quad k \geq 2 \tag{5.3.11}$$

The initial values for $k = 0$ and $k = 1$ can be found either by evaluating the hypergeometric function in Eq. (5.3.10) (using identities 17.3.9 and 17.3.10 in [235]) or by evaluating the integral in Eq. (5.3.3) for $t \to \infty$ (using identities 6.612.4 and 6.612.5 in [243]),

$$\begin{aligned} W_{0,0} &= \frac{1}{\pi p}K\left(\frac{2s}{p}\right) \\ W_{1,1} &= \left(\frac{p}{2\pi s^2} - \frac{1}{\pi p}\right)K\left(\frac{2s}{p}\right) - \frac{p}{2\pi s^2}E\left(\frac{2s}{p}\right) \end{aligned} \tag{5.3.12}$$

where $K(k)$ and $E(k)$ are the complete elliptic integrals of the first and second kind, respectively (see section 17.3 in [235]),

$$\begin{aligned} K(k) &= \int_0^{\pi/2} \frac{1}{\sqrt{1-k^2\sin^2\theta}} d\theta \\ E(k) &= \int_0^{\pi/2} \sqrt{1-k^2\sin^2\theta} d\theta \end{aligned} \tag{5.3.13}$$

In Appendix D.1.3, the exponential decay rate of the diagonal elements of the controllability Gramian is derived as,

$$W_{k,k} \sim \left(\frac{p^2}{2s^2} - 1 + \frac{p}{2s^2}\sqrt{p^2-4s^2}\right)^k \tag{5.3.14}$$

For $p \gg 2s$, the diagonal elements of the controllability Gramian approximately scale as $W_{k,k} \sim \left(\frac{p}{s}\right)^{2k}$ and so the control energy decays approximately as,

$$\frac{d}{dk}\log E_k \sim 2(\log p - \log s) \tag{5.3.15}$$

This behavior in the large $p$ regime was seen numerically for the finite path graph in the previous section where we hypothesized this expression in Eq. (5.2.3). Note that Eq. (5.2.3) was missing the $\log s$ term as, in the numerical studies in that section, we implicitly set $s = 1$, and so we missed the dependence on the edge weights as well.

The exponential approximation of the diagonal elements of the controllability Gramian in Eq. (5.3.14) of the infinite path graph is quite accurate even for small values of $k$ as seen in Fig. 5.9.

The controllability Gramian of the bidirectional path graph in the $t \to \infty$ limit has appeared previously as the lattice Green's function in the context of diffusion on crystalline structures. Some of the results in this subsection have been derived independently in other contexts such as Eq. (5.3.8) and Eq. (5.3.10) which were derived in a much more circuitous manner using Appell's double hypergeometric function in [244] and the recurrence relation we developed in Eq. (5.3.11) has been derived using a change of index and solving a different integral in two different ways in [245]. A

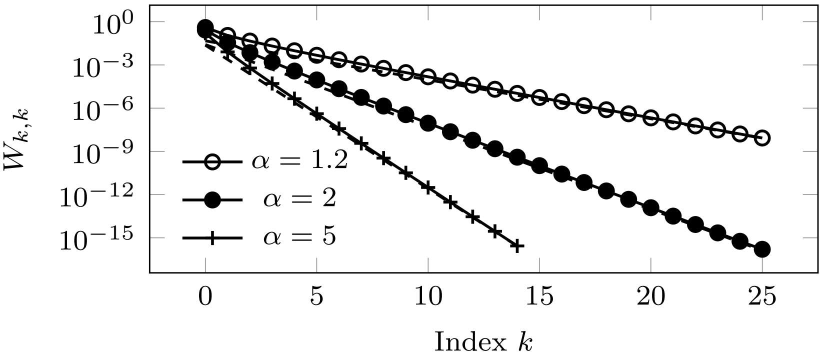


**Figure 5.9:** The exponential scaling of the diagonal elements of the controllability Gramian of the infinite path graph. For some values of $\alpha = \frac{p^2}{2s^2} - 1$, the marks represent the control energy found using the recurrence in Eq. (5.3.11) while the solid lines are the scaling in Eq. (5.3.14).

more numerically stable version of the recurrence relation from [245] was developed in [246] that does not require any numerical integrations (as required when computing the complete elliptic integrals, Eq. (5.3.13), in Eq. (5.3.12)).

### 5.3.2 The Controllability Gramian of the Unidirectional Path Graph

The next simple model, the unidirectional path graph shown in Fig. 5.5(B), will be used in the following sections for both derivations concerning redundancy and for strict upper bounding the single driver single target control energy in complex networks. The unidirectional path graph is the directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

$$\mathcal{V} = \{v_k | k = 0, 1, \ldots\} \quad \text{and} \quad \mathcal{E} = \{(v_{k-1}, v_k) | k = 1, 2, \ldots\} \tag{5.3.16}$$

As noted previously, the contribution of each driver node is independent, and so without loss of generality, we derive the controllability Gramian for $\mathcal{D} = \{v_0\}$ as we can shift the indices of the result to place the driver node at other nodes. The controllability Gramian of the unidirectional path graph in Eq. (5.3.16) satisfies the differential equation,

$$\begin{aligned}
\dot{W}_{j,k}(t) &= -2pW_{j,k}(t) + sW_{j-1,k}(t) + sW_{j,k-1}(t), \quad W_{j,k}(0) = 0, \quad j,k \geq 1 \\
\dot{W}_{j,0}(t) &= -2pW_{j,0}(t) + sW_{j-1,0}(t), \quad W_{j,0}(0) = 0, \quad j \geq 1 \\
\dot{W}_{0,k}(t) &= -2pW_{0,k}(t) + sW_{0,k-1}(t), \quad W_{0,k}(t) = 0, \quad k \geq 1 \\
\dot{W}_{0,0}(t) &= -2pW_{0,0}(t) + 1, \quad W_{0,0}(0) = 0
\end{aligned} \tag{5.3.17}$$

for all pairs $j,k \in \mathbb{Z}^+$ with the formality that $W_{j,k}(t) = 0$ if $j < 0$ or $k < 0$. The system of differential equations in Eq. (5.3.17) is solved using a Laplace transform and a two term generating function in Appendix D.2 yielding the time-varying controllability Gramian elements,

$$W_{j,k}(t) = \frac{1}{2p}\left(\frac{s}{2p}\right)^{j+k}\binom{j+k}{k}\left[1 - e^{-2pt}\sum_{\ell=0}^{j+k}\frac{(2pt)^\ell}{\ell!}\right] \tag{5.3.18}$$

Note that the expression for the controllability Gramian of the unidirectional path in Eq. (5.3.18) is equal to the first term of the expression for the controllability Gramian of the bidirectional path in Eq. (5.3.7). Taking the $t \to \infty$ when $p > s$ yields the steady state controllability Gramian of the unidirectional path.

$$W_{j,k} = \frac{1}{2p}\left(\frac{s}{2p}\right)^{j+k}\binom{j+k}{k} \tag{5.3.19}$$

The expression for the steady state controllability Gramian in Eq. (5.3.8) of the bidirectional path graph and the unidirectional path graph in Eq. (5.3.19) are equal except for a factor that depends on $j$ and $k$, the ${}_4F_3$ hypergeometric function. By examining the behavior of ${}_4F_3$, the affect of undirected graphs versus directed graphs can be quantified.

The scaling of the single driver single target control energy can be derived using Eq. (5.3.19) as well. In Appendix D.2, the scaling of the control energy in the asymptotic limit is shown to be,

$$W_{k,k} \sim \left(\frac{s}{p}\right)^{2k}, \tag{5.3.20}$$

which is equal to the control energy scaling of the bidirectional path graph in the large $p$ limit as discussed below Eq. (5.3.14). Letting $k$ be a continuous value, we can compute the scaling of the control energy by differentiating the negative logarithm of Eq. (5.3.20),

$$\frac{d}{dk}\log E_k \sim 2(\log p - \log s). \tag{5.3.21}$$

The exponential growth of the control energy for the unidirectional path graph in Eq. (5.3.21) in the asymptotic limit is equal to the exponential growth of the control energy for the bidirectional path graph in Eq. (5.3.15) which only holds for $p \gg 2s$, so that, in the large $p$ regime, bidirectional path graphs behave more like unidirectional path graphs.

### 5.3.3 The Controllability Gramian of the Bidirectional Ring Graph

The bidirectional ring is defined on a set of $n$ nodes, $v_0, v_1, \ldots, v_{n-1}$, with undirected edges $(v_k, v_{k \bmod n})$, $k = 0, \ldots, n-1$. The controllability Gramian for the bidirectional ring is derived in Appendix D.3 in a similar manner as the bidirectional path graph. The time-varying controllability Gramian for the bidirectional ring is,

$$\begin{aligned} W_{j,k}(t) = \sum_{v_a \in \mathcal{D}} \int_0^t e^{-2p\tau} &\left[\frac{1}{n}\sum_{\hat{j}=0}^{n-1} e^{2s\tau\cos\frac{2\pi}{n}\hat{j}} \cos\frac{2\pi}{n}(a-j)\hat{j}\right] \\ \times &\left[\frac{1}{n}\sum_{\hat{k}=0}^{n-1} e^{2s\tau\cos\frac{2\pi}{n}\hat{k}} \cos\frac{2\pi}{n}(a-k)\hat{k}\right] d\tau \end{aligned} \tag{5.3.22}$$

and the steady state controllability Gramian for the bidirectional ring is,

$$W_{j,k}\frac{1}{2n^2}\sum_{v_a\in\mathcal{D}}\sum_{\hat{j}=0}^{n-1}\sum_{\hat{k}=0}^{n-1}\frac{\cos\left(\frac{2\pi}{n}(a-j)\hat{j}\right)\cos\left(\frac{2\pi}{n}(a-k)\hat{k}\right)}{2p-2s\cos\frac{2\pi}{n}\hat{j}-2s\cos\frac{2\pi}{n}\hat{k}} \tag{5.3.23}$$

Noting that in the $n\to\infty$ the bidirectional ring becomes the bidirectional path, Eq. (5.3.22) approaches the expression in Eq. (5.3.3) for finite time while in the $t\to\infty$ Eq. (5.3.23) approaches the expression in Eq. (5.3.6). For $n$ large enough, the scaling of the control energy derived for the bidirectional path graph appear to hold for nodes $v_k$, $k=0,\dots,n/2$ (as because of the symmetries in the ring, the control energy for $v_k$ and $v_{n-k}$ are equal).

### 5.3.4 The Controllability Gramian of the Unidirectional Ring Graph

The unidirectional ring graph, shown in Fig. 5.5(D), is defined on a set of $n$ nodes, like the bidirectional ring, except that the edges, $(v_k, v_{(k+1)\bmod n})$, are directed. Rather than using a discrete Fourier transform or a generating function as was used to derive the results in Subsection 5.3.2 and 5.3.3, instead we prove that the controllability Gramian of the unidirectional ring is a rational function in terms of the parameter $\rho=\frac{s}{2p}$.

**Theorem 5.3.1** (Controllability Gramian of Unidirectional Ring)**.** *The controllability Gramian of the unidirectional ring, which is goverened by the cyclic recurrence relation,*

$$W_{j,k}=\rho W_{j-1\bmod n,k}+\rho W_{j,k-1\bmod n}+\alpha\delta_{j,0}\delta_{k,0},\quad j,k=0,\dots,n-1$$

*has elements equal to,*

$$W_{j,k}=\alpha\frac{\sum_{\ell=0,1,\dots}a_{j,k}^{(\ell)}\rho^{\ell n+j+k}}{\sum_{\ell=0,1,\dots}b_\ell\rho^{\ell n}} \tag{5.3.24}$$

*where all of the coefficients $a_\ell^{(j,k)}$ and $b_\ell$ are integers.*

| $n$ | $b_1$ | $b_2$ | $b_3$ | $b_4$ | $b_5$ | $b_6$ |
|---|---|---|---|---|---|---|
| 3 | $-7$ | $-8$ | - | - | - | - |
| 4 | $-12$ | $-64$ | - | - | - | - |
| 5 | $-21$ | $-353$ | 32 | - | - | - |
| 6 | $-38$ | $-1691$ | 1728 | - | - | - |
| 7 | $-71$ | $-7585$ | 36991 | 128 | - | - |
| 8 | $-136$ | $-32880$ | 552704 | 65536 | - | - |
| 9 | $-265$ | $-139823$ | 6826204 | 6965249 | $-512$ | - |
| 10 | $-522$ | $-587797$ | 75135226 | 392963125 | $-3200000$ | - |
| 11 | $-1035$ | $-2452907$ | 767040961 | 15735018638 | $-2061791231$ | $-2048$ |
| 12 | $-2060$ | $-10184978$ | 7435036420 | 508938068953 | $-516171815360$ | $-191102976$ |

**Table 5.1:** Coefficients of the Denominator Polynomial. The coefficients for the first few rings of length $n \geq 3$ are collected which were computed numerically. The first column is equal to $b_1 = -2^n - n - 1$.

The proof is contained in Appendix D.4 where the coefficients $a_{j,k}^{(\ell)}$ in Eq. (5.3.24) are shown to be

$$\begin{aligned} a_{j,k}^{(\ell)} &= \sum_{p=0}^{\ell} c_{j,k}^{(p)} b_{\ell-p} \\ c_{j,k}^{(p)} &= \sum_{q=0}^{p} \binom{np+j+k}{nq+j} \end{aligned} \tag{5.3.25}$$

To find the values of $b_\ell$ in Eq. (5.3.24), the system of $m = \lceil \frac{n}{2} \rceil$ linear diophantine equations are constructed, denoted $M\boldsymbol{b} = \boldsymbol{h}$ where the matrix $M \in \mathbb{Z}^{m \times m}$ and vector $\boldsymbol{h} \in \mathbb{Z}^m$ have elements composed from $c_{j,k}^{(p)}$ in Eq. (5.3.25),

$$M_{j,k} = c_{j,j}^{(m-k)}, \quad \boldsymbol{h}_j = -c_{j,j}^{(m)} \tag{5.3.26}$$

The benefit of this approach is that the coefficients $b_\ell$ and $a_{j,k}^{(\ell)}$ need only be computed once using Eq. (5.3.26) as they are independent of the edge weights $s$ and loops $p$. The coefficients $b_\ell$ for the first few values of $n$ are collected in Table 5.1. Note that the first term in the expression for the controllability Gramian for the unidirectional ring in Eq. (5.3.24) is equal to the controllability Gramian of the unidirectional path. As $\rho < \frac{1}{4}$, the contribution of each term $\ell \geq 1$ becomes negligible and the unidirectional ring behaves much like the unidirectional path.

Having analytically derived the scaling of the single driver single target control energy in paths and rings, which explains the exponential scaling, we turn now to redundancy which was briefly explored in the second half of Section 5.2 which will add a corrective term to the results.

## 5.4 Redundancy

To analyze the role that redundant shortest paths play in the control energy, we exploit the fact that each redundant path in the balloon graphs discussed briefly before contain is symmetric. We now consider graphs with symmetries as defined in Def. 2.2.22 and further elaborated on in Chapter 4.

**Theorem 5.4.1** (Controllability Gramian of Symmetric Graphs)**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with symmetries represented by permutation matrices $P$. By definition, $PAP^T = A$ and $PP^T = I$. We use the driver node refined automorphism group, $Aut_{\mathcal{D}}(\mathcal{G}$, which is a subgroup of the automorphism group $Aut(\mathcal{G})$, which consists of only those permutations,*

$$P \in Aut_{\mathcal{D}}(\mathcal{G}) \quad \textit{if} \quad PAP^T = A \quad \textit{and} \quad PB = B$$

*Then, the controllability Gramian is invariant under the permutation,*

$$PW(t)P^T = W(t)$$

*Proof.* The proof is straightforward as we only need to pre- and post-multiply the differential Lyapunov equation by $P$ and $P^T$, respectively.

$$\begin{aligned} P\dot{W}(t)P^T &= PAP^T PW(t)P^T + PW(t)P^T PA^T P^T + PBB^T P^T, \quad PW(0)P^T = O_n \\ &= APW(t)P^T + PW(t)P^T A^T + BB^T, \quad PW(0)P^T = O_n \end{aligned}$$

As $W(t)$ and $PW(t)P^T$ satisfy the same differential equation with the same initial condition, $W(t) = PW(t)P^T$. □

Let $Q$ be the quotient graph found from the equitable partition induced by the

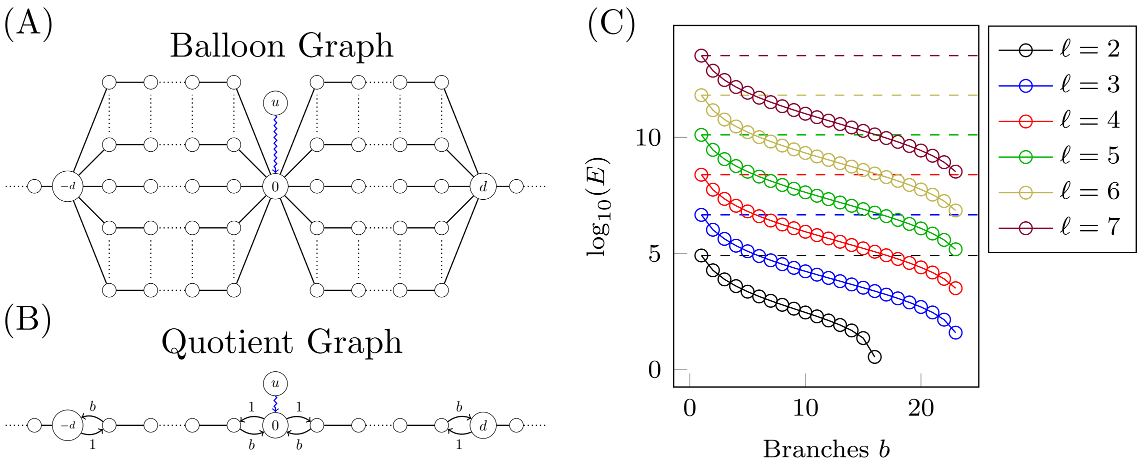


**Figure 5.10:** The control energy of the balloon graph found using its quotient graph. The balloon graph of distance $d$ and branches $b$ is shown in (A) with its associated quotient graph in (B). The control energy for different values of $d$ is shown in (C) as a function of $b$ with the dashed lines equal to the infinite path graph energy in Eq. (5.3.5).

driver node refined automorphism group $\mathrm{Aut}_{\mathcal{D}}(\mathcal{G})$. The controllability Gramian of the quotient network, $V(t)$, can be written,

$$\dot{V}(t) = QV(t) + V(t)Q^T + E^{\dagger}BB^TE^{\dagger^T}$$

where $E$ is the orbit indicator matrix such that $E_{k,\bar{k}} = 1$ and is equal to zero otherwise. Then, if nodes $v_j \in \mathcal{O}_{\bar{j}}$ and $v_k \in \mathcal{O}_{\bar{k}}$ , are in the stated orbits of $\mathrm{Aut}_{\mathcal{D}}(\mathcal{G})$, it can be proven [5] that

$$V_{\bar{j},\bar{k}}(t) = W_{j,k}(t) \tag{5.4.1}$$

We return to the balloon graph now noting that the nodes along each branch in a balloon graph are symmetric. Applying Thm. 5.4.1 and Eq. (5.4.1) the quotient graph of the balloon graph, shown in Fig. 5.10(B), is an infinite path graph except that four edges now have edge weights equal to $b$, namely the edges $(v_{-1}, v_0)$, $(v_1, v_0)$, $(v_{-d+1}, v_{-d})$, and $(v_{d-1}, v_d)$. This relatively small change reduces the control energy by orders of magnitude as shown in Fig. 5.10(C) as seen previously for the finite balloon graph in Fig. 5.2(D). Using Def. 5.2.1, the redundancy in a balloon graph

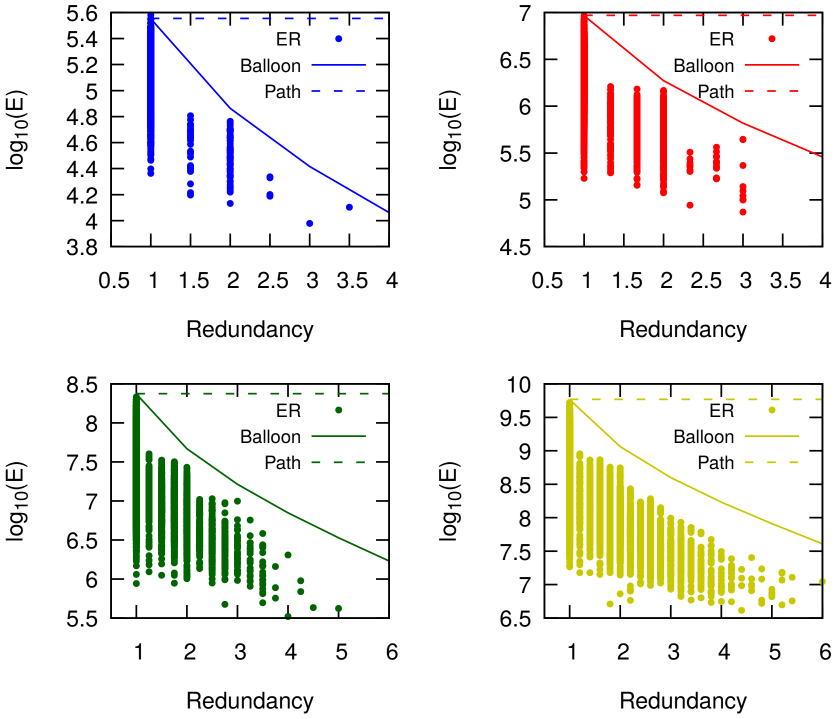


**Figure 5.11:** The control energy reduction with redundancy as prediction by the balloon graph. The points are the single driver single target control energy for an Erdős-Rényi graph with $n = 300$ nodes and edge probability $p = 0.02$ as a function of each pair of nodes' redundancy. The dashed lines are the control energies for the bidirectional path graph for each distance $d$ (which is invariant with $b$) and the solid lines are the control energies for the balloon graph for each distance $d$. The blue panel is $d = 3$, the red panel is distance $d = 4$, the green panel is distance $d = 5$, and the yellow panel is distance $d = 6$.

between nodes $v_0$ and $v_d$ can be found as,

$$r_{0,d} = \frac{(2 + (d-1)b) - 2}{d-1} = b$$

that is, the number of brances in the balloon graph. For an example graph in Fig. 5.11, we see that the dashed lines, which represents the control energy for the infinite path graph, over-estimates the control energy considerably as the branch connectivity increases. On the other hand, we see that the solid lines, which represent the control energy for the balloon graph, better approximates the control energy as the redundancy increases.

Despite constructing a model to represent the control energy as a function of distance $d$ and redundancy $r$, we still see that the control energy holding $d$ and $b$ constant in a complex network may still span an order of magnitude. Other graph properties must play additional roles in the control energy, possibly measures of graph centrality of both the driver node and the target node.

The above control energies for the bidirectional symmetric balloon graph were computed numerically. To make the problem analytically tractable, we construct the unidirectional balloon graph. Using the analysis above, we construct the *unidirectional balloon graph.* The quotient graph of this graph is a unidirectional path graph with uniform edge weights as described before except the last edge $(v_{d-1}, v_d)$ which has weight $b$. After applying the result of Thm. 5.4.1 we compute the element $W_{d,d}$ of the steady state controllability Gramian. The derivation is presented in Appendix D.5 and results in the expression,

$$W_{d,d} = \frac{b^2}{2p}\left(\frac{s}{2p}\right)^{2d}\binom{2d}{d} \tag{5.4.2}$$

where we see the corrective factor $b^2$. The logarithm of the control energy (found by inverting Eq. (5.4.2)) makes clear the independent contributions of the distance and the redundancy.

$$\log E(d,b) = -\log\frac{b^2}{2p} - 2d\log\frac{s}{2p} - \log\binom{2d}{d} \tag{5.4.3}$$

Even though the previous balloon graphs we examined in Fig. 5.2(C) and in Fig. 5.10(A) were *undirected*, the rate of decay of the control energy as a function of $b$ appeared to be *independent* of $d$ as well, as shown in Fig. 5.2(D) and Fig. 5.10(C). Computing the derivatives of the logarithm of the control energy of the unidirectional balloon graph in Eq. (5.4.3),

$$\begin{aligned}\frac{d}{d(d)}\log E_d &= 2\log\frac{2p}{s} + 2\sum_{k=d+1}^{2d}\frac{1}{k} \\ &= 2(\log p - \log s)\end{aligned} \tag{5.4.4}$$

and

$$\frac{d}{d(b)}\log E_d = -\frac{2}{b} \tag{5.4.5}$$

Details of these results can be found in Appendix D.5. These results appear to agree with the numericaly results in Fig. 5.10(C) where each curve is spaced evenly in the log scale (as Eq. (5.4.4) is independent of distance $d$) while looking along each curve, for smaller $b$, appears to decay more slowly as $b$ grows, as suggested by Eq. (5.4.5). For values of $b$ that make $sb$ just slightly less than $p$, other effects begin to play a larger role as the numerical results in Fig. 5.10 are for an undirected balloon graph. Equations (5.4.4) and (5.4.5) hold exactly for unidirectional balloon graph.

## 5.5 Control Energy of Hypercubic Lattices [6]

A $d$-dimensional hypercubic lattice, denoted $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, has nodes,

$$\mathcal{V} = \{v_{\boldsymbol{j}} | \forall \boldsymbol{j} \in \mathbb{Z}^d\}$$

and edges,

$$\mathcal{E} = \left\{(v_{\boldsymbol{j}\pm\boldsymbol{e}_\ell}, v_{\boldsymbol{j}}) | \forall \boldsymbol{j} \in \mathbb{Z}^d \wedge \ell = 1, \dots, d\right\} \tag{5.5.1}$$

where $\boldsymbol{e}_\ell$ is the $\ell$'th unit vector. For brevity, using Eq. (5.5.1), let the set of relative neighbors be defined as,

$$\mathcal{N} = \{\pm \boldsymbol{e}_\ell | \ell = 1, \dots d\} \tag{5.5.2}$$

Performing the same series of steps as in the previous sections where we take the Laplace transform of the differential Lyapunov equation using the new set of neighbors in Eq. (5.5.2), and then the $2d$-dimensional DTFT,

$$\hat{\tilde{W}}_{\hat{\boldsymbol{j}},\hat{\boldsymbol{k}}}(\epsilon) = \frac{1}{\epsilon} \sum_{v_{\boldsymbol{a}} \in \mathcal{D}} \frac{e^{-i\boldsymbol{a}\hat{\boldsymbol{j}}} e^{-i\boldsymbol{a}\hat{\boldsymbol{k}}}}{\epsilon + 2p - \phi(\hat{j}) - \phi(\hat{k})}$$

where $\phi(\hat{\boldsymbol{j}})$ is the lattice function defined for this lattice to be,

$$\phi(\hat{\boldsymbol{j}}) = \sum_{\ell=1}^{d} 2s \cos \hat{\boldsymbol{j}}_\ell \tag{5.5.3}$$

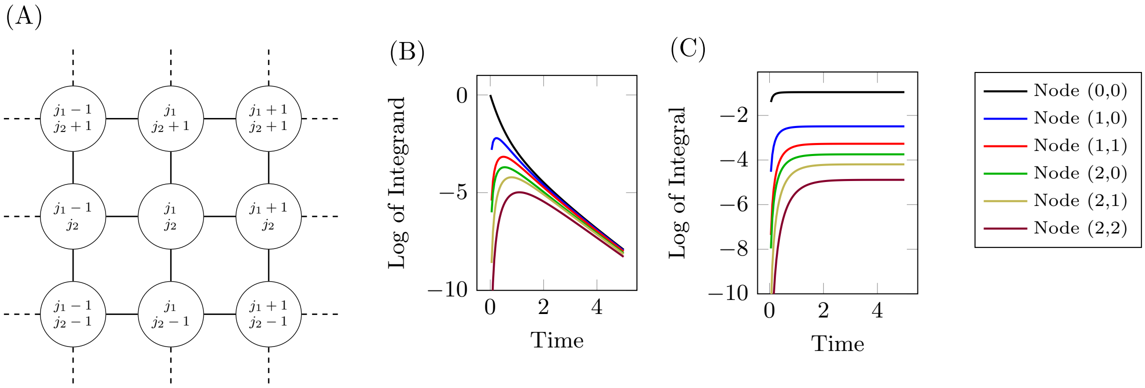


**Figure 5.12:** Example of the square lattice's Gramian. (A) A unit block of the square lattice with nodes labeled with 2-tuple indices. (B) The log of the integrand of Eq. (5.5.5) for the case $\mathcal{D} = \{v_{(0,0)}\}$. (C) The log of the diagonal elements of the controllability Gramian.

and the vector products $\boldsymbol{a}\hat{\boldsymbol{j}} = \sum_{\ell=1}^{d} \boldsymbol{a}_\ell \hat{\boldsymbol{j}}_\ell$. Now taking the inverse DTFT applying Eq. (5.5.3)

$$\tilde{W}_{j,k}(\epsilon) = \frac{1}{\epsilon}\frac{1}{(2\pi)^{2d}} \sum_{v_{\boldsymbol{a}}\in\mathcal{D}} \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} \frac{e^{i(\boldsymbol{j}-\boldsymbol{a})\hat{\boldsymbol{j}}} e^{i(\boldsymbol{k}-\boldsymbol{a})\hat{\boldsymbol{k}}}}{\epsilon + 2p - \phi(\hat{\boldsymbol{j}}) - \phi(\hat{\boldsymbol{k}})} d\hat{\boldsymbol{j}}\, d\hat{\boldsymbol{k}} \tag{5.5.4}$$

where each integral is taken over the $[-\pi,\pi]^d$ $d$-dimensional hypercube. Taking the inverse Laplace transform of Eq. (5.5.4) we can find the time-varying controllabiltiy Gramian of the infinite hypercubic lattice.

$$\begin{aligned} W_{\boldsymbol{j},\boldsymbol{k}}(t) &= \sum_{v_{\boldsymbol{a}}\in\mathcal{D}} \int_0^t e^{-2p\tau} \prod_{\ell=1}^{d} \left[\frac{1}{2\pi}\int_{-\pi}^{\pi} e^{2s\tau\cos\hat{\boldsymbol{j}}_\ell} e^{i(\boldsymbol{j}_\ell-\boldsymbol{a}_\ell)\hat{\boldsymbol{j}}_\ell} d\hat{\boldsymbol{j}}_\ell\right] \\ &\times \left[\frac{1}{2\pi}\int_{-\pi}^{\pi} e^{2s\tau\cos\hat{\boldsymbol{k}}_\ell} e^{i(\boldsymbol{k}_\ell-\boldsymbol{a}_\ell)\hat{\boldsymbol{k}}_\ell} d\hat{\boldsymbol{k}}_\ell\right] d\tau \\ &= \sum_{v_{\boldsymbol{a}}\in\mathcal{D}} \int_0^t e^{-2p\tau} \prod_{\ell=1}^{d} I_{\boldsymbol{j}_\ell-\boldsymbol{a}_\ell}(2s\tau) I_{\boldsymbol{k}-\boldsymbol{a}}(2s\tau) d\tau \end{aligned} \tag{5.5.5}$$

An example of the square lattice, $d = 2$, is shown in Fig. 5.12(A). For the square lattice, the controllability Gramian is now indexed by four values, $W_{(j_1,j_2),(k_1,k_2)}$. We plot the log of the integrand of some *diagonal* elements of the controllability Gramian in Eq. (5.5.5), by which we mean $j_1 = k_1$ and $j_2 = k_2$ in Fig. 5.12(B), for the node indices listed at the far right of Fig. 5.12. Similar to the case when $d = 1$ which we plotted in Fig. 5.6(B), we see that for $p > 2ds$, the integrand reaches a maximum before exponentially decreasing. As the integrand has a single maximum before exponentially decreasing, controllability Gramian approaches a constant as $t$ grows in Fig. 5.12(C). This again is similar to the behavior for $d = 1$ shown in Fig. 5.6(C).

## 5.6 Control Energy of General Lattices [14]

For a general lattice of $d$ dimensions, we introduce the framework of a *lattice operator* which describes the connectivity. We keep the concept of a lattice index vector $\boldsymbol{j} \in \mathbb{Z}^d$ used in the previous section. Two examples of lattices are shown in Fig. 5.13. The familiar bidirectional chain is shown in Fig. 5.13(A) to show how it fits in the general framework discussed here and a two dimensional directed lattice is shown in Fig. 5.13(B). The differential Lyapunov equation can be written using the notation,

$$\dot{W}_{\boldsymbol{j},\boldsymbol{k}}(t) = -2pW_{\boldsymbol{j},\boldsymbol{k}}(t) + \Delta_j W_{\boldsymbol{j},\boldsymbol{k}}(t) + \Delta_k W_{\boldsymbol{j},\boldsymbol{k}}(t) + \sum_{v_{\boldsymbol{a}} \in \mathcal{D}} \delta_{\boldsymbol{j},\boldsymbol{a}} \delta_{\boldsymbol{k},\boldsymbol{a}} \tag{5.6.1}$$

where the lattice operator is defined as,

$$\Delta_j W_{j,k}(t) = \sum_{\boldsymbol{\ell} \in \mathcal{I}} s_{\boldsymbol{\ell}} W_{\boldsymbol{j}+\boldsymbol{\ell},\boldsymbol{k}}(t) \tag{5.6.2}$$

The lattice operator for the bidirectional chain in Fig. 5.13(A) is $\Delta_j W_{j,k}(t) = sW_{j+1,k}(t) + sW_{j-1,k}(t)$ while for the directed two dimensional lattice in Fig. 5.13(B) it is,

$$\Delta_{\boldsymbol{j}} W_{\boldsymbol{j},\boldsymbol{k}}(t) = s_1 W_{j_1+1,j_2,k_1,k_2}(t) + s_2 W_{j_1,j_2+1,k_1,k_2} + s_3 W_{j_1-1,j_2-1,k_1,k_2}(t)$$

In Appendix D.7, Eq. (5.6.1) is solved for finite time,

$$W_{\boldsymbol{j},\boldsymbol{k}}(t) = \sum_{v_{\boldsymbol{a}} \in \mathcal{D}} \int_0^t e^{-2p\tau} A_{\boldsymbol{j}-\boldsymbol{a}}(\tau) A_{\boldsymbol{k}-\boldsymbol{a}}(\tau) d\tau \tag{5.6.3}$$

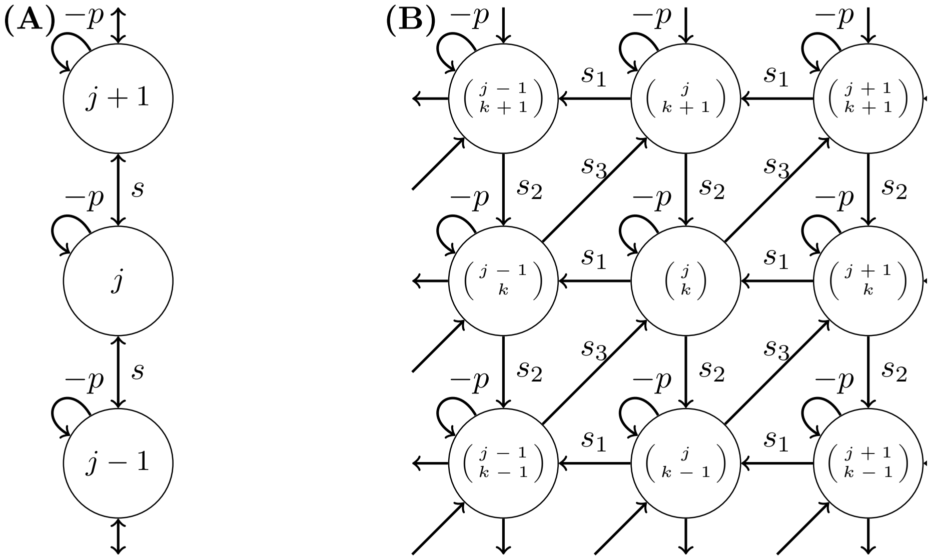


**Figure 5.13:** Two lattice graphs are shown. (A) The familiar bidirectional chain graph with uniform loop weight $-p$ and uniform edge weight $s$. (B) A two dimensional lattice with relative neighbors $(1, 0)$, $(0, 1)$, and $(-1, -1)$. There is a uniform loop weights of $-p$ and each of the three relative neighbors, $s_1$, $s_2$, and $s_3$.

where the functions,

$$A_{\boldsymbol{j}}(t) = \frac{1}{(2\pi)^d} \int_{-\pi}^{\pi} e^{\sigma(\boldsymbol{x})t} \cos(\omega(\boldsymbol{x})t + \boldsymbol{j}\boldsymbol{x}) d\boldsymbol{x} \tag{5.6.4}$$

are integrals over the hyper cube of dimension $d$ and side-length $2\pi$. The functions $\sigma(\boldsymbol{x})$ and $\omega(\boldsymbol{x})$ are real and imaginary components of the discrete time Fourier transform of the lattice operator,

$$\mathcal{F}(\Delta_{\boldsymbol{j}}) = \sum_{\boldsymbol{\ell} \in \mathcal{I}} s_{\boldsymbol{\ell}} e^{i\boldsymbol{\ell}\boldsymbol{x}} = \sigma(\boldsymbol{x}) + i\omega(\boldsymbol{x})$$

For the bidirectional chain in Fig. 5.13(A),

$$\mathcal{F}(\Delta_j) = se^{ix} + se^{-ix} = 2\cos x$$

while for the directed two dimensional lattice in Fig. 5.13(B) it is,

$$\begin{aligned}\mathcal{F}(\Delta_{\boldsymbol{j}}) =& s_1 e^{ix_1} + s_2 e^{ix_2} + s_3 e^{-ix_1} e^{-ix_2} \\ &= (s_1 \cos x_1 + s_2 \cos x_2 + s_3 \cos(x_1 + x_2)) + i\,(s_1 \sin x_1 + s_2 \sin x_2 - s_3 \sin(x_1 + x_2))\end{aligned}$$

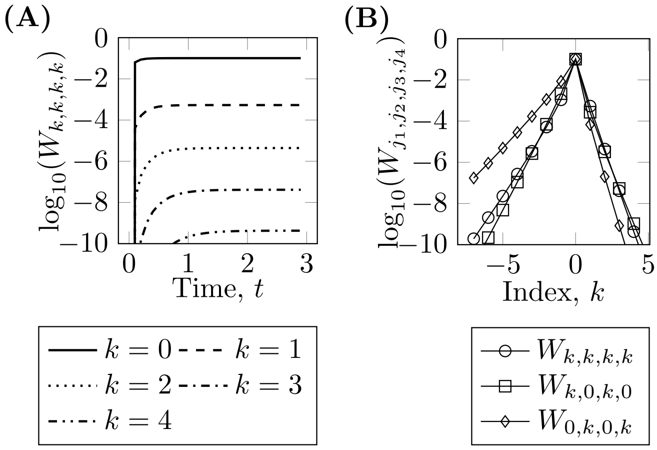


**Figure 5.14:** Numerical results for the single driver problem applied to the lattice drawn in Fig. 5.13(B). The single driver is set to the origin node, $\mathcal{D} = \{v_{0,0}\}$. For $p = 5$, $s_1 = 1$, $s_2 = 2$, and $s_3 = 0.5$, in (A) the elements $W_{k,k,k,k}(t)$ are shown to converge to a constant value that decays exponentially as $k$ grows linearly. (B) For three different axes of the controllability Gramian, the rate of growth is shown to be different.

In Fig. 5.14, elements of the controllability Gramian for the lattice shown in Fig. 5.13(B) are computed numerically. The regulation parameter $p = 5$ and the edge weights $s_1 = 1$, $s_2 = 2$, and $s_3 = 0.5$. For a two dimensional lattice, it is useful to think of the controllability Gramian as having four indices, two from node $v_{\boldsymbol{j}}$ and two from node $v_{\boldsymbol{k}}$. For the single driver single target problem, where $\mathcal{D} = \{v_{0,0}\}$ and $\mathcal{T} = \{v_{k,k}\}$, a node diagonal from the driver node, the time traces of the controllability Gramian are shown in Fig. 5.14(A) which are seen to converge to constant values as expected. For other axes, $\mathcal{T} = \{v_{k,0}\}$ (moving horizontally from the dirver node) and $\mathcal{T} = \{v_{0,k}\}$ (moving vertically from the driver node), the steady state Gramian values are plotted in Fig. 5.14(B). We see that as $|k|$ grows, the values of the controllability Gramian decay exponentially, but at different rates.

As an example of the applicability of this general framework, we apply it to compute the Gramian using Eqs. (5.6.3) and (5.6.4) for the intermediate path lattice that provides the connection between the bidirectional path lattice and the unidirecitonal path lattice. Define a lattice with neighbors $\mathcal{I} = \{-1, 1\}$ like the bidirectional chain but with edge weights $s_{-1} = s$ and $s_1 = \bar{s}$. The lattice operator in Eq. (5.6.2) then acts as follows,

$$\Delta_j W_{j,k}(t) = sW_{j-1,k}(t) + \bar{s}W_{j+1,k}(t)$$

Evaluating the Fourier transform of the lattice operator,

$$\mathcal{F}(\Delta_{\boldsymbol{j}}) = se^{-ix} + \bar{s}e^{ix} = s\cos x + \bar{s}\cos x + i(-s\sin x + \bar{s}\sin x)$$

Evaluating the function $A_j(t)$,

$$\begin{aligned} A_j(t) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{(s+\bar{s})\cos x}\cos((\bar{s}-s)\sin x + jx)dx \\ &= \frac{1}{2\pi}\int_{0}^{2\pi} e^{(s+\bar{s})\cos x}\cos((s-\bar{s}) - jx)dx \end{aligned}$$

Using identity 3.937.2 in [243] with the parameters,

$$\begin{aligned} p &= (s+\bar{s})t \quad & a &= 0 \\ q &= 0 & b &= (s-\bar{s})t \end{aligned}$$

and the derived values,

$$\begin{aligned} A &= 4s\bar{s}t^2, \quad & C &= 4s\bar{s}t^2 \\ B &= 0, & D &= 0 \\ (b-p)^2 &= 4\bar{s}^2t^2, & (a+q)^2 &= 0 \end{aligned}$$

the solution of the integral becomes,

$$\begin{aligned} A_j(t) &= \frac{1}{2\pi}\left[\frac{2\pi}{[4\bar{s}^2t^2]^{j/2}}(4s\bar{s}t^2)^{j/2}I_j(2\sqrt{s\bar{s}}t)\right] \\ &= \left(\frac{s}{\bar{s}}\right)^{j/2} I_j(2\sqrt{s\bar{s}}t) \end{aligned}$$

For the bidirectional path, $\bar{s} = s$, and we recover the original result. For the unidirectional path, we take the limit $\bar{s} \to 0$, and use the power series form of the modified Bessel function of the first kind of integer order,

$$\lim_{\bar{s}\to 0} A_j(t) = \left(\frac{s}{\bar{s}}\right)^{j/2}\left(\left(\sqrt{s\bar{s}}t\right)^j \sum_{\ell=0}^{\infty}\frac{1}{(j+\ell)!\ell!}\left(\sqrt{s\bar{s}}t\right)^{2j}\right) = \frac{(st)^j}{j!}$$

Evaluating the controllability Gramian elements yields,

$$W_{j,k}(t) = \int_0^t e^{-2p\tau} \frac{(s\tau)^{j+k}}{j!k!} d\tau$$

This integral can be solved using identity 3.351.1 from [243],

$$\begin{aligned} W_{j,k}(t) &= \frac{s^{j+k}}{j!k!} \left[ \frac{(j+k)!}{(2p)^{j+k+1}} - e^{-2pt} \sum_{\ell=0}^{j+k} \frac{(j+k)!}{\ell!} \frac{(t)^\ell}{(2p)^{(j+k)-\ell+1}} \right] \\ &= \frac{1}{2p} \left( \frac{s}{2p} \right)^{j+k} \frac{(j+k)!}{j!k!} \left[ 1 - e^{-2pt} \sum_{\ell=0}^{\infty} \frac{(2pt)^\ell}{\ell!} \right] \end{aligned}$$

which agrees with the previous result derived using generating functions. For intermediate values of $\bar{s}$, we have the Gramian elements,

$$W_{j,k}(t) = \int_0^t e^{-2p\tau} I_j(2\sqrt{s\bar{s}}\tau) I_k(2\sqrt{s\bar{s}}\tau) d\tau$$

which provides the connection between the bidirectional chain and the unidirectional chain.

## 5.7 Driver Node Selection Procedure

As briefly discussed in Chapter 2, a problem of great importance is the design problem to construct the set of driver nodes $\mathcal{D}$ to minimize a control metric. The control metrics of interest are related to the control energy repeated here for clarity,

$$E(B) = \boldsymbol{\beta}^T \bar{W}^{-1}(B) \boldsymbol{\beta} = \sum_{k=1}^{n_t} \frac{\theta_k^2}{\mu_k} \tag{5.7.1}$$

where $\mu_k$, $\boldsymbol{\xi}_k$ is an eigenvalue, eigenvector pair of the output controllability Gramian $\bar{W}$ and $\theta_k = \boldsymbol{\beta}^T \boldsymbol{\xi}_k$. The dependence on $B$ is made explicit as this is the quantity which we will develop a method to design in this section. The control energy expression in Eq. (5.7.1) is dependent on the particular choice of control maneuver $\boldsymbol{\beta}$ which may not be desirable if one wants to select sets of driver nodes for an arbitrary control maneuver. Instead, noting that $E(B)$ is a weighted sum of the inverse eigenvalues, we can instead choose to minimize,

$$-\log\det\bar{W} = -\log\prod_{k=1}^{n_t}\mu_k = \sum_{k=1}^{n_t}\log\frac{1}{\mu_k} \tag{5.7.2}$$

The $\log\det\bar{W}$ is related to the volume of the ellipsoid reachable by one unit of energy discussed in Sec. 3.5.

$$\log(V) = \log\left(\frac{\pi^{n_t/2}}{\Gamma(n_t/2+1)}\right) + \frac{1}{n_t}\sum_{k=1}^{n_t}\log\mu_k \tag{5.7.3}$$

One may equivalently either maximize the volume of the ellipsoid in Eq. (5.7.3) or minimize the $\log\det\bar{W}$ in Eq. (5.7.2). Additional choices include the trace of the controllability Gramian or the trace of the inverse of the controllability Gramian. The negative trace of the controllability Gramian,

$$-\operatorname{Tr}(\bar{W}) = -\sum_{k=1}^{n_t}\mu_k \tag{5.7.4}$$

is inversely related to the control energy and is related to the $H_2$ norm of the system [138]. The trace of the inverse of the controllability Gramian,

$$\operatorname{Tr}(\bar{W}^{-1}) = \sum_{k=1}^{n_t}\frac{1}{\mu_k} \tag{5.7.5}$$

is the most closely related to the original expression for the control energy in Eq. (5.7.1).

The problem we are interested in solving in this section is the driver node selection that was described in Sec. 2.3.7. The general optimization problem is,

$$\begin{aligned} \min \quad & f(\bar{W}) \\ \text{s.t.} \quad & |\mathcal{D}| = n_d \end{aligned} \tag{5.7.6}$$

where the cost function should be either the control energy in Eq. (5.7.1) or any of the surrogate cost functions in Eqs. (5.7.2), (5.7.4), or (5.7.5). This optimization problem is *combinatorial*, that is, the set of candidate solutions is *finite*. The number of possible solutions is $\binom{n}{n_d}$ if any node could be chosen to be a driver node which eliminates a brute force search as an option to solve Eq. (5.7.6) for all but the smallest (either $n$ very small or $n_d/(n - n_d)$ very small) problems.

The optimization problem in Eq. (5.7.6) for any of the cost functions discussed so far have been proven to be NP-hard [96, 148, 149] so heuristics and approximation algorithms must be used instead. The constraint in the optimization problem in Eq. (5.7.6) places this problem in the Driver Node Framework discussed at the beginning of Sec. 2.3.2.

There are two main methods to solve Eq. (5.7.6) currently published:

1. The first family of methods [151] lets the entries of $B$ be continuous and uses a gradient descent method to solve the optimization problem,

$$\begin{aligned} \min \quad & E(B) = \mathrm{Trace}(W^{-1}(B)X_f) \\ \text{s.t.} \quad & N(B) = \mathrm{Trace}(B^T B) - n_d - \epsilon = 0 \end{aligned} \tag{5.7.7}$$

   where $n_d$ is the final desired number of driver nodes, $\epsilon > 0$ is a positive constant, and $X_f = e^{A^T t_f} e^{A t_f}$. After a local optimum of Eq. (5.7.7) is found, the entries of $B$ are rounded to construct the set of driver nodes $\mathcal{D}$. The success of this method, as a heuristic, requires many random initializations, and the hope that the solution to the integer version of the problem in which we are really interested lies close enough to one of the local minima of the continuous relaxation.

2. The second family of methods uses the fact that some control metrics are submodular set functions [138] over the set of nodes in a graph. Specifically, the negative versions of Eqs. (5.7.2), (5.7.4), and (5.7.5) were all shown to be submodular set functions over the set of nodes $\mathcal{V}$ [138]. Some results on submodular

set function optimization is discussed in Def. 2.3.8 and the discussion that follows. The benefit of showing a cost function is submodular is that a greedy algorithm has an approximation guarantee. Specifically, let $\mathcal{D}_j = \{v_j\}$ be the single driver node sets and $\bar{W}(\mathcal{D}_j) = \bar{W}_j$ be the controllability Gramian when only node $v_j$ is selected to be a driver node. The controllability Gramian can then be written as a sum over the contributions of each driver node.

$$\bar{W}(\mathcal{D}) = \sum_{v_k \in \mathcal{D}} \bar{W}_k \tag{5.7.8}$$

The greedy approximation algorithm starts with $\mathcal{D}^{(0)} = \emptyset$ and for $k = 1, \ldots, n_d$ augments $\mathcal{D}^{(k)}$ by solving,

$$\mathcal{D}^{(k)} = \mathcal{D}^{(k-1)} \bigcup \underset{v_j \notin \mathcal{D}^{(k-1)}}{\operatorname{argmin}} \left\{ f\left( \bar{W}(\mathcal{D}^{(k-1)} \cup \{v_j\}) \right) \right\}, \tag{5.7.9}$$

exploiting the additivity in Eq. (5.7.8). Note that the procedure in Eq. (5.7.9) requires storing $n$ symmetric matrices $\bar{W}_j$, $j = 1, \ldots, n$, and computing the function $f(\cdot)$ $nn_d - \frac{n_d(n_d-1)}{2}$ times which may become expensive.

Before moving on, one additional note is made about the greedy approximation algorithm. For $n_d$ small relative to $n_t$, computing the cost functions may be impossible using double precision floats, which will occur for the first few steps of the greedy algorithm. A work around suggested in [138] is to instead use rank$(\bar{W})$ as the cost function for the first few steps of the greedy approximation algorithm. At step $k$ when rank$(\bar{W}(\mathcal{D}^{(k)}) = n_t$, the original cost function is used for the remaining $n_d - k$ steps. An alternative work around is suggested in [149] where the cost function used is $f(\bar{W} + \epsilon I_{n_t})$ where the value of $\epsilon$ is chosen carefully in order to preserve the approximation guarantee of the greedy algorithm.

Both methods discussed attempt to perform the minimization directly on the controllability Gramian without explicitly exploiting the graph structure. Instead, we suggest a structure based heuristic using results developed in the previous sections in this chapter covering the relationships between graph distance and redudancy and control energy.

### 5.7.1 Theorem on the Gramian of Subgraphs

Before presenting the theorem we introduce some useful notation to order matrices. Let $M$ be an array (vector or matrix), then $M > 0$ ($M \geq 0$) denotes that all entries of $M$ are strictly positive (non-negative). Similarly, if $M_1$ and $M_2$ are two arrays of equal dimension, then $M_1 > M_2$ ($M_1 \geq M_2$) denotes that $(M_1 - M_2) > 0$ ($(M_1 - M_2) \geq 0$). The algebraic Lyapunov equation, $AW + WA^T = -BB^T$, is a linear equation and so it can be written in a more familiar form,

$$\hat{A}V = -\hat{B}$$

where $\hat{A} = (A \otimes I_{n^2}) + (I_{n^2} \otimes A)$ is the state matrix, $V \in \mathbb{R}^{n^2}$ is the stacked columns of the controllability Gramian into a vector and $\hat{B}$ is the stacked columns of the matrix $BB^T$. As $-A$ is an M-matrix, so is $-\hat{A}$. Similarly, we assume that $B \geq 0$ and so $-\hat{B} \leq 0$.

A useful property of M-matrices is that if $M$ is an M-matrix, then $M^{-1} \geq 0$. Related to this is that $M^{-1}\boldsymbol{x} \geq 0$ if $\boldsymbol{x} \geq 0$.

**Theorem 5.7.1** (Lower Bounding the Controllability Gramian). *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with adjacency matrix $A$. Similarly, let $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$ be a subgraph of $\mathcal{G}$ so that $\mathcal{E}' \subseteq \mathcal{E}$ with adjacency matrix $A'$. For both graphs, let the input matrices be $B$ and $B'$, respectively, such that $\Delta B = B - B' \geq 0$. By definition of the edge weights, $\Delta A = A - A' \geq 0$. We assume that $-A$ is an M-matrix and thus so is $-A'$. Then, if $W$ is the controllability Gramian for pair $(A, B)$ and $W'$ is the controllability Gramian for pair $(A', B')$, the difference $\Delta W = W - W' \geq 0$.*

*Proof.* The proof is inductive. For the base case, let $\mathcal{G} = (\mathcal{V}, \{(v_1, v_2)\})$ be a graph with $|\mathcal{V}| = n$ nodes and a single edge, and let the subgraph then be $\mathcal{G} = (\mathcal{V}, \emptyset)$. Without loss of generality, we assume that each node has self-loop $-p < 0$ and the single edge has weight $s$. The controllability Gramian of the disconnected graph is,

$$W' = \frac{1}{2p}B'B'^T \geq 0$$

which is a diagonal matrix. The controllability Gramian of the graph with a single edge, $(v_1, v_2)$, denoted $W$, is equal to $W'$, except for the elements $W_{1,2} = W_{2,1}$ which has value,

$$W_{1,2} = \begin{cases} 0 & \text{if } v_1, v_2 \notin \mathcal{D} \\ \frac{s}{4p^2} & \text{if } v_1 \in \mathcal{D}, \quad v_2 \notin \mathcal{D} \\ \frac{1}{2p} & \text{if } v_1 \notin \mathcal{D}, \quad v_2 \in \mathcal{D} \\ \frac{1}{2p} + \frac{1}{2p} + \frac{s}{4p^2} & \text{if } v_1, v_2 \in \mathcal{D} \end{cases} \tag{5.7.10}$$

and the element $W_{2,2}$ which is equal to,

$$W_{2,2} = 2sW_{1,2} + \sum_{v_a \in \mathcal{D}} \delta_{a,2}$$

To show $W_{1,2} - W'_{1,2} \geq 0$ and $W_{2,2} - W'_{2,2} \geq 0$, one simply goes through each case in Eq. (5.7.10).

For the inductive hypothesis, we assume that $W' \geq 0$, the controllability Gramian of some subgraph of an original graph $\mathcal{G}$. The controllability Gramians of the two graphs are,

$$\hat{A}V = -\hat{B} \quad \text{and} \quad \hat{A}'V' = -\hat{B}'$$

Subtracting these two equations from each other,

$$\begin{aligned} \hat{A}V - \hat{A}'V' &= \hat{A}V + \Delta\hat{A}V' - \hat{A}V' \\ &= \hat{A}\Delta V + \Delta\hat{A}V' = -\hat{B} + \hat{B}' = -\Delta\hat{B} \end{aligned}$$

Solving for the change in the Gramian,

$$\Delta V = (-\hat{A})^{-1}(\Delta\hat{B} + \Delta\hat{A}V')$$

where we note that $\Delta\hat{A} \geq 0$ and $\Delta\hat{B} \geq 0$ by construction and $V' \geq 0$ by the inductive hypothesis so that $(\Delta\hat{B} + \Delta\hat{A}V') \geq 0$. Also, as $-\hat{A}$ is an M-matrix, then $(-\hat{A})^{-1} \geq 0$ and so the whole right hand side of is a non-negative vector, thus implies that $\Delta V \geq 0$. Thus concludes the proof that $\Delta W = W - W' \geq 0$. □

The usefulness of Thm. 5.7.1 is that for a graph $\mathcal{G}$ with a set of driver nodes $\mathcal{D}$ and a set of target nodes $\mathcal{T}$, we can construct a subgraph $\mathcal{G}'$ that consists of paths from a driver node to a target node, or rings from a driver node to a target node and back to a driver node and we know that each element $W_{j,k} \geq W'_{j,k}$. In particular, if we trying to minimize $-\mathrm{Tr}(\bar{W}(\mathcal{D}))$, then we know $-\mathrm{Tr}(\bar{W}'(\mathcal{D}))$ is less than or equal to the true value.

In the remainder of this section, the graph's structure is exploited to choose driver nodes independent of evaluating the controllability Gramian directly. The method developed is compared to the greedy heuristic described in [138] with the rank$(\bar{W})$ initialization.

### 5.7.2 Facility Location and k-Median Problems

As demonstrated throughout this chapter, the control energy required to drive a particular target node to a particular state is exponentially less if the driver node used is closer to the target node. This result suggests that if one must select the driver node to perform such a task, the driver node available for input that is closest (with the maximum amount of redundancy) to the target node should be chosen. On the other hand, if there is a significant number of target nodes which we must control using more than one driver node then the selection process becomes far less clear. There is competition between target nodes as choosing a driver node close to one target node may be very far from another target node.

In operations research, facility location problems are concerned with choosing a subset of nodes that are minimal in terms of some metric with respect to another set of nodes. Originally, this type of problem was designed to choose where a company should open distribution centers from which a set of retailers or population centers would be supplied. A number of variants hace since emerged that add additional constraints or costs depending on the application. For the purpose here, we use the $k$-median variation which adds the constraint that we must assign exactly $k$ nodes to be driver nodes (facilities). Without loss of generality, we assume that the first $n_t$ nodes are target nodes, i.e., $\mathcal{T} = \{v_j | j = 1, \dots, n_t\}$. The first set of variables are $X_k \in \{0, 1\}$, $k = 1, \dots, n$, which is 1 if node $v_k \in \mathcal{D}$ and 0 otherwise. The second set of variables are $Y_{j,k} \in \{0, 1\}$, $j = 1, \dots, n_t$ and $k = 1, \dots, n$ which is 1 if target $v_j$ is assigned to driver node $v_k$. There is a cost associated with assigning target node $v_j$ to node $v_k$, denoted $c_{j,k}$, which is chosen from the scaling behavior derived previously in the chapter.

$$\begin{aligned} \min \quad & \sum_{j=1}^{n_t} \sum_{k=1}^{n} c_{j,k} Y_{j,k} X_k \\ \text{s.t.} \quad & \sum_{k=1}^{N} X_k = n_d \\ & \sum_{k=1}^{n} Y_{j,k} = \ell, \quad j = 1, \dots, n_t \\ & Y_{j,k} \leq X_k, \quad j = 1, \dots, n_t, \quad k = 1, \dots, n \\ & Y_{j,k}, X_k \in \{0, 1\}, \quad j = 1, \dots, n_t, \quad k = 1, \dots, n \end{aligned} \tag{5.7.11}$$

The first constraint ensures we choose exactly $n_d$ driver nodes. The second constraint ensures each target node is assigned to $\ell \leq n_d$ driver nodes. In the usual facility location problem, $\ell = 1$. The final constraint ensures that target node $v_j$ is assigned to driver node $v_k$ only if node $v_k$ has been selected to be a driver node.

The proper choice of the costs $c_{j,k}$ is imperative for the solution of Eq. (5.7.11) to return useful results. Let $FLP(\mathcal{D})$ be the cost associated with a particular set of driver nodes evaluated with Eq. (5.7.11). The optimal choice of $c_{j,k}$ is one which satisfies the following inequalities for any pair of driver node sets $\mathcal{D}$ and $\mathcal{D}'$ such that $|\mathcal{D}| = |\mathcal{D}'| = n_d$.

$$FLP(\mathcal{D}) < FLP(\mathcal{D}') \quad \text{iff} \quad f(\bar{W}(\mathcal{D})) < f(\bar{W}(\mathcal{D}')) \tag{5.7.12}$$

where $f(\cdot)$ is one of the submodular set functions described above. If we can find such a cost that makes Eq. (5.7.12) true, then the solution of Eq. (5.7.11) returns the optimal solution to the original problem in Eq. (5.7.6).

To investigate Eq. (5.7.12) numerically, we must be able to compute $FLP(\mathcal{D})$ for any set of driver nodes. If $\ell = 1$ as in the usual formulation of Facility Location Problems, this can be done easily by evaluating,

$$FLP(\mathcal{D}) = \sum_{j=1}^{n_t} \min_{v_k \in \mathcal{D}} c_{j,k}$$

where extensions to $\ell > 0$ simply replace the summand with the $\ell$ smallest elements.

To determine the behavior of $FLP(\mathcal{D})$ relative to the cost $f(\bar{W}(\mathcal{D}))$, we first choose random sets of driver nodes and evaluate both costs for these sets. We have found that random sampling from the space of driver node sets tends to result in points tightly clustered in a narrow region in the $FLP(\mathcal{D})$ vs $-\log\det\bar{W}$ space. To sample more uniformly from this space, we use the fact that computing $FLP(\mathcal{D})$ is cheap, we use a hill climbing algorithm to find sets of driver nodes $\mathcal{D}$ such that $FLP(\mathcal{D}) \in [c-\epsilon, c+\epsilon]$ for some tolerance $\epsilon$. The hill climbing algorithm works by first selecting a random set $\mathcal{D}^{(0)}$ Then, for $k = 1, 2, \ldots$, while $FLP(\mathcal{D}^{(k)}) \notin [c-\epsilon, c+\epsilon]$, swap two nodes between $\mathcal{D}^{(k)}$ and $\mathcal{V}\backslash\mathcal{D}^{(k)}$ and set the new set to $\bar{\mathcal{D}}^{(k)}$. If $|FLP(\mathcal{D}^{(k)}) - c| > |FLP(\bar{\mathcal{D}}^{(k)}) - c|$ then set $\mathcal{D}^{(k+1)} = \bar{\mathcal{D}}^{(k)}$, otherwise $\mathcal{D}^{(k+1)} = \mathcal{D}^{(k)}$.

Before investigating more complex choices for $c_{j,k}$, we try simply using the distance from node $v_k$ to node $v_j$, and setting $\boldsymbol{f}_k = 0$ for $k = 1, \ldots, n$, as well as setting $\ell = 1$. We call this choice the control distance. An example comparing the greedy algorithm described in [138] and the FLP formulation in Eq. (5.7.11) using the control distance cost is shown in Fig. 5.15 for three graphs with $n = 300$ nodes, $n_d = 33$ driver nodes and $n_t = 100$ target nodes. The three graphs are an Erdős-Réyi graph with $\langle\kappa\rangle = 5$ in Fig. 5.15(A), a Barabási-Albert graph with $m = 3$ in Fig. 5.15(B), and a Watts-Strogatz graph with rewiring probability $p = 0.01$ in Fig. 5.15. The black dots are the costs associated with 100 random choices of driver node. For each graph,

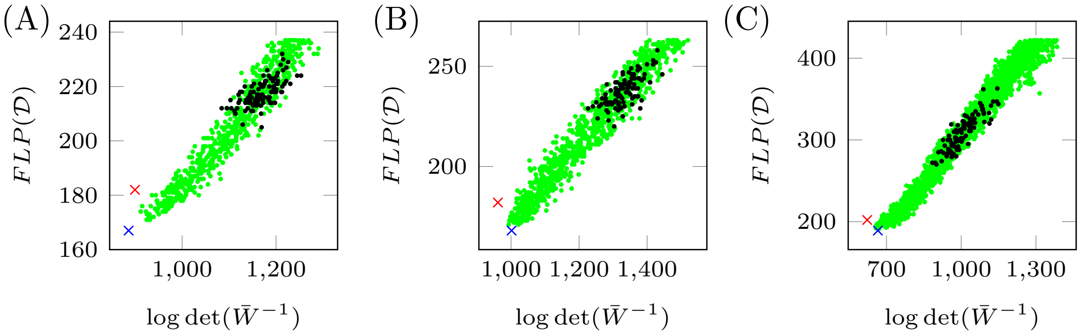


**Figure 5.15:** The relationship between the control distance choice of $FLP(\mathcal{D})$ and the control metric $\log\det\bar{W}^{-1}$. Three graphs with $n = 300$ nodes are examined, (A) an Erdoős-Rényi graph with $p = 1/120$, (B) a Barabási-Albert graph with $m = 3$, and (C) a Watts-Strogatz graph with $p = 0.01$. For each graph, the black dots are the costs associated with one hundred random choices of driver nodes. The green dots are found using the hill climbing algorithm described in the text. The red cross is the cost of the driver node set found using the greedy method described in [138]. The blue cross is the cost of the driver node set found using the control distance cost in Eq. (5.7.11).

the random choices are clustered together and perform poorly. To better understand the relationship between the two costs, the hill climbing algorithm described above is used to find driver node sets with other values of $FLP(\mathcal{D})$ whose cost $\log\det(\bar{W}^{-1})$ is also computed. These values are shown in green where the linear trend between the two costs is seen for all three of the graphs. The red cross is the costs for the driver node set found by the greedy algorithm suggested in [138] where nodes are selected at first that maximize the numerical rank of $\bar{W}$ before switching to add nodes which minimize $\log\det\bar{W}^{-1}$. The blue cross is the costs for the driver node set found by the FLP formulation in Eq. (5.7.11) with the costs equal to the distance between nodes. We see that the FLP out performs random choice and it is competitive with the greedy algorithm.

It appears that the FLP problem is able to perform better than the greedy algorithm for some graphs, vice versa for others. Let $\bar{W}_{FLP}^{-1}$ be the inverse of the output controllability Gramian found using the FLP method and let $\bar{W}_{greedy}^{-1}$ be the inverse of the output controllability Gramian found using the greedy algorithm. In Fig. 5.16(A), 100 Erdős-Rényi graphs with $\langle\kappa\rangle = 5$ are created with $n = 200$ nodes, $n_d = 33$ driver nodes and $n_t = 100$ target nodes. The difference of the performances for the two driver node sets is shown where, if $\log\det(\bar{W}_{FLP}^{-1}) - \log\det(\bar{W}_{greedy}^{-1}) < 0$ then the FLP

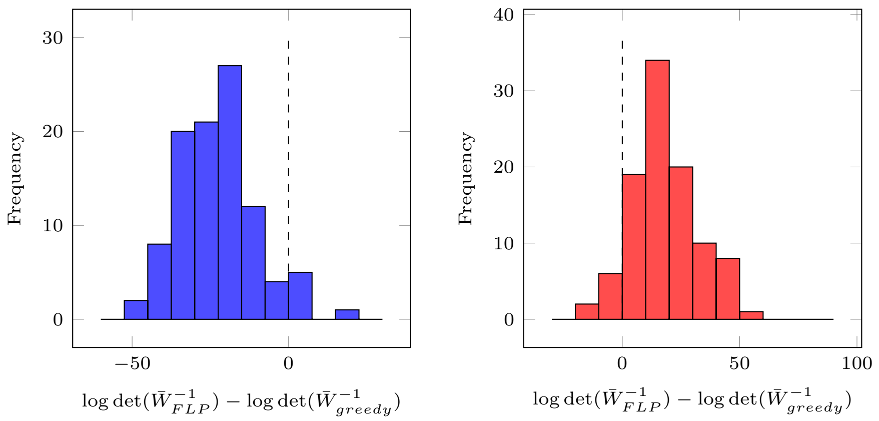


**Figure 5.16:** Difference between the performance of the FLP method and the greedy algorithm. The blue histogram collects the results from 100 Erdős-Rényi graphs with average degree $\langle \kappa \rangle = 5$ and the red histogram collects the results from 100 Watts-Strogatz graphs with rewiring probability $p = 1\%$. All graphs have $n = 200$ nodes, $n_d = 33$ driver nodes and $n_t = 100$ target nodes.

method performed better while if it is positive then the greedy algorithm performed better. For the Erdoős-Rényi graphs, we see the FLP method typically out performs the greedy algorithm. In Fig. 5.16(B), the difference in the performances of the driver node sets found for 100 Watts-Strogatz graphs with rewriring probability $p = 1\%$ is shown.

The promising results demonstrated here for the simple choice of cost matrix suggests that a more informed choice may perform better. Future work will examine the precise differences between the sets of driver nodes returned by the two algorithms.

## 5.8 Conclusion

This chapter investigated, in detail, the role of graph distance and redundancy on the control energy in lattice graphs. Of particular interest were the path graphs which provide the simplest model of graph distance. For the path graph with uniform loop weight $-p$ and edge weight $s$, unidirectional or bidirectional, the control energy required when using a single driver node $v_0$ to control target node $v_j$ is approximately,

$$\log E_j \sim 2(\log p - \log s)$$

A corrective factor was introduced that takes into account *redundancy* (see Def. 5.2.1) which is derived using the unidirectional balloon graph in Eq. (5.4.3).

The results were then compared to the control energy in complex networks and seen to provide a good approximation when the network is very sparse. With this in mind, we develop a driver node selection heuristic that exploits the graph structure using a Facility Location Problem type framework. This is in contrast the currently published controllability Gramian based methods. We show that the structure based heuristic is competitive with the greedy approximation algorithm and that potentially, with a proper choice of the cost matrix, could be made to perform at least as well as the greedy approximation algorithm.

# Chapter 6

# Conclusion

The results in this thesis cover three important areas in the larger field of controlling complex networks. In chapter 3, an optimistic response was made to earlier work which suggested that controlling complex networks with only a few control inputs was too costly. The response was broken into two parts that removed implicit conditions in the earlier work, (i) that every state in the complex network must be controlled and (ii) that the control action must be exact. In Sec. 3.2 the first condition is removed so that only a subset of the nodes in the network have a final constraint, called target nodes, which yields an exponential decrease in the *control energy.* In Sec. 3.3 the second condition is removed so that the final state of the target nodes is satisfied approximately, which we call balanced control, which also yields an exponential decrease in the control energy. These two modification to the original problem provide the conclusion that complex networks can be controlled with a few control inputs if the control action desired is chosen more carefully.

The balanced control problem in Sec. 3.3 is used to address the problem of controlling a network ensemble in Sec. 3.4, that is, a finite set of systems, one of which describes the actual system of interest. This problem can be used to address when the system is unknown due to uncertainty. We find that, with the proper choice of a weighting parameter, we can find controllers capable of performing a control action on an entire network ensemble even when the number of systems grows to infinity.

A geometric interpretation of the minimum energy control problem is used in Sec. 3.5 to derive a piecewise controller capable of driving a nonlinear system to a desired stable attractor. The piecewise controller treats the nonlinear system at each step as a linear system in a local region of state space. This result is a direct response to the previously introduced concept of *nonlocality* that stated minimum energy control actions are not applicable to linearizations of nonlinear systems.

Chapter 4 turns to the topic of *graph symmetries* which have recently been shown to play an important role in the controllability of complex networks and synchronization. While it has long been understood how to find the symmetries in graphs, a mechanism for how to construct a graph with desirable symmetries was unknown. In Sec. 4.3 the conditions which a particular symmetry pattern must satisfy in order to be realized in a graph are derived along with two methods by which one may create a *feasible quotient graph.* With a feasible quotient graph, the method to create a graph with the desired symmetries is described in Sec. 4.4 along with proofs.

An appplication to the *group consensus* problem is described in Sec. 4.7 which uses a block diagonalizing transformation derived from the automorphism group of a graph to decouple synchronous motion from incoherent motion. This transformation allows one to analyze the stability of the synchronous manifold separately from the stability of the overall system.

Graph symmetries are strictly a structural property of a graph and do not take into account edge weights if they exist. Rather than facing the symmetry problem directly, we instead focus on a related problem, that of the *minimum equitable partition.* To address this, we attempt to include the edge weight information in Sec. 4.8 where we define the Minimum Approximate Equitable Partition. Two solution methods are presented, (i) an ILP formulation and (ii) a heuristic. The ILP formulation, even with a number of improvements over the first naive description, suffers from a number of issues that result in its inability to solve the problem on small graphs even when using the Wheeler machine at UNM's CARC. Instead, the heuristic developed is a modification of the deterministic, polynomial time algorithm for finding the minimum equitable partition on unweighted graphs, but it typically does not perform well when compared to those problems the ILP is able to solve. Nonetheless, there is a significant improvement when the heuristic method is used as a pre-processor for the ILP formulation as it can better upper bound the optimal solution and reduce the number of variables required by the ILP.

Finally, in Chapter 5, the relationship between structural properties of the graph and the control energy are investigated. In particular, the controllability Gramian of lattice graphs, that is, graphs which are *translationally invariant*, are shown to be analytically solvable. The bidirectional path graph is chosen for closer inspection as the distance between two nodes is simply the different in their indices and so the graph distance between driver nodes and target nodes appears explicitly in the expression for its controllability Gramian. Beyond graph distance, the role of redundancy is investigated by examining the effect of redundant shortest paths between a driver node and target node. Overall, we are able to explain the exponential growth of the control energy with distance between driver nodes and target nodes as well as the fact redundancy is reduces the control energy independent of the control distance.

The results derived that connect graph structure and control energy is used to form a heuristic for the driver node selection problem. Current methods used to solve this problem are based on the controllability Gramian and ignore the structure of the network. The structure based heuristic developed is found to be competitive with current greedy algorithms while requiring less memory.

# Appendices

# List of Appendices

# Appendix A

# Details for Applied Optimal Control Problems

## A.1 Parameters for the Autophagy Model

The parameters used in the autophagy model in Sec. 2.4.3 as written in Eq. (2.4.11) are collected in Table A.1. These parameters are also used in the application to network ensembles in Sec. 3.4.

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $r_{b,12}$ | 0 | $k_1$ | $1.00 \times 10^{-1}$ |
| $r_{m,12}$ | $1.00 \times 10^{-1}$ | $k_2$ | $3.00 \times 10^{-1}$ |
| $\theta_{12}$ | $3.00 \times 10^{-1}$ | $k_3$ | $4.00 \times 10^{0}$ |
| $n_{12}$ | $4.00 \times 10^{0}$ | $k_4$ | $1.00 \times 10^{-1}$ |
| $r_{b,13}$ | 0 | $\delta_1$ | $3.10 \times 10^{-4}$ |
| $r_{m,13}$ | $1.00 \times 10^{1}$ | $\delta_2$ | $1.93 \times 10^{-3}$ |
| $\theta_{13}$ | $6.00 \times 10^{-1}$ | $\delta_3$ | $5.78 \times 10^{-3}$ |
| $n_{13}$ | $6.00 \times 10^{0}$ | $\delta_4$ | $1.15 \times 10^{-2}$ |
| $r_{b,23}$ | 0 | $\delta_5$ | $2.31 \times 10^{-3}$ |
| $r_{m,23}$ | $6.00 \times 10^{0}$ | $\delta_6$ | $1.16 \times 10^{-3}$ |
| $\theta_{23}$ | $1.00 \times 10^{0}$ | $r_b$ | 0 |
| $n_{23}$ | $4.00 \times 10^{0}$ | $r_m$ | $1.00 \times 10^{0}$ |
| $r_{b,21}$ | $1.00 \times 10^{-1}$ | $\theta$ | $5.00 \times 10^{-1}$ |
| $r_{m,21}$ | $6.00 \times 10^{0}$ | $n$ | $2.00 \times 10^{0}$ |
| $\theta_{21}$ | $6.00 \times 10^{-1}$ | $T$ | $1.00 \times 10^{0}$ |
| $n_{21}$ | $4.00 \times 10^{0}$ | | |
| $r_{b,42}$ | $1.00 \times 10^{-1}$ | | |
| $r_{m,42}$ | $6.00 \times 10^{0}$ | | |
| $\theta_{42}$ | $5.00 \times 10^{-1}$ | | |
| $n_{42}$ | $4.00 \times 10^{0}$ | | |

**Table A.1:** Parameters used for simulations in the autophagy model.

## A.2 Equations and Parameters for the Diabetes Model

The dynamical model that describes the human insulin-glucose-glucagon system consists of a number of differential equations and a number of algebraic equations. Further details of the meaning and units behind each of the equations can be found in the supplementary information of [11].

### Glucose Subsystem

$$\dot{G}_p(t) = EGP(t) + Ra(t) - U_{ii} - E(t) - k_1 G_p(t) + k_2 G_t(t), \quad G_p(0) = G_{pb} \qquad \text{(A.2.1a)}$$

$$\dot{G}_t(t) = -U_{id}(t) + k_1 G_p(t) - k_2 G_t(t), \quad G_t(0) = G_{tb} \qquad \text{(A.2.1b)}$$

$$G(t) = \frac{G_p}{V_G} \qquad \text{(A.2.1c)}$$

### Insulin Subsystem

$$\dot{I}_p(t) = -(m_2 + m_4) I_p(t) + m_1 I_\ell(t) + R_{ia}(t), \quad I_p(0) = I_{pb} \qquad \text{(A.2.2a)}$$

$$\dot{I}_\ell(t) = -(m_1 + m_3) I_\ell(t) + m_2 I_p(t), \quad I_\ell(0) = I_{\ell b} \qquad \text{(A.2.2b)}$$

$$I(t) = \frac{I_p(t)}{V_I} \qquad \text{(A.2.2c)}$$

### Glucose Rate of Appearance

$$Q_{sto}(t) = Q_{sto1}(t) + Q_{sto2}(t), \qquad Q_{sto}(0) = 0 \tag{A.2.3a}$$

$$\dot{Q}_{sto1}(t) = -k_{gri}Q_{sto1}(t) + D\delta(t - \tau_D), \qquad Q_{sto1}(0) = 0 \tag{A.2.3b}$$

$$\dot{Q}_{sto2}(t) = -k_{empt}(Q_{sto}(t))Q_{sto2}(t) + k_{gri}Q_{sto1}(t), \qquad Q_{sto2}(0) = 0 \tag{A.2.3c}$$

$$\dot{Q}_{gut}(t) = -k_{abs}Q_{gut}(t) + k_{empt}Q_{sto}(t)Q_{sto2}(t), \qquad Q_{gut}(0) = 0 \tag{A.2.3d}$$

$$Ra(t) = \frac{fk_{abs}Q_{gut}(t)}{BW}, \qquad Ra(0) = 0 \tag{A.2.3e}$$

$$k_{empt}(Q_{sto}) = k_{\min} + \frac{k_{\max} - k_{\min}}{2} \tag{A.2.3f}$$

$$\times \left[\tanh\left(\alpha(Q_{sto} - bD\right) - \tanh\left(\beta(Q_{sto} - cD)\right) + 2\right] \tag{A.2.3g}$$

### Endogeneous Glucose Production

$$EGP(t) = k_{p1} - k_{p2}G_p(t) - k_{p3}X^L(t) + \xi X^H(t), \qquad EGP(0) = EGP_b \tag{A.2.4a}$$

$$\dot{I}'(t) = -k_i[I'(t) - I(t)], \qquad I'(0) = I_b \tag{A.2.4b}$$

$$\dot{X}^L(t) = -k_i[X^L(t) - I'(t)], \qquad X^L(0) = I_b \tag{A.2.4c}$$

$$\dot{X}^H(t) = -k_H X^H(t) + k_H \max\{H(t) - H_b, 0\}, \qquad X^H(0) = 0 \tag{A.2.4d}$$

### Glucose Utilization

$$U_{ii}(t) = F_{cns} \tag{A.2.5a}$$

$$U_{id}(t) = \frac{[V_{m0} + V_{mx}X(t)]G_t(t)}{K_{m0} + G_t(t)} \tag{A.2.5b}$$

$$\dot{X}(t) = -p_{2U}X(t) + p_{2U}[I(t) - I_b], \qquad X(0) = 0 \tag{A.2.5c}$$

### Renal Excretion

$$E(t) = \begin{cases} k_{e1}[G_p(t) - k_{e2}, & \text{if} \quad G_p(t) > k_{e2} \\ 0, & \text{if} \quad G_p(t) \leq k_{e2} \end{cases} \tag{A.2.6a}$$

### Glucagon Kinetics and Secretion

$$\dot{H}(t) = -nH(t) + SR_H(t) = Ra_H(t), \qquad H(0) = H_b \tag{A.2.7a}$$

$$SR_H(t) = SR_H^s(t) + SR_H^d(t) \tag{A.2.7b}$$

$$\dot{SR}_H^s(t) = -\rho \left[ SR_H^s(t) - \max \left\{ \frac{\sigma[G_{th} - G(t)]}{\max\{I(t) - I_{th}, 0\} + 1} + SR_H^b, 0 \right\} \right], \quad SR_H^s(0) = nH_b \tag{A.2.7c}$$

$$SR_H^d(t) = \delta \max \left\{ -\frac{dG(t)}{dt}, 0 \right\} \tag{A.2.7d}$$

### Subcutaneous Insulin Kinetics

$$R_{ia}(t) = k_{a1} I_{sc1}(t) + k_{a2} I_{sc2}(t) \tag{A.2.8a}$$

$$\dot{I}_{sc1}(t) = -(k_d + k_{a1}) I_{sc1}(t) + IIR(t), \qquad I_{sc1}(0) = I_{sc1ss} \tag{A.2.8b}$$

$$\dot{I}_{sc2}(t) = k_d I_{sc1}(t) - k_{a2} I_{sc2}(t), \qquad I_{sc2}(0) = I_{sc2ss} \tag{A.2.8c}$$

$$IIR(t) = IIR_b + \frac{u_I(t)}{BW} \tag{A.2.8d}$$

### Subcutaneous Glucagon Kinetics

$$\dot{H}_{sc1}(t) = -(k_{h1} + k_{h2}) H_{sc1}(t) + GIR(t), \qquad H_{sc1}(0) = H_{sc1ss} \tag{A.2.9a}$$

$$\dot{H}_{sc2}(t) = k_{h1} H_{sc1}(t) - k_{h3} H_{sc2}(t), \qquad H_{sc2}(0) = H_{sc2ss} \tag{A.2.9b}$$

$$Ra_H(t) = k_{h3} H_{sc2}(t) \tag{A.2.9c}$$

$$GIR(t) = GIR_b + \frac{u_G(t)}{BV} \tag{A.2.9d}$$

Details on the parameters and basal values can be found in the supplementary tables of [11] as well as discussions in [247, 248].

# Appendix B

# Derivations Related to Target Control

## B.1 Combined Target Problem

### B.1.1 Problem Statement

In chapter 3, the problem of *target control* is discussed at length. Specifically, we consider two problems, one where there is a constraint applied to the final target nodes' states and one where there is a cost associated with the final target nodes' states. These two formulations can be combined into a general problem solved using Pontryagin's minimum principle, described in Section 2.4.1. The result for the general problem is then specialized for the particular cases addressed in the main text.

### B.1.2 Derivation of the Optimal Control and Optimal State Trajectory

The general problem consists of the following matrices; the state matrix $A \in \mathbb{R}^{n \times n}$, the input matrix $B \in \mathbb{R}^{n \times n_d}$, the final constraint matrix $C_1 \in \mathbb{R}^{n_{t,1} \times n}$, and the final cost matrix $C_2 \in \mathbb{R}^{n_{t,2} \times n}$, where $n_{t,1} + n_{t,2} \leq n$. Additionally, without loss of generality, we require that the rows of $C_1$ and $C_2$ are *orthogonal*, that is, $C_1 C_2^T = O_{n_{t,1}, n_{t,2}}$. This is because any row of $C_2$ that lies in the row-space of $C_1$ is constrained already, and adding it as a cost is redundant. The Gram-Schmidt process [170] can be used to find an alternative matrix $C_2$ that is orthogonal with respect to $C_1$.

Besides the matrices that describe the system, $A, B, C_1, C_2$, we introduce the cost matrices, $F$ and $R$, that are symmetric and positive definite, that weight the importance of the final costs and the control inputs. The two cost terms are also weighted with respect to each other using a parameter $\alpha \in (0, 1)$. The complete optimal control problem can now be stated:

$$\begin{aligned}
\min \quad & J = \frac{1-\alpha}{2}\left(C_2\boldsymbol{x}(t_f) - \boldsymbol{y}_{f,2}\right)^T F \left(C_2\boldsymbol{x}(t_f) - \boldsymbol{y}_{f,2}\right) + \frac{\alpha}{2}\int_0^{t_f} \mathbf{u}^T(t)R\mathbf{u}(t)dt, \quad \alpha \in [0,1] \\
\text{s.t.} \quad & \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \\
& \mathbf{x}(0) = \mathbf{x}_0, \quad \boldsymbol{y}_{f,1} = C_1\boldsymbol{x}(t_f)
\end{aligned} \tag{B.1.1}$$

The desired final output vectors, $\boldsymbol{y}_{f,1} \in \mathbb{R}^{n_{t,1}}$ and $\boldsymbol{y}_{f,2} \in \mathbb{R}^{n_{t,2}}$, are chosen to be the desired values of each of the targets. In the network framework described previously, the state matrix $A$ is the adjacency matrix of the graph, the input matrix $B$ has unit vectors as columns corresponding to the driver nodes, and the matrices $C_1$ and $C_2$ have unit vectors as rows corresponding to which target nodes have constrained final states and which have final costs.

### B.1.3 Solution via Pontryagin

To solve the optimal control problem in Eq. (B.1.1), the minimum principle of Pontryagin is used as described in Sec. 2.4.1. The Hamiltonian to minimize from Eq. (2.4.2) is specialized to be,

$$\min \quad \frac{\alpha}{2}\boldsymbol{u}^T(t)R\boldsymbol{u}(t) + \boldsymbol{\lambda}^T(t)A\boldsymbol{x}(t) + \boldsymbol{\lambda}^T(t)B\boldsymbol{u}(t) \tag{B.1.2}$$

The optimal input is found from the stationarity condition in Eq. (2.4.4), by differentiating Eq. (B.1.2) with respect to the inputs.

$$\alpha R\boldsymbol{u}(t) + B^T\boldsymbol{\lambda}(t) = \mathbf{0} \quad \Rightarrow \quad \boldsymbol{u}(t) = -\frac{1}{\alpha}R^{-1}B^T\boldsymbol{\lambda}(t) \tag{B.1.3}$$

The costates, $\boldsymbol{\lambda}(t)$, are found from Eq. (2.4.5), by differentiating Eq. (B.1.2) with respect to the states.

$$\dot{\boldsymbol{\lambda}}(t) = -A^T \boldsymbol{\lambda}(t) \quad \Rightarrow \quad \boldsymbol{\lambda}(t) = e^{A^T(t_f - t)} \boldsymbol{\lambda}(t_f) \tag{B.1.4}$$

The end-point minimization problem in Eq. (2.4.3) is specialized to be,

$$\begin{aligned} \min \quad & \frac{1-\alpha}{2} \left(C_2 \boldsymbol{x}(t_f) - \boldsymbol{y}_{f,2}\right)^T F \left(C_2 \boldsymbol{x}(t_f) - \boldsymbol{y}_{f,2}\right) \\ \text{s.t.} \quad & C_1 \boldsymbol{x}(t_f) - \boldsymbol{y}_{f,1} = \mathbf{0} \\ & \boldsymbol{x}(0) - \boldsymbol{x}_0 = \mathbf{0} \end{aligned} \tag{B.1.5}$$

The final costate values are found by differentiating the Lagrangian of the end-point minimization problem in Eq. (B.1.5), by introduction $n_t$ Lagrange multipliers $\boldsymbol{\nu}_{f,k}$, $k = 1, \ldots, n_t$, with respect to the final states $\boldsymbol{x}(t_f)$,

$$(1-\alpha) C_2^T F (C_2 \boldsymbol{x}(t_f) - \boldsymbol{y}_{f,2}) + C_1^T \boldsymbol{\nu}_f = \boldsymbol{\lambda}(t_f) \tag{B.1.6}$$

To simplify the following derivations, we define the difference between the final outputs that appear in the cost and their desired values as,

$$\boldsymbol{\gamma}_f = C_2 \boldsymbol{x}(t_f) - \boldsymbol{y}_{f,2} \tag{B.1.7}$$

The controlled state trajectory for arbitrary control input is determined with a convolution integral, which is then specialized using the optimal control found in Eq. (B.1.3), the costate evolution in Eq. (B.1.4), and the final values of the costates in

Eq. (B.1.6) using $\boldsymbol{\gamma}_f$ as defined in Eq. (B.1.7),

$$\begin{aligned}\boldsymbol{x}(t) &= e^{At}\boldsymbol{x}_0 + \int_0^t e^{A(t-\tau)}B\boldsymbol{u}(\tau)d\tau \\ &= e^{At}\boldsymbol{x}_0 - \frac{1}{\alpha}\int_0^t e^{A(t-\tau)}BR^{-1}B^T\boldsymbol{\lambda}(\tau)d\tau \\ &= e^{At}\boldsymbol{x}_0 - \frac{1}{\alpha}\int_0^t e^{A(t-\tau)}BR^{-1}B^T e^{A^T(t_f-\tau)}d\tau\boldsymbol{\lambda}(t_f) \\ &= e^{At}\boldsymbol{x}_0 - \frac{1}{\alpha}W(t)e^{A^T(t_f-t)}\left((1-\alpha)C_2^T F\boldsymbol{\gamma}_f + C_1^T\boldsymbol{\nu}_f\right)\end{aligned} \tag{B.1.8}$$

The matrix $W(t)$ is the solution of the differential Lyapunov equation in Eq. (2.3.8), repeated here for clarity,

$$\dot{W}(t) = AW(t) + W(t)A^T + BR^{-1}B^T, \quad W(0) = O_n \tag{B.1.9}$$

which has formal solution,

$$W(t) = \int_0^t e^{A\tau}BR^{-1}B^T e^{A^T\tau}d\tau \tag{B.1.10}$$

To determine $\nu_f$ and $\boldsymbol{\gamma}_f$, we enforce the final constraints by pre-multiplying Eq. (B.1.8) by $C_1$ and evaluate the expression at $t = t_f$.

$$C_1\boldsymbol{x}(t_f) = C_1e^{At_f}\boldsymbol{x}_0 - \frac{1-\alpha}{\alpha}C_1W(t_f)C_2^T F\boldsymbol{\gamma}_f - \frac{1}{\alpha}C_1W(t_f)C_1^T\boldsymbol{\nu}_f = \boldsymbol{y}_{1,f} \tag{B.1.11}$$

To determine the other necessary equations, we pre-multiply Eq. (B.1.8) by $C_2$, evaluate at time $t = t_f$, and subtract $\boldsymbol{y}_{f,2}$ from both sides.

$$C_2\boldsymbol{x}(t_f) - \boldsymbol{y}_{f,2} = C_2e^{At_f}\boldsymbol{x}_0 - \frac{1-\alpha}{\alpha}C_2W(t_f)C_2^T F\boldsymbol{\gamma}_f - \frac{1}{\alpha}C_2W(t_f)C_1^T\boldsymbol{\nu}_f - \boldsymbol{y}_{f,2} = \boldsymbol{\gamma}_f \tag{B.1.12}$$

The full system of equations is composed by combining Eq. (B.1.11) and (B.1.12) into the linear system where the matrix $W(t_f)$ is found by either integration Eq. (B.1.9) or numerically solving Eq. (B.1.10). Let $W_{j,k}(t_f) = C_jW(t_f)C_k^T$ for $j, k = 1, 2$ so the system becomes,

$$\begin{bmatrix} -\frac{1}{\alpha}W_{1,1}(t_f) & -\frac{1-\alpha}{\alpha}W_{1,2}(t_f)F \\ -\frac{1}{\alpha}W_{2,1}(t_f) & -\frac{1-\alpha}{\alpha}W_{2,2}(t_f)F - I_{n_{t,2}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\nu}_f \\ \boldsymbol{\gamma}_f \end{bmatrix} = \begin{bmatrix} \boldsymbol{y}_{f,1} - C_1 e^{At_f}\boldsymbol{x}_0 \\ \boldsymbol{y}_{f,2} - C_2 e^{At_f}\boldsymbol{x}_0 \end{bmatrix} \tag{B.1.13}$$

This linear system can be simplified, and made symmetric, by pre-multiplying both sides by

$$-\alpha \begin{bmatrix} I_{n_{t,1}} & O_{n_{t,1},n_{t,2}} \\ O_{n_{t,2},n_{t,1}} & (1-\alpha)F \end{bmatrix},$$

so the system in Eq. (B.1.13) becomes,

$$\begin{aligned} &\begin{bmatrix} W_{1,1}(t_f) & (1-\alpha)W_{1,2}(t_f)F \\ (1-\alpha)FW_{2,1}(t_f) & (1-\alpha)^2 FW_{2,2}(t_f)F + \alpha(1-\alpha)F \end{bmatrix} \begin{bmatrix} \boldsymbol{\nu}_f \\ \boldsymbol{\gamma}_f \end{bmatrix} \\ &= \alpha \begin{bmatrix} C_1 e^{At_f}\boldsymbol{x}_0 - \boldsymbol{y}_{f,1} \\ (1-\alpha)F(C_2 e^{At_f}\boldsymbol{x}_0 - \boldsymbol{y}_{f,2}) \end{bmatrix} \end{aligned} \tag{B.1.14}$$

With just a small example graph, some of the complicated behavior that arises from this solution can be seen. The two costs of interest are the control energy,

$$E = \int_0^{t_f} \boldsymbol{u}^T(t)\boldsymbol{u}(t)dt \tag{B.1.15}$$

and the deviation,

$$D = \boldsymbol{\gamma}_f^T \boldsymbol{\gamma}_f \tag{B.1.16}$$

Both of these expressions are implicit functions of the weighting parameter $\alpha$ which appears in Eq. (B.1.14) whose solution is necessary to evaluate Eqs. (B.1.15) and (B.1.16). In Fig. B.1(A) a seven node network is shown with all edge weights set equal to 1 and self-loops at every node equal to -4. The light blue nodes are the driver nodes, $\mathcal{D} = \{v_1, v_2\}$, the first set of target nodes with final constraints are colored magenta, $\mathcal{T} = \{v_3, v_4\}$, and the second set of target nodes with final costs are colored green, $\mathcal{T}' = \{v_0, v_6\}$. Node $v_5$ is not in any of the subsets. The other

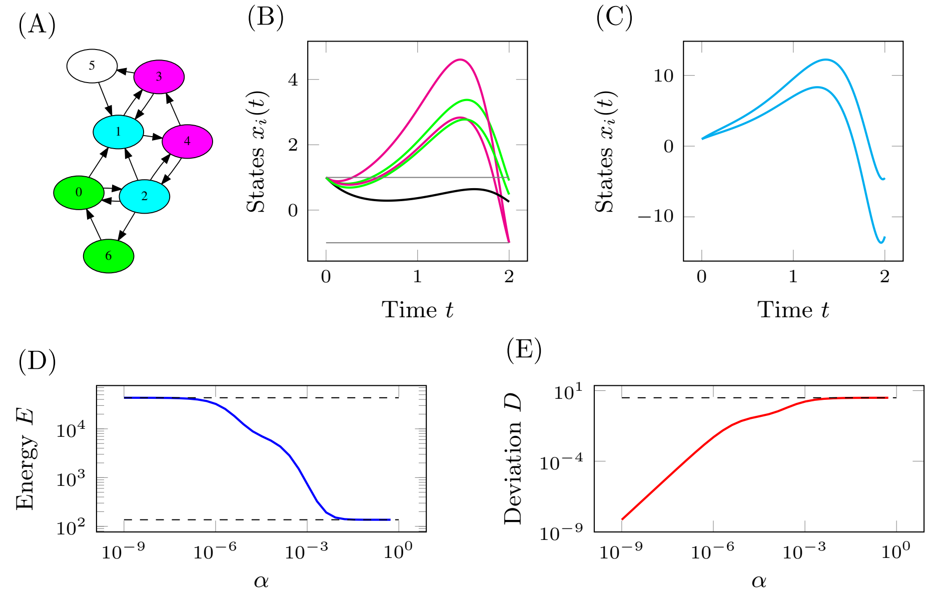


**Figure B.1:** An Example of the Mixed Strategy with Hard and Soft Final Constraints. (A) A seven node network with all edge weights equal to 1 and self-loops at every node with weight equal to -4. The light blue nodes are the driver nodes $\mathcal{D}$, the magenta nodes $\mathcal{T}$ have a final constrinat and the green nodes $\mathcal{T}'$ have a final cost. Node $v_5$ is not in any of the subsets. (B) An example of the time trajectories subject to the optimal control input for all states associated with non-driver nodes. In this simulation $\alpha = 10^{-4}$. (C) The time trajectors of the driver nodes. Note that the driver nodes' states have a larger maximum and minimum state than the non-driver states which is typical for this type of control. (D) The optimal control energy $E(\alpha)$ for values of $\alpha$ spanning multiple orders of magnitude. The dashed lines correspond to the control energy for the case $\mathcal{T} = \{v_0, v_3, v_4, v_6\}$ (upper dashed line) and $\mathcal{T} = \{v_3, v_4\}$ (lower dashed line) with $\mathcal{T}' = \emptyset$. (E) The optimal deviation for the same values of $\alpha$. The dashed line corresponds to the deviation of nodes $v_0$ and $v_6$ when $\mathcal{T} = \{v_3, v_4\}$ and $\mathcal{T}' = \emptyset$.

parameters needed for the simulation are $\boldsymbol{x}_0 = \mathbf{1}_7$, $t_f = 2$, $\boldsymbol{x}_{1,f} = -\mathbf{1}_2$, $\boldsymbol{x}_{2,f} = \mathbf{1}_2$, and $\alpha = 10^{-4}$. In Figs. B.1(B) and B.1(C) the time evolution of the states of the non-driver nodes and the driver nodes are plotted. The states corresponding to the magenta nodes (those with final contraints) can be seen to equal $x_3(t_f) = x_4(t_f) = -1$ at the final time $t_f = 2$ (represented by the lower gray line). On the other hand, the states with final costs (the green nodes) do not quite reach the desired final condition, $x_0(t_f) \approx 1 \approx x_6(t_f)$ as represented by the upper gray line. The black line is the time evolution of node $v_5$. The time evolution of the driver nodes is shown separately as the maximum and minimum values are more than twice as large as the states of any of the non-driver nodes which is typical in this type of control. Letting $\alpha$ vary over multiple orders of magnitude in Figs. B.1(D) and B.1(E) shows the behavior of $E(\alpha)$ and $D(\alpha)$ found using Eqs. (B.1.15) and (B.1.16). First, the control energy decreases monotonically as $\alpha \to 1$ while remaining bounded between the dashed lines which represent the control energy for the cases $\mathcal{T} = \{v_0, v_3, v_4, v_6\}$ (setting all target nodes into the final constraint set), the upper dashed line, and $\mathcal{T} = \{v_3, v_4\}$ (removing the final cost target nodes from either target node set), the lower dashed line. Similarly, in Fig. B.1(E) the deviation, which in this case is $D(\alpha) = (x_0(t_f) - 1)^2 + (x_6(t_f) - 1)^2$, we see $D$ monotonically increases as $\alpha \to 1$ while remaining bounded between $D(0) = 0$ (assuming target controllability) and $D(1)$ which corresponds to evaluating $D$ when there are no nodes with final costs.

With these general results, the solution described is specialized to the two cases, one where $n_{t,1} > 0$ and $n_{t,2} = 0$ which we call *target control* which is covered in Sec. 3.2 in the main text and Appendix B.2 here, and one where $n_{t,1} = 0$ and $n_{t,2} > 0$ which we call *balance control* covered in Sec. 3.3 in the main text and Appendix B.3 here.

## B.2 Specialization to Target Control

For target control, we consider the case when $n_{t,1} > 0$ and $n_{t,2} = 0$, that is, every final output is a constraint. We simplify the equations so that $n_t = n_{t,1}$, $C = C_1$,

$\boldsymbol{y}_{f,1} = \boldsymbol{y}_f$, $\boldsymbol{\beta} = Ce^{At_f} - \boldsymbol{y}_f$ and the linear system in Eq. (B.1.14) is simply,

$$\left(CW(t_f)C^T\right)\boldsymbol{\nu}_f = \alpha\boldsymbol{\beta} \tag{B.2.1}$$

It will be clear that the choice of $\alpha$ is arbitrary as when Eq. (B.2.1) is used to fully define the optimal contorl input we see that,

$$\boldsymbol{u}(t) = -R^{-1}B^T e^{A^T(t_f-t)}C^T\left(CW(t_f)C^T\right)^{-1}\boldsymbol{\beta} \tag{B.2.2}$$

In the results in the main text, we assume $R = I_{n_d}$, which we assume in the following derivations. Let $\bar{W} = CW(t_f)C^T$ be the *output controllability Gramian* so that, with the optimal control in Eq. (B.2.2), the control energy can be determined,

$$\begin{aligned}
E &= \boldsymbol{\nu}_f^T C\int_0^{t_f} e^{A\tau}BB^Te^{A^T\tau}d\tau C^T\boldsymbol{\nu}_f \\
&= \boldsymbol{\beta}^T\bar{W}^{-1}C\int_0^{t_f} e^{A\tau}BB^Te^{A\tau}d\tau C^T\bar{W}^{-1}\boldsymbol{\beta} \\
&= \boldsymbol{\beta}^T\bar{W}^{-1}\boldsymbol{\beta}
\end{aligned} \tag{B.2.3}$$

It is Eq. (B.2.3) that is investigated in detail in [1] for the case that $A$ is a Hurwitz adjacency matrix of a graph, $B$ has distinct unit vectors as columns corresponding to the set of driver nodes and $C$ has distinct unit vectors as rows corresponding to the set of target nodes. Further details are contained in Sec. 3.2 in the main text.

## B.3 Specialization to Balanced Control

For balanced control, $n_{t,1} = 0$ while $n_{t,2} > 0$, so that we set $n_t = n_{t,2}$, $C = C_2$, $\boldsymbol{y}_f = \boldsymbol{y}_{f,2}$, and $\boldsymbol{\beta} = Ce^{At_f}\boldsymbol{x}_0 - \boldsymbol{y}_f$. The results discussed in Sec. 3.3 focus on the case $F = I_{n_t}$ and so this assumption is used in the following derivations. The expression in Eq. (B.1.14) is simplified to,

$$((1-\alpha)CW(t_f)C^T + \alpha I)\boldsymbol{\gamma}_f = \alpha\boldsymbol{\beta} \tag{B.3.1}$$

which does not require that the triplet $(A, B, C)$ be output controllable as was necessary for target control. Define the matrix $\bar{U}(\alpha) = (1-\alpha)CW(t_f)C^T + \alpha I$ which is similar to the matrix $CW(t_f)C^T$. The eigenpairs of $CW(t_f)C^T$ are defined as $(\mu_k, \boldsymbol{\xi}_k)$, such that $(CW(t_f)C^T)\boldsymbol{\xi}_k = \mu_k \boldsymbol{\xi}_k$ for $k = 1, \ldots, n_t$. For each eigenvalue $\mu_k$ of $CW(t_f)C^T$, there is a corresponding eigenvalue of $\bar{U}(\alpha)$ denoted $\nu_k = (1-\alpha)\mu_k + \alpha$.

With Eq. (B.3.1), the optimal control input in Eq. (B.1.3) is specialized to become,

$$\begin{aligned} \boldsymbol{u}(t) &= -\frac{1-\alpha}{\alpha} B^T e^{A^T(t_f - t)} C^T \boldsymbol{\gamma}_f \\ &= -(1-\alpha)^2 B^T e^{A^T(t_f-t)} C^T \bar{U}^{-1}(\alpha) \boldsymbol{\gamma}_f \end{aligned} \tag{B.3.2}$$

The control input is similar to the expression in Eq. (B.2.2) except with the dependence on the weighting parameter $\alpha$. If $\alpha = 1$, it is clear that $\boldsymbol{u}(t) = \mathbf{0}$ while if $\alpha = 0$, and $CW(t_f)C^T$ is invertible, then Eq. (B.3.2) becomes the control input in Eq. (B.2.2).

The control energy in Eq. (B.1.15) is specialized for this case using Eq. (B.3.1),

$$\begin{aligned} E(\alpha) &= \int_0^{t_f} \boldsymbol{u}^T(t)\boldsymbol{u}(t)dt \\ &= \frac{(1-\alpha)^2}{\alpha^2} \boldsymbol{\gamma}_f^T C \int_0^{t_f} e^{A\tau} B B^T e^{A^T \tau} d\tau C^T \boldsymbol{\gamma}_f \\ &= (1-\alpha)^2 \boldsymbol{\beta}^T \bar{U}^{-1}(\alpha) \bar{W} \bar{U}^{-1}(\alpha) \boldsymbol{\beta} \end{aligned} \tag{B.3.3}$$

and the deviation in Eq. (B.1.16) is specialized as well,

$$\begin{aligned} D(\alpha) &= \boldsymbol{\gamma}_f^T \boldsymbol{\gamma}_f \\ &= \alpha^2 \boldsymbol{\beta}^T \bar{U}^{-1}(\alpha) \bar{U}^{-1}(\alpha) \boldsymbol{\beta} \end{aligned} \tag{B.3.4}$$

The expressions in Eqs. (B.3.3) and (B.3.4) are combined to derive the total cost that appears as the cost function in Eq. (B.1.1),

$$\begin{aligned} J(\alpha) &= \frac{1-\alpha}{2} D(\alpha) + \frac{\alpha}{2} E(\alpha) \\ &= \frac{1-\alpha}{2} \boldsymbol{\gamma}_f^T \boldsymbol{\gamma}_f + \frac{\alpha}{2} \boldsymbol{\gamma}_f^T \bar{W} \boldsymbol{\gamma}_f \\ &= \frac{\alpha(1-\alpha)}{2} \boldsymbol{\beta}^T \bar{U}^{-1}(\alpha) \left(\alpha I + (1-\alpha)\bar{W}\right) \bar{U}^{-1}(\alpha) \boldsymbol{\beta} \\ &= \frac{\alpha(1-\alpha)}{2} \boldsymbol{\beta}^T \bar{U}^{-1}(\alpha) \boldsymbol{\beta} \end{aligned} \tag{B.3.5}$$

These three costs in Eqs. (B.3.3), (B.3.4), and (B.3.5), are all quadratic forms of *similar* matrices, and can be expressed as summations over the eigenvalues $\mu_k$ and eigenvectors $\theta_k = \boldsymbol{\beta}^T \boldsymbol{\xi}_k$. The deviation, control energy, and total cost become,

$$D(\alpha) = \alpha^2 \sum_{k=1}^{n_t} \frac{\theta_k^2}{(\alpha + (1-\alpha)\mu_k)^2} \tag{B.3.6a}$$

$$E(\alpha) = (1-\alpha)^2 \sum_{k=1}^{n_t} \frac{\theta_k^2 \mu_k}{(\alpha + (1-\alpha)\mu_k)^2} \tag{B.3.6b}$$

$$J(\alpha) = \frac{\alpha(1-\alpha)}{2} \sum_{k=1}^{n_t} \frac{\theta_k^2}{\alpha + (1-\alpha)\mu_k} \tag{B.3.6c}$$

These summations in Eqs. (B.3.6a), (B.3.6b), and (B.3.6c) are more useful in explaining the behavior of the optimal solution as $\alpha$ is varied between 0 and 1 and they are used in many of the derivations in Sec. 3.3.

## B.4 Controlling Network Ensembles: Derivations

In the network ensemble framework, we consider that the state matrix is one of $N$ possibilities, denoted $A_k$, $k = 0, \dots, N-1$. The input matrix and output matrix remain the same, $B$ and $C$. The composite system is constructed in the following way,

$$\bar{A} = \begin{bmatrix} A_0 & O_n & \cdots & O_n \\ O_n & A_1 & \cdots & O_n \\ \vdots & \vdots & \ddots & \vdots \\ O_n & O_n & \cdots & A_{N-1} \end{bmatrix} \quad \bar{B} = \begin{bmatrix} B \\ B \\ \vdots \\ B \end{bmatrix} \tag{B.4.1}$$
$$\bar{C} = \begin{bmatrix} C & C & \cdots & C \end{bmatrix}$$

The network ensemble derivations use the balanced control method derived in Sec. B.3 for the composite system. The special form of the composite state matrices in Eq. (B.4.1) leads to the output controllability Gramian having a special form as well, which we call the composite output controllability Gramian.

$$\bar{W} = \begin{bmatrix} CW_{0,0}(t_f)C^T & CW_{0,1}(t_f)C^T & \cdots & CW_{0,N-1}(t_f)C^T \\ CW_{1,0}(t_f)C^T & CW_{1,1}(t_f)C^T & \cdots & CW_{1,N-1}(t_f)C^T \\ \vdots & \vdots & \ddots & \vdots \\ CW_{N-1,0}(t_f)C^T & CW_{N-1,1}(t_f)C^T & \cdots & CW_{N-1,N-1}(t_f)C^T \end{bmatrix} \tag{B.4.2}$$

where each block $W_{j,k}(t_f)$ in Eq. (B.4.2) is the solution of the differential Sylvester equation,

$$\dot{W}_{j,k}(t) = A_j W_{j,k}(t) + W_{j,k}(t) A_k^T + BB^T, \quad W_{j,k}(0) = O_n. \tag{B.4.3}$$

Similar to the differential Lyapunov equation in Eq. (B.1.9), Eq. (B.4.3) has formal solution,

$$W_{j,k}(t) = \int_0^t e^{A_j \tau} BB^T e^{A_k^T \tau} d\tau$$

which may or may not be a useful form depending on the difficult of computing the matrix exponentials for $A_j$ and $A_k$. The summation forms of the costs in Eq. (B.3.6) are repeated here for completeness, with the modification that the deviation is normalized with the number of state matrices, $N$.

$$\begin{aligned} D_N(\alpha)/N &= \frac{\alpha^2}{N}\sum_{k=0}^{Nn_t-1}\frac{\theta_k^2}{(\alpha+(1-\alpha)\mu_k)^2} \\ E_N(\alpha) &= (1-\alpha)^2\sum_{k=0}^{Nn_t-1}\frac{\mu_k\theta_k^2}{(\alpha+(1-\alpha)\mu_k)^2} \\ J_N(\alpha) &= \frac{(1-\alpha)\alpha}{2}\sum_{k=0}^{Nn_t-1}\frac{\theta_k^2}{\alpha+(1-\alpha)\mu_k} \end{aligned} \tag{B.4.4}$$

In Sec. 3.4, the following form of $\alpha$ is selected as it leads to predictable behavior of the three costs.

$$\alpha(b) = \frac{Nn_t}{Nn_t+b} \tag{B.4.5}$$

Applying the form of $\alpha$ in Eq. (B.4.5) to the summation forms of the costs in Eq. (B.4.4) yields the expressions in terms of the new parameter $b$,

$$\begin{aligned} D_N(b)/N &= Nn_t^2\sum_{k=0}^{Nn_t-1}\frac{\theta_k^2}{(Nn_t+b\mu_k)^2} \\ E_N(b) &= b^2\sum_{k=0}^{Nn_t-1}\frac{\mu_k\theta_k^2}{(Nn_t+b\mu_k)^2} \\ J_N(b) &= \frac{Nn_tb}{2(Nn_t+b)}\sum_{k=0}^{Nn_t-1}\frac{\theta_k^2}{Nn_t+b\mu_k} \end{aligned} \tag{B.4.6}$$

At this point, to move further, some idea of how $\mu_k$ and $\theta_k^2$ change as a function of both $k$ and $N$ is required. The two assumptions laid out in Sec. 3.4 are repeated here for clarity.

**Assumption 1:** $\mu_k \approx \mu_0 r_1^k, \qquad \mu_0 \approx c_1 Nn_t$

**Assumption 2:** $\theta_k^2 \approx \max\{\theta_0^2 r_2^k, \theta_c^2\}, \quad \theta_0^2 \approx c_2 Nn_t$

With these assumptions, we can find the index $k = \bar{k}$ when $\theta_k^2$ stops decaying exponentially and switches to a constant value by solving the following inequality for the largest integer $k$ for which it holds.

$$c_2 Nn_t r_2^k \le \theta_c^2 \quad \Rightarrow \quad k \le \frac{\log\theta_c^2 - \log c_2 Nn_t}{\log r_2} \tag{B.4.7}$$

The value of $\bar{k}$ is the largest integer that satisfies the right inequality in Eq. (B.4.7). Note that for large $N$, $\bar{k} \sim \log N$ which will be important in the following derivations. Applying Assumptions 1 and 2 to the summation form of the average deviation in Eq. (B.4.6) yields the following approximate forms,

$$\begin{aligned} D_N(b)/N &\approx Nn_t^2 \sum_{k=0}^{Nn_t-1} \frac{\max\{c_2 Nn_t r_2^k, \theta_c^2\}}{(Nn_t + bc_1 Nn_t r_1^k)^2} \\ &= c_2 n_t \sum_{k=0}^{\bar{k}} \frac{r_2^k}{(1+bc_1 r_1^k)^2} + \frac{\theta_c^2}{N} \sum_{k=\bar{k}+1}^{Nn_t-1} \frac{1}{(1+bc_1 r_1^k)^2} \end{aligned}$$

In this form, we can upper bound the average deviation,

$$D_N(b)/N \leq c_2 n_t \sum_{k=0}^{\bar{k}} r_2^k + \frac{\theta_c^2}{N} \sum_{k=\bar{k}+1}^{Nn_t-1} 1 = c_2 n_t \frac{1-r_2^{\bar{k}+1}}{1-r_2} + \frac{\theta_c^2}{N}\left(Nn_t - \bar{k} + 1\right)$$

which, in the limit of large $N$, becomes

$$\lim_{N\to\infty} D_N(b)/N \leq \frac{c_2 n_t}{1-r_2} + \theta_c^2 n_t \tag{B.4.8}$$

Similarly, applying the assumptions to the control energy yields the expression,

$$\begin{aligned} E_N(b) &\approx \frac{b^2}{N^2 n_t^2} \sum_{k=0}^{Nn_t-1} \frac{c_1 Nn_t r_1^k \max\{c_2 Nn_t r_2^k, \theta_c^2\}}{(1+bc_1 r_1^k)^2} \\ &= b^2 c_1 c_2 \sum_{k=0}^{\bar{k}} \frac{(r_1 r_2)^k}{(1+bc_1 r_1^k)^2} + \frac{b^2 c_1 \theta_c^2}{Nn_t} \sum_{k=\bar{k}+1}^{Nn_t-1} \frac{r_1^k}{(1+bc_1 r_1^k)^2} \end{aligned}$$

The expression for the control energy can be upper bounded,

$$\begin{aligned} E_N(b) &\leq b^2 c_1 c_2 \sum_{k=0}^{\bar{k}} (r_1 r_2)^k + \frac{b^2 c_1 \theta_c^2}{Nn_t} \sum_{k=\bar{k}+1}^{Nn_t-1} r_1^k \\ &= b^2 c_1 c_2 \frac{1-(r_1 r_2)^{\bar{k}+1}}{1-r_1 r_2} + \frac{b^2 c_1 \theta_c^2}{Nn_t} \frac{r_1^{\bar{k}+1} - r_1^{Nn_t}}{1-r_1} \end{aligned}$$

Finally, in the large $N$ limit, we show that the control energy is upper bounded by a constant.

$$\lim_{N\to\infty} E_N(b) \leq \frac{b^2 c_1 c_2}{1 - r_1 r_2} \tag{B.4.9}$$

Moving to the total cost, applying assumptions 1 and 2 to the expression in Eq. (B.4.6) yields the expression,

$$\begin{aligned} J_N(b) &\approx \frac{b}{2(Nn_t + b)} \sum_{k=0}^{Nn_t-1} \frac{\max\{c_2 N n_t r_2^k, \theta_c^2\}}{1 + bc_1 r_1^k} \\ &= \frac{bc_2 Nn_t}{2(Nn_t + b)} \sum_{k=0}^{\bar{k}} \frac{r_2^k}{1 + bc_1 r_2^k} + \frac{b\theta_c^2}{2(Nn_t + b)} \sum_{k=\bar{k}+1}^{Nn_t-1} \frac{1}{1 + bc_1 r_1^k} \end{aligned}$$

This approximation for the total cost can be upper bounded using the same method for the previous two costs.

$$\begin{aligned} J_N(b) &\leq \frac{bc_2 Nn_t}{2(Nn_t + b)} \sum_{k=0}^{\bar{k}} r_2^k + \frac{b\theta_c^2}{2(Nn_t + b)} \sum_{k=\bar{k}+1}^{Nn_t-1} 1 \\ &= \frac{bc_2 Nn_t}{2(Nn_t + b)} \frac{1 - r_2^{\bar{k}+1}}{1 - r_2} + \frac{b\theta_c^2}{2(Nn_t + b)} (Nn_t - \bar{k} + 1) \end{aligned}$$

In the limit of large $N$, the total cost is shown to approach a constant as well,

$$\lim_{N\to\infty} J_N(b) \leq \frac{bc_2}{2(1 - r_2)} + \frac{b\theta_c^2}{2} \tag{B.4.10}$$

The bounds derived in Eqs. (B.4.8), (B.4.9), and (B.4.10) suggest that no matter how large $N$ grows, there exists control inputs of finite $\mathcal{L}_2$ norm which can drive each possible realization, $A_k$, to any final output $\boldsymbol{y}_f$ with arbitrarily small average deviation, $D_N(b)/N$. Examples and applications are shown in Sec. 3.4.

# Appendix C

# Derivations concerning Symmetries and Equitable Partitions

This appendix contains the details of proofs and derivation in Chapter 4 concering graph symmetries.

## C.1 Proof of Intra-Cluster Edges (Thm. 4.4.1)

*Proof.* The set of intra-cluster edges must be proven to (i) not contain a self-loop and (ii) if $(v_\ell^k, v_{\ell'}^k) \in \mathcal{E}_k$ then so is $(v_{\ell'}^k, v_\ell^k) \in \mathcal{E}_k$.

For a loop to exist there must be a $b_j$ in the set of integers such that $\ell = (\ell + b_j) \bmod p_k$, or that $b_j = qp_k$ for some integer $q$. This cannot occur though because $b_j > 0$ so $q \neq 0$ and $b_j \leq p_k/2$ so $q < 1$.

For the second part of the proof, let $(\ell, (\ell + b_j) \bmod p_k)$ be an edge. We know that $((\ell + b_j) \bmod p_k, ((\ell + b_j) \bmod p_k - b_j) \bmod p_k)$ is also an edge. Then, it can be shown simply that,

$$((\ell + b_j) \bmod p_k - b_j) \bmod p_k = (\ell + b_j - b_j) \bmod p_k = \ell$$

proving that if $(v_\ell^k, v_{\ell'}^k) \in \mathcal{E}_k$ implies $(v_{\ell'}^k, v_\ell^k) \in \mathcal{E}_k$. □

## C.2 Proof of Inter-Cluster Edges (Thm. 4.4.2)

*Proof.* To prove that the set of edges in either Eq. (4.4.7) or Eq. (4.4.8) satisfies the requirements, we must show for each edge $(v_\ell^j, v_{\ell'}^k) \in \mathcal{E}_{j,k}$, then so is $(v_{\ell'}^k, v_\ell^j) \in \mathcal{E}_{j,k}$. Start with the edge $(v_\ell^j, v_{\ell'}^k)$ where

$$\ell' = \left(\ell + r_1 c + \sum_{j=1}^{r_2} b_j\right) \bmod p_k, \quad 0 \le r_1 \le d_k - 1, \quad 1 \le r_2 \le m$$

and we show that there exists unique integers $0 \le r_3 \le d_j - 1$ and $1 \le r_4 \le m$ such that for the edge $(v_{\ell'}^k, v_{\ell''}^j)$,

$$\begin{aligned}
\ell'' = \ell &= \left(\ell' + r_3 c + \sum_{a=1}^{r_4} b_{m-a+1}\right) \bmod p_j \\
&= \left(\left(\ell + r_1 c + \sum_{a=1}^{r_2} b_a\right) \bmod p_k + r_3 c + \sum_{a=1}^{r_4} b_{m-a+1}\right) \bmod p_j \\
&= \left(\ell + r_1 c + \sum_{a=1}^{r_2} b_a - q p_k + r_3 c + \sum_{a=1}^{r_4} b_{m-a+1}\right) \bmod p_j
\end{aligned}$$

where $q$ is chosen such that,

$$0 \le \ell + r_1 c + \sum_{a=1}^{r_2} b_a - q n_k < p_k$$

There are two possibilities we differentiate between, (i) if $r_2 < m$ and (ii) if $r_2 = m$.

**Case 1:** Let $r_4 = m - r_2$ so that the summation,

$$\sum_{a=1}^{r_2} b_a + \sum_{a=1}^{m-r_2} b_{m-a+1} = \sum_{a=1}^{r_2} b_a + \sum_{a=r_2+1}^{m} b_a = \sum_{a=1}^{m} b_a = c$$

The expression for $\ell'' = \ell$ can now be simplified to,

$$\ell = (\ell + (r_1 + r_3 + 1)c - q p_k) \bmod p_j$$

For this expression to hold true, we must find the unique $r_3$ such that,

$$(r_1 + r_3 + 1)c - qp_k = tp_j \quad \Rightarrow \quad r_3 = td_j + qd_k - r_1 - 1 \tag{C.2.1}$$

From the bounds on $r_3$,

$$0 \leq td_j + qd_k - r_1 - 1 \leq d_j - 1$$
$$r_1 + 1 - qd_k \leq td_j \leq d_j + r_1 - qd_k$$

What remains is to show there exists a single integer $t$ that satisfies the above inequality, i.e., the upper bound and lower bound spans exactly $d_j$ integers.

$$(d_j + r_1 - qd_k) - (r_1 + 1 - qd_k) + 1 = d_j$$

Thus, the interval only permits a single integer $t$ which satisfies the inequality assigned to $r_3$.

**Case 2:** Now, let $r_2 = m$ and set $r_4 = m$ so that,

$$\sum_{a=1}^{m} b_a + \sum_{a=1}^{m} b_{m-a+1} = 2c$$

The expression for $\ell'' = \ell$ can be simplified to,

$$\ell = (\ell + (r_1 + r_3 + 2)c - qp_k) \bmod p_j$$

As before, we show there is a single integer $t$,

$$(r_1 + r_3 + 2)c - qp_k = tn_j \quad \Rightarrow \quad r_3 = td_j + qd_k - r_1 - 2$$

that satisfies the bounds,

$$0 \leq td_j + qd_k - r_1 - 2 \leq d_j - 1$$
$$r_1 + 2 - qd_k \leq td_j \leq d_j + r_1 + 1 - qd_k$$

Once again, we show that the number of integers covered by the lower and upper bounds are exactly equal to $d_j$,

$$d_j + 1 - qd_k - (r_1 + 2 - qd_k) + 1 = d_j$$

so that there can only be a single value of $t$ that satisfies the inequality.

In summary, given a node $v_\ell^j$, the $md_k = \frac{W_{j,k}c}{p_k}\frac{p_k}{c} = W_{j,k}$ pairs of integers $(r_1, r_2)$ such that $0 \leq r_1 \leq d_k - 1$ and $1 \leq r_2 \leq m$ each yield a unique neighboring node $v_{\ell'}^k$. Each pair $(r_1, r_2)$ corresponds to a unique pair $(r_3, r_4)$ so that if $v_{\ell'}^k$ is a neighbor of $v_\ell^j$, then so is $v_\ell^j$ a neighbor of $v_{\ell'}^k$. □

# Appendix D

# Derivations for Results in Control of Lattice Graphs

This appendix contains the derivations of the results presented in Chapter 5. Derivations are presented in the same order as they appear in Chapter 5. The derivation of the time-varying controllability Gramian of the bidirectional path graph is presented in Sec. D.1.1 and its steady state value in Sec. D.1.2. The recurrence relation for the diagonal values of the steady state controllability Gramian of the bidirectional path graph is derived in Sec. D.1.3. The controllability Gramian of the unidirectional path graph is derived in Sec. D.2, the bidirectional ring graph is derived in Sec. D.3, and the unidirectional ring graph is proven in Sec. D.4. The controllability Gramian of the unidirectional balloon graph is derived in Sec. D.5. In Sec. D.6 the controllability Gramian of the hypercubic lattice is derived in Sec. D.6 and the controllability Gramian for general lattices is derived in Sec. D.7.

## D.1 Derivations for the Bidirectional Path Graph

### D.1.1 Time-Varying Gramian

In this section, the time-varying elements of the controllability Gramian are derived for the bidirectional path graph as described in Sec. 5.3.1. The set of driver nodes $\mathcal{D}$ consists of $n_d$ nodes placed along the path. The controllability Gramian elements satisfy the linear differential equation,

$$\begin{aligned}\dot{W}_{j,k}(t) &= -2pW_{j,k}(t) + sW_{j+1,k}(t) + sW_{j-1,k}(t) \\ &\quad + sW_{j,k+1}(t) + sW_{j,k-1}(t) + \sum_{v_a \in \mathcal{D}} \delta_{j,a}\delta_{k,a}, \quad W_{j,k}(0) = 0, \quad \forall j,k \in \mathbb{Z}\end{aligned} \tag{D.1.1}$$

To solve this system of equations in Eq. (D.1.1), first a Laplace transform is taken where $\tilde{W}_{j,k}(\epsilon) = \mathcal{L}(W_{j,k}(t))$ such that,

$$\tilde{W}_{j,k}(\epsilon) - \rho\left(\tilde{W}_{j+1,k}(\epsilon) + \tilde{W}_{j-1,k}(\epsilon) + \tilde{W}_{j,k+1}(\epsilon) + \tilde{W}_{j,k-1}(\epsilon)\right) = \frac{1}{\epsilon(\epsilon+2p)} \sum_{v_a \in \mathcal{D}} \delta_{j,a}\delta_{k,a}, \quad \forall j,k \in \mathbb{Z} \tag{D.1.2}$$

where $\rho = \frac{s}{\epsilon+2p}$. To decouple the algebraic equations in Eq. (D.1.2), the discrete time Fourier transform (DTFT) is used, defined as,

$$\hat{\tilde{W}}_{\hat{j},\hat{k}}(\epsilon) = \sum_{j,k\in\mathbb{Z}} e^{-ij\hat{j}} e^{-ik\hat{k}} \tilde{W}_{j,k}(\epsilon) \tag{D.1.3}$$

with inverse,

$$\tilde{W}_{j,k}(\epsilon) = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} e^{ij\hat{j}} e^{ik\hat{k}} \hat{\tilde{W}}_{\hat{j},\hat{k}}(\epsilon) d\hat{j} d\hat{k} \tag{D.1.4}$$

From the definition of the transform in Eq. (D.1.3), it is simple to show that, for any pairs of integers $\ell$ and $p$, the transformed value of a Gramian element $\tilde{W}_{j+\ell,k+p}(\epsilon)$ can be found to be,

$$\sum_{j,k\in\mathbb{Z}} \tilde{W}_{j+\ell,k+p} e^{-ij\hat{j}} e^{-ik\hat{k}} = e^{i\ell\hat{j}} e^{ip\hat{k}} \sum_{j,k\in\mathbb{Z}} \tilde{W}_{j+\ell,k+p}(\epsilon) e^{-i(j+\ell)\hat{j}} e^{-i(k+p)\hat{k}} = e^{i\ell\hat{j}} e^{ip\hat{k}} \hat{\tilde{W}}_{\hat{j},\hat{k}}(\epsilon) \tag{D.1.5}$$

Multiplying Eq. (D.1.2) by $e^{-ij\hat{j}}e^{-ik\hat{k}}$, summing over all pairs of integers $j,k \in \mathbb{Z}$, and using the transform in Eq. (D.1.3) with the definition of transformed elements in Eq.

(D.1.5),

$$\begin{aligned}\sum_{j,k\in\mathbb{Z}} \tilde{W}_{j,k}(\epsilon)e^{-ij\hat{j}}e^{-ik\hat{k}} - \rho\sum_{j,k\in\mathbb{Z}}\left(\tilde{W}_{j+1,k}(\epsilon)+\tilde{W}_{j-1,k}(\epsilon)+\tilde{W}_{j,k+1}(\epsilon)+\tilde{W}_{j,k-1}(\epsilon)\right)e^{-ij\hat{j}}e^{-ik\hat{k}}&\\ = \frac{1}{\epsilon(\epsilon+2p)}\sum_{j,k\in\mathbb{Z}}\sum_{v_a\in\mathcal{D}}\delta_{j,a}\delta_{k,a}e^{-ij\hat{j}}e^{-ik\hat{k}}&\\ \left(1-\rho e^{i\hat{j}}-\rho e^{-i\hat{j}}-\rho e^{i\hat{k}}-\rho e^{-i\hat{k}}\right)\hat{\tilde{W}}_{\hat{j},\hat{k}}(\epsilon) = \frac{1}{\epsilon(\epsilon+2p)}\sum_{v_a\in\mathcal{D}}e^{-ia\hat{j}}e^{-ia\hat{k}}&\\ (1-2\rho\cos\hat{j}-2\rho\cos\hat{k})\hat{\tilde{W}}_{\hat{j},\hat{k}}(\epsilon) = \frac{1}{\epsilon(\epsilon+2p)}\sum_{v_a\in\mathcal{D}}e^{-ia\hat{j}}e^{-ia\hat{k}}&\end{aligned} \tag{D.1.6}$$

Rearranging Eq. (D.1.6) for the transformed controllability Gramian elements,

$$\hat{\tilde{W}}_{j,k}(\epsilon) = \frac{1}{\epsilon(\epsilon+2p)}\frac{\sum_{v_a\in\mathcal{D}}e^{-ia\hat{j}}e^{-ia\hat{k}}}{1-2\rho\cos\hat{j}-2\rho\cos\hat{k}} \tag{D.1.7}$$

Applying the inverse DTFT in Eq. (D.1.4) to the transformed Gramian elements in Eq. (D.1.7) yields the bidirectional path graph controllability Gramian elements in the frequency domain,

$$\begin{aligned}\tilde{W}_{j,k}(\epsilon) &= \frac{1}{4\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}e^{ij\hat{j}}e^{ik\hat{k}}\left[\frac{1}{\epsilon(\epsilon+2p)}\frac{\sum_{v_a\in\mathcal{D}}e^{-ia\hat{j}}e^{-ia\hat{k}}}{1-2\rho\cos\hat{j}-2\rho\cos\hat{k}}\right]d\hat{j}d\hat{k}\\ &= \frac{1}{\epsilon}\sum_{v_a\in\mathcal{D}}\frac{1}{4\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\frac{e^{-i(a-j)\hat{j}}e^{-i(a-k)\hat{k}}}{\epsilon+2p-2s\cos\hat{j}-2s\cos\hat{k}}d\hat{j}d\hat{k}\end{aligned} \tag{D.1.8}$$

The controllability Gramian elements in the time domain are now determined by taking the inverse Laplace transform of the form in Eq. (D.1.8),

$$\begin{aligned}W_{j,k}(t) &= \mathcal{L}^{-1}\left(\tilde{W}_{j,k}(\epsilon)\right)\\ &= \sum_{v_a\in\mathcal{D}}\int_0^t e^{-2p\tau}\left[\frac{1}{2\pi}\int_{-\pi}^{\pi}e^{2s\tau\cos\hat{j}}e^{-i(a-j)\hat{j}}d\hat{j}\right]\left[\frac{1}{2\pi}\int_{-\pi}^{\pi}e^{2s\tau\cos\hat{k}}e^{-i(a-k)\hat{k}}d\hat{k}\right]d\tau\end{aligned} \tag{D.1.9}$$

Examining the integrals in Eq. (D.1.9), one can rearrange the exponential with the imaginary exponent to show that,

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} e^{2s\tau\cos\hat{j}}(\cos((a-j)\hat{j}) - i\sin((a-j)\hat{j}))d\hat{j} = \frac{1}{\pi}\int_{0}^{\pi} e^{2s\tau\cos\hat{j}}\cos((a-j)\hat{j})d\hat{j} \tag{D.1.10}$$

where the right hand side of Eq. (D.1.10) is the integral definition of the modified Bessel function of the first kind (MBFFK) of integer order (see identity 9.6.19 in [235]),

$$I_n(z) = \frac{1}{\pi}\int_0^{\pi} e^{z\cos\theta}\cos n\theta d\theta \tag{D.1.11}$$

Some important properties of the MBFFK of integer order are,

1. $I_n(z) > 0$ for $z > 0$,
2. $I_n(z) = 0$ for $z = 0$ and $n \neq 0$, i.e., $I_n(0) = \delta_{n,0}$,
3. $\frac{d}{dz}I_n(z) > 0$,
4. and they are symmetric about $n = 0$, i.e., $I_n(z) = I_{-n}(z)$.

Using the result of Eq. (D.1.10) and the definition of MBFFK in Eq. (D.1.11) in Eq. (D.1.9), the elements of the controllability Gramian of the bidirectional path graph can be succinctly expressed as the integral,

$$W_{j,k}(t) = \sum_{v_a\in\mathcal{D}}\int_0^t e^{-2p\tau}I_{a-j}(2s\tau)I_{a-k}(2s\tau)d\tau \tag{D.1.12}$$

The first result of Eq. (D.1.12) is that the contribution of each driver node is independent, that is, we can evaluate Eq. (D.1.12) for each driver node $v_a \in \mathcal{D}$ separately. With this, we can simplify the evaluation of Eq. (D.1.12) by evaluating it for $\mathcal{D} = \{v_0\}$ only. Then, for each original driver node $v_a \in \mathcal{D}$, we shift the indices of $W_{j,k}$ to $j + a \to j$ and $k + a \to k$. With this in mind, moving forwards, we focus only on the case $\mathcal{D} = \{v_0\}$.

For finite $t$, the expression in Eq. (D.1.12) cannot be simplified into an expression containing elementary functions, but it can be reformulated in terms of an infinite summation which will be useful in the derivation of the steady state controllability Gramian in the next section. To do this, the modified Bessel functions are expanded as a power series (see 9.6.10 in [235]),

$$I_n(z) = \left(\frac{z}{2}\right)^n \sum_{\ell=0}^{\infty} \frac{\left(\frac{z}{2}\right)^{2\ell}}{\ell!(n+\ell)!} \tag{D.1.13}$$

Using the Cauchy product to evaluate Eq. (D.1.12) using the power series expansion in Eq. (D.1.13),

$$\begin{aligned} W_{j,k}(\tau) &= \int_0^t e^{-2p\tau} \left( (s\tau)^j \sum_{\ell=0}^{\infty} \frac{(s\tau)^{2\ell}}{\ell!(j+\ell)!} \right) \left( (s\tau)^k \sum_{\ell=0}^{\infty} \frac{(s\tau)^{2k}}{\ell!(k+\ell)!} \right) dt \\ &= \sum_{\ell=0}^{\infty} \int_0^t e^{-2p\tau} (s\tau)^{j+k+2\ell} \sum_{a=0}^{\ell} \frac{1}{a!(\ell-a)!(j+a)!(k+\ell-a)!} d\tau \\ &= \sum_{\ell=0}^{\infty} c_{j,k}^{(\ell)} s^{j+k+2\ell} \int_0^t e^{-2p\tau} \tau^{j+k+2\ell} d\tau \end{aligned} \tag{D.1.14}$$

where the coefficients $c_{j,k}^{(\ell)}$ can be evaluated using identity 3.4 from [249],

$$\begin{aligned} c_{j,k}^{(\ell)} &= \sum_{a=0}^{\ell} \frac{1}{a!(\ell-a)!(j+a)!(k+\ell-a)!} \\ &= \frac{1}{\ell!(j+k+\ell)!} \sum_{a=0}^{\ell} \frac{\ell!}{a!(\ell-a)!} \frac{(j+k+\ell)!}{(j+a)!(k+\ell-a)!} \\ &= \frac{1}{(j+k+2\ell)!} \frac{(j+k+2\ell)!}{\ell!(j+k+\ell)!} \sum_{a=0}^{\ell} \binom{\ell}{a} \binom{j+k+\ell}{k+\ell-a} \\ &= \frac{1}{(j+k+2\ell)!} \binom{j+k+2\ell}{\ell} \binom{j+k+2\ell}{k+\ell} \end{aligned} \tag{D.1.15}$$

The integral in Eq. (D.1.14) can be solved using identity 3.351.1 in [243].

$$\begin{aligned} \int_0^t e^{-2p\tau} \tau^{j+k+2\ell} d\tau &= \frac{(j+k+2\ell)!}{(2p)^{j+k+2\ell+1}} - e^{-2pt} \sum_{a=0}^{j+k+2\ell} \frac{(j+k+2\ell)!}{a!} \frac{t^a}{(2p)^{j+k+2\ell-a+1}} \\ &= \frac{(j+k+2\ell)!}{(2p)^{j+k+2\ell+1}} \left[ 1 - e^{-2pt} \sum_{a=0}^{j+k+2\ell} \frac{(2pt)^a}{a!} \right] \end{aligned} \tag{D.1.16}$$

Plugging the results from Eq. (D.1.15) and Eq. (D.1.16) into the expression in Eq. (D.1.14) yields the result,

$$W_{j,k}(t) = \frac{1}{2p}\sum_{\ell=0}^{\infty}\left(\frac{s}{2p}\right)^{j+k+2\ell}\binom{j+k+2\ell}{\ell}\binom{j+k+2\ell}{k+\ell}\left[1 - e^{-2pt}\sum_{a=0}^{j+k+2\ell}\frac{(2pt)^a}{a!}\right] \tag{D.1.17}$$

The time-varying portion of Eq. (D.1.17) decays to zero as $t \to \infty$, which can be shown by including the term $\left(\frac{s}{2p}\right)^{j+k+2\ell}$ in the fraction and noting that the exponential $e^{-2pt}$ decays to zero faster than the leading term in summation over the index $a$. On the other hand, for the summation over $\ell$ to converge, $\frac{s}{2p} < \frac{1}{4}$, which is equivalent to the requirements for the bidirectional path graph to be Hurwitz.

### D.1.2 Steady State Gramian

In this section, the steady state controllability Gramian of the bidirectional path graph is derived in two different forms as shown in Eqs. (5.3.6) and (5.3.8) in Sec. 5.3.1 of the main text. The first form is simply an application of the final value theorem applied to Eq. (D.1.8).

$$W_{j,k} = \sum_{v_a \in \mathcal{D}} \frac{1}{2\pi^2}\int_0^{\pi}\int_0^{\pi}\frac{\cos((a-j)\hat{j})\cos((a-k)\hat{k})}{p - s\cos\hat{j} - s\cos\hat{k}}d\hat{j}d\hat{k} \tag{D.1.18}$$

This form of the steady state controllability Gramian has appeared previously in other contexts as the lattice Green's function of a square lattice. Alternatively, a second form can be derived using Eq. (D.1.17) and the definition of the hypergeometric functions.

$$\begin{aligned}
W_{j,k} &= \lim_{t\to\infty} W_{j,k}(t) \\
&= \frac{1}{2p}\sum_{\ell=0}^{\infty}\left(\frac{s}{2p}\right)^{j+k+2\ell}\binom{j+k+2\ell}{\ell}\binom{j+k+2\ell}{k+\ell} \\
&= \frac{1}{2p}\left(\frac{s}{2p}\right)^{j+k}\binom{j+k}{k} \\
&\times {}_4F_3\left[\begin{matrix}\frac{1}{2}(j+k+1), & \frac{1}{2}(j+k+1), & \frac{1}{2}(j+k+2), & \frac{1}{2}(j+k+2) \\ & j+1, & k+1, & j+k+1\end{matrix};\left(\frac{2s}{p}\right)^2\right]
\end{aligned} \tag{D.1.19}$$

where ${}_4F_3$ is the generalized hypergeometric function (see chapter 16 in [242]),

$${}_4F_3\left[\begin{matrix}a_1, & a_2, & a_3, & a_4 \\ b_1, & b_2, & b_3 & \end{matrix};z\right] = \sum_{\ell=0}^{\infty}\frac{(a_1)_\ell(a_2)_\ell(a_3)_\ell(a_4)_\ell}{(b_1)_\ell(b_2)_\ell(b_3)_\ell}\frac{z^\ell}{\ell!}$$

and $(a)_\ell = \prod_{k=0}^{\ell-1}(a+k)$ is the rising factorial. Specializing the result in Eq. (D.1.19) to a diagonal element, the generalized hypergeometric function reduces to the more familiar Gauss hypergeometric function (see Chapter 15 in [242]),

$$W_{k,k} = \frac{1}{2p}\left(\frac{s}{2p}\right)^{2k}\binom{2k}{k}{}_2F_1\left[\begin{matrix}k+\frac{1}{2}, & k+\frac{1}{2} \\ 2k+1 & \end{matrix};\left(\frac{2s}{p}\right)^2\right] \tag{D.1.20}$$

There exist computational libraries to compute hypergeometric functions accurately and so the expressions in Eqs. (D.1.19) and (D.1.20) can be more reliable than numerically evaluating the infinite integral in Eq. (D.1.12) derived in the previous section.

### D.1.3 Diagonal Elements

In this section, the three term recurrence relation for the diagonal elements of the controllability Gramian of the bidirectional path graph shown in Eq. (5.3.11) in Sec. 5.3.1 of the main text is derived. We return to the expression derived previously in Eq. (D.1.12) for the time-varying controllability Gramian elements of the bidirectional path graph and we take the $t \to \infty$ limit.

$$W_{j,k} = \lim_{t\to\infty} W_{j,k}(t) = \int_0^\infty e^{-2pt} I_j(2st) I_k(2st) dt \tag{D.1.21}$$

The integral in Eq. (D.1.21) is expressed in terms of Bessel functions using the connection formula (see 9.6.3 in [235])

$$I_k(2st) = (-i)^k J_k(2ist)$$

so that the diagonal elements of the controllability Gramian in Eq. (D.1.21) become,

$$W_{k,k} = (-1)^k \int_0^\infty e^{-2pt} J_k^2(2ist) dt \tag{D.1.22}$$

The integral in Eq. (D.1.22) appears as identity 6.612.3 in [243] where it can be expressed in terms of the Legendre function of the second kind.

$$W_{k,k} = \frac{(-1)^k}{2i\pi s} Q_{k-\frac{1}{2}}\left(1 - \frac{p^2}{2s^2}\right) \tag{D.1.23}$$

The Legendre functions of the second kind satisfy the three term recurrence relation as stated in identity 8.5.3 in [235],

$$(\nu+1)Q_{\nu+1}(z) = (2\nu+1)zQ_\nu(z) - \nu Q_{\nu-1}(z) \tag{D.1.24}$$

The recurrence in Eq. (D.1.24) is specialized to the expression in Eq. (D.1.23) by shifting the index $\nu + 1 \to k - \frac{1}{2}$, defining $-z = \frac{p^2}{2s^2} - 1$, and multiplying through by $\frac{1}{2i\pi s}$,

$$\frac{2k-1}{2i\pi s} Q_{k-\frac{1}{2}}(-z) + \frac{4k-4}{2i\pi s} zQ_{k-\frac{3}{2}}(-z) + \frac{2k-3}{2i\pi s} Q_{k-\frac{5}{2}}(-z) = 0, \quad k \geq 2 \tag{D.1.25}$$

Comparing the form $W_{k,k}$ in Eq. (D.1.23) to the expressions in each term in Eq. (D.1.25) yields the recurrence relation,

$$W_{k,k} = \frac{4k-4}{2k-1}\left(\frac{p^2}{2s^2} - 1\right) W_{k-1,k-1} - \frac{2k-3}{2k-1} W_{k-2,k-2}, \quad k \geq 2 \tag{D.1.26}$$

The initial values of the recurrence relation, for $k = 0$ and $k = 1$, are found using integral identity 6.612.4 in [243]

$$\begin{aligned} W_{0,0} &= \int_0^\infty e^{-2pt} J_0^2(2ist)dt \\ &= \frac{2}{\pi\sqrt{4p^2 - 16s^2}} K\left(\frac{4is}{\sqrt{4p^2-16s^2}}\right) \\ &= \frac{1}{\pi p} K\left(\frac{2s}{p}\right) \end{aligned} \tag{D.1.27}$$

and integral identity 6.612.5 in [243]

$$\begin{aligned} W_{1,1} &= -\int_0^\infty e^{-2pt} J_1^2(2ist)dt \\ &= \frac{(2p^2 - 4s^2)K\left(\frac{2is}{\sqrt{p^2-4^2}}\right)}{4\pi s^2\sqrt{p^2-4s^2}} - \frac{(p^2-4s^2)E\left(\frac{2is}{\sqrt{p^2-4s^2}}\right)}{4\pi s^2\sqrt{p^2-4s^2}} \\ &= \left(\frac{p}{2\pi s^2} - \frac{1}{\pi p}\right) K\left(\frac{2s}{p}\right) - \frac{p}{2\pi s^2} E\left(\frac{2s}{p}\right) \end{aligned} \tag{D.1.28}$$

where $K(k)$ and $E(k)$ are the complete elliptic integrals of the first and second kind [235], respectively.

$$K(k) = \int_0^{\pi/2} \frac{1}{\sqrt{1-k^2\sin^2\theta}} d\theta E(k) \quad = \int_0^{\pi/2} \sqrt{1-k^2\sin^2\theta} d\theta$$

Also used were the imaginary argument identities for the complete elliptic integrals (see 17.4.17 and 17.4.18 in [235]).

$$K(ik) = \frac{1}{\sqrt{1+k^2}} K\left(\frac{k}{\sqrt{1+k^2}}\right), \quad E(ik) = \sqrt{1+k^2} E\left(\frac{k}{\sqrt{1+k^2}}\right)$$

The expressions in Eqs. (D.1.27) and (D.1.28) were derived previously using the fact $W_{k,k}$ can be expressed as a hypergeometric function as discussed in Sec. D.1.2. The scaling behavior of the diagonal elements are found by defining the decay rate $R_k = \frac{W_{k,k}}{W_{k-1,k-1}}$ and dividing Eq. (D.1.26) through by $W_{k-1,k-1}$,

$$R_k = \frac{4k-4}{2k-1}\left(\frac{p^2}{2s^2}-1\right) - \frac{2k-3}{2k-1}\frac{1}{R_{k-1}}, \quad k \geq 2 \tag{D.1.29}$$

In the asymptotic limit of large $k$, we assume $\lim\limits_{k\to\infty} = R$, and rewrite Eq. (D.1.29) in this limit,

$$R^2 = 2\left(\frac{p^2}{2s^2}-1\right)R - 1 \tag{D.1.30}$$

Letting $\alpha = \frac{p^2}{2s^2} - 1$ where, as $p > 2s$, $\alpha > 1$, and noting that $0 < R < 1$, the quadratic equation in Eq. (D.1.30) can be solved for the meaningful value of $R$,

$$\begin{aligned} R &= \alpha - \sqrt{\alpha^2 - 1} \\ &= \frac{s^2}{p^2}\left[\frac{p^4}{2s^4} - \frac{p^2}{s^2} - \sqrt{\frac{p^8}{4s^8} - \frac{p^6}{s^6}}\right] \end{aligned} \tag{D.1.31}$$

With the decay rate in Eq. (D.1.31), we can approximate the values $W_{k,k}$ as,

$$W_{k,k} \sim (\alpha - \sqrt{\alpha^2-1})^k \tag{D.1.32}$$

The asymptotic control energy for the single driver single target problem can be derived by inverting Eq. (D.1.32) and taking its logarithm.

$$\log E_k \sim k\left[2\log p - 2\log s - \log\left(\frac{p^4}{2s^4} - \frac{p^2}{s^2} - \sqrt{\frac{p^8}{4s^8} - \frac{p^6}{s^6}}\right)\right] \tag{D.1.33}$$

To understand the third term in Eq. (D.1.33) better, define $x = \frac{p^2}{s^2}$ and perform the following simplification,

$$\begin{aligned}\frac{x^2-2x}{2} - \sqrt{\frac{x^4-4x^3}{4}} &= \frac{\left(\frac{x^2-2x}{2} - \sqrt{\frac{x^4-4x^3}{4}}\right)\left(\frac{x^2-2x}{2} + \sqrt{\frac{x^4-4x^3}{4}}\right)}{\left(\frac{x^2-2x}{2} + \sqrt{\frac{x^4-4x^3}{4}}\right)} \\ &= \frac{1}{2}\frac{(x^2-2x)^2-(x^4-4x^3)}{x^2-2x+\sqrt{x^4-4x^3}} \\ &= 2\left[\frac{1}{1-\frac{2}{x}+\sqrt{1-\frac{4}{x}}}\right]\end{aligned} \tag{D.1.34}$$

Applying Eq. (D.1.34) to the asymptotic energy in Eq. (D.1.33) yields,

$$\log E_k \sim k\left[2\log p - 2\log s - \log\left(\frac{2}{1-\frac{2s^2}{p^2}+\sqrt{1-\frac{4s^2}{p^2}}}\right)\right] \tag{D.1.35}$$

For $p \gg 2s$ and $k$ large enough, the right hand side is approximately $2k(\log p - \log s)$ which can be shown by taking the limit of the expression in the third logarithm,

$$\lim_{\frac{s^2}{p^2}\to 0} \frac{2}{1-\frac{2s^2}{p^2}+\sqrt{1-\frac{4s^2}{p^2}}} = 1$$

The exponential growth rate of the control energy in this asymptotic limit can be found by differentiating Eq. (D.1.33),

$$\frac{d}{dk}\log E_k \sim 2\log p - 2\log s \tag{D.1.36}$$

The scaling behavior in Eq. (D.1.36) agrees with the numerical results witnessed for the finite path graph described at the beginning of Chapter 5 where $\log E \sim 2k\log p$ and, as $s = 1$ was chosen for the edge weights, the constant term was zero.

## D.2 Derivations for the Unidirectional Path Graph

In this section, the results in Sec. 5.3.2 are derived using a generating function. The differential Lyapunov equation for the controllability Gramian of the unidirectional path graph, defined on the nodes $\mathcal{V} = \{v_k | k =\geq 0\}$, with edges $(v_k, v_{k+1})$, $\forall k \geq 0$, and a single driver node $\mathcal{D} = \{v_0\}$, can be expressed as a time-varying recurrence equation.

$$\begin{aligned}
&\dot{W}_{j,k}(t) = -2pW_{j,k}(t) + sW_{j-1,k}(t) + sW_{j,k-1}(t), \quad W_{j,k}(0) = 0, \quad j,k \geq 1\\
&\dot{W}_{j,0}(t) = -2pW_{j,0}(t) + sW_{j-1,0}(t), \quad W_{j,0}(0) = 0, \quad j \geq 1\\
&\dot{W}_{0,k}(t) = -2pW_{0,k}(t) + sW_{0,k-1}(t), \quad W_{0,k}(t) = 0, \quad k \geq 1\\
&\dot{W}_{0,0}(t) = -2pW_{0,0}(t) + 1, \quad W_{0,0}(0) = 0
\end{aligned} \tag{D.2.1}$$

First, applying the Laplace transform to the system of equations in Eq. (D.2.1), yields the system of algebraic equations,

$$\begin{aligned}
\tilde{W}_{j,k}(\epsilon) - \rho\tilde{W}_{j-1,k}(\epsilon) - \rho\tilde{W}_{j,k-1}(\epsilon) &= 0, \quad j,k \geq 1\\
\tilde{W}_{j,0}(\epsilon) - \rho\tilde{W}_{j-1,0}(\epsilon) &= 0, \quad j \geq 1\\
\tilde{W}_{0,k}(\epsilon) - \rho\tilde{W}_{0,k-1}(\epsilon) &= 0, \quad k \geq 1\\
\tilde{W}_{0,0}(\epsilon) &= \alpha
\end{aligned} \tag{D.2.2}$$

where the parameters $\alpha = \frac{1}{\epsilon(\epsilon+2p)}$ and $\rho = \frac{s}{\epsilon+2p}$. The boundary elements are symmetric, that is, $\tilde{W}_{j,0}(\epsilon) = \tilde{W}_{0,j}(\epsilon)$ for all $j \geq 0$, and so we need only to solve for one of them. The boundary elements are found using a generating function.

$$\tilde{W}(x,0) = \sum_{j\geq 0} \tilde{W}_{j,0}(\epsilon)x^j \tag{D.2.3}$$

Multiplying the second equation in Eq. (D.2.2) by $x^j$ and summing over $j \geq 0$,

$$\begin{aligned}
0 &= \sum_{j\geq 0} \tilde{W}_{j+1,0}(\epsilon)x^j - \rho \sum_{j,0}(\epsilon)x^j \\
&= \frac{1}{x}\sum_{j\geq 1} \tilde{W}_{j,0}(\epsilon)x^j - \rho\tilde{W}(x,0) \\
&= \frac{1}{x}\left(\tilde{W}(x,0) - \tilde{W}_{0,0}(\epsilon)\right) - \rho\tilde{W}(x,0)
\end{aligned} \tag{D.2.4}$$

Rearranging Eq. (D.2.4) to solve for the boundary generating function and using the binomial theorem to put the expression in the form of Eq. (D.2.3),

$$\begin{aligned}
\tilde{W}(x,0) &= \frac{1}{1-\rho x}\tilde{W}_{0,0}(\epsilon) \\
&= \tilde{W}_{0,0}(\epsilon)\sum_{j\geq 0} \rho^j x^j
\end{aligned} \tag{D.2.5}$$

Comparing the forms of Eq. (D.2.3) and Eq. (D.2.5), we see the boundary elements are,

$$\tilde{W}_{j,0}(\epsilon) = \frac{1}{\epsilon(\epsilon+2p)}\left(\frac{s}{\epsilon+2p}\right)^j \tag{D.2.6}$$

From symmetry in the controllability Gramian, we also know that $\tilde{W}(0,y) = \frac{1}{1-\rho y}\tilde{W}_{0,0}(\epsilon)$ and $\tilde{W}_{0,k}(\epsilon) = \frac{1}{\epsilon(\epsilon+2p)}\left(\frac{s}{\epsilon+2p}\right)^k$. Turning to the interior terms, define the two term generating function,

$$\tilde{W}(x,y) = \sum_{j,k\geq 0} \tilde{W}_{j,k}(\epsilon)x^j y^k \tag{D.2.7}$$

Multiplying the first line of Eq. (D.2.2) by $x^j y^k$ and summing over all pairs $j, k \geq 0$ (using the boundary element form in Eq. (D.2.6)),

$$\begin{aligned}
0 &= \sum_{j,k\geq 0} \tilde{W}_{j+1,k+1}(\epsilon) x^j y^k - \rho \sum_{j,k\geq 0} \tilde{W}_{j,k+1}(\epsilon) x^j y^k - \rho \sum_{j,k\geq 0} \tilde{W}_{j+1,k}(\epsilon) x^j y^k \\
&= \frac{1}{xy} \sum_{j,k\geq 0} \tilde{W}_{j+1,k+1}(\epsilon) x^{j+1} y^{k+1} - \frac{\rho}{y} \sum_{j,k\geq 0} \tilde{W}_{j,k+1} x^j y^{k+1} - \frac{\rho}{x} \sum_{j,k\geq 0} \tilde{W}_{j+1,k}(\epsilon) x^{j+1} y^k \\
&= \frac{1}{xy} \left[ \sum_{j,k\geq 0} \tilde{W}_{j,k}(\epsilon) x^j y^k - \sum_{j\geq 0} \tilde{W}_{j,0} x^j - \sum_{k\geq 0} \tilde{W}_{0,k} y^k + \tilde{W}_{0,0}(\epsilon) \right] \\
&- \frac{\rho}{y} \left[ \sum_{j,k\geq 0} \tilde{W}_{j,k}(\epsilon) x^j y^k - \sum_{j\geq 0} \tilde{W}_{j,0}(\epsilon) x^j \right] - \frac{\rho}{x} \left[ \sum_{j,k\geq 0} \tilde{W}_{j,k}(\epsilon) x^j y^k - \sum_{k\geq 0} \tilde{W}_{0,k}(\epsilon) y^k \right] \\
&= \tilde{W}(x,y) - \tilde{W}(x,0) - \tilde{W}(0,y) + \tilde{W}_{0,0}(\epsilon) \\
&- \rho x(\tilde{W}(x,y) + \rho x \tilde{W}(x,0)) - \rho y(\tilde{W}(x,y) + \rho y \tilde{W}(0,y))
\end{aligned} \tag{D.2.8}$$

Rearranging Eq. (D.2.8) for the generating function $\tilde{W}(x,y)$,

$$\begin{aligned}
\tilde{W}(x,y) &= \frac{1}{1 - \rho x - \rho y} \left[ (1 - \rho x) \tilde{W}(x,0) + (1 - \rho y) \tilde{W}(0,y) - \tilde{W}_{0,0}(\epsilon) \right] \\
&= \frac{\tilde{W}_{0,0}(\epsilon)}{1 - \rho x - \rho y} \\
&= \tilde{W}_{0,0}(\epsilon) \sum_{\ell \geq 0} \left( \frac{s}{\epsilon + 2p} \right)^{\ell} (x + y)^{\ell} \\
&= \tilde{W}_{0,0}(\epsilon) \sum_{\ell \geq 0} \left( \frac{s}{\epsilon + 2p} \right)^{\ell} \sum_{a=0}^{\ell} \binom{\ell}{a} x^{\ell - a} y^a \\
&= \tilde{W}_{0,0}(\epsilon) \sum_{j,k\geq 0} \binom{j+k}{k} \left( \frac{s}{\epsilon + 2p} \right)^{j+k} x^j y^k
\end{aligned} \tag{D.2.9}$$

Comparing the forms of Eq. (D.2.7) and Eq. (D.2.9), we see the interior elements of controllability Gramian for the unidirectional path is,

$$\tilde{W}_{j,k}(\epsilon) = \frac{1}{\epsilon(\epsilon + 2p)} \left( \frac{s}{\epsilon + 2p} \right)^{j+k} \binom{j+k}{k} \tag{D.2.10}$$

Using the inverse Laplace transform identity 5.2.18 in [250] and applying it to Eq. (D.2.10), the time-varying controllability Gramian is,

$$W_{j,k}(t) = \frac{1}{2p}\left(\frac{s}{2p}\right)^{j+k}\binom{j+k}{k}\left[1 - e^{-2pt}\sum_{\ell=0}^{j+k}\frac{(2pt)^{\ell}}{\ell!}\right]$$

or, for $p > s$, one can use the final value theorem on Eq. (D.2.10) to evaluate the steady state controllability Gramian for the unidirectional path.

$$W_{j,k} = \lim_{t\to\infty} W_{j,k}(t) = \frac{1}{2p}\left(\frac{s}{2p}\right)^{j+k}\binom{j+k}{k} \tag{D.2.11}$$

To examine the decay rate of the Gramian elements, define $R_k = \frac{W_{k,k}}{W_{k-1,k-1}}$, and evaluate the ratio of diagonal elements of Eq. (D.2.11),

$$R_k = \frac{\frac{1}{2p}\left(\frac{s}{2p}\right)^{2k}\frac{(2k)!}{(k!)^2}}{\frac{1}{2p}\left(\frac{s}{2p}\right)^{2k-2}\frac{(2k-2)!}{((k-1)!)^2}} = \left(\frac{s}{2p}\right)^2\frac{2k(2k-1)}{k^2} \tag{D.2.12}$$

In the asymptotic limit of large $k$, the decay rates in Eq. (D.2.12) approach,

$$R = \lim_{k\to\infty} R_k = \left(\frac{s}{p}\right)^2$$

so that the diagonal elements scale as,

$$W_{k,k} \sim R^k = \left(\frac{s}{p}\right)^{2k} \tag{D.2.13}$$

We can derive the scaling of the control energy in the asymptotic limit by inverting Eq. (D.2.13) and taking the logarithm. Relaxing $k$ to be continuous, we can take the derivative of the logarithm of Eq. (D.2.13).

$$\frac{d}{dk}\log E_k \sim \frac{d}{dk}\left(-2k\log\frac{s}{p}\right) = 2\log p - 2\log s \tag{D.2.14}$$

which captures the constant exponential growth seen numerically in the main text. From Eq. (D.2.14), we see that the growth rate is equal to that of the bidirectional chain derived in Eq. (D.1.36) when $p \gg s$.

## D.3 Derivations for the Bidirectional Ring Graph

The derivation of the controllability Gramian of the bidirectional ring discussed in Sec. 5.3.3 proceeds along much the same lines as for the bidirectional path graph. The bidirectional ring is an undirected graph with $|\mathcal{V}| = n$ nodes labeled $\mathcal{V} = \{v_k | k = 0, 1, \dots, n-1\}$ and undirected edges $(v_k, v_{k+1 \bmod n})$. The time-varying controllability Gramian evolves according to the following differential Lyapunov equation.

$$\begin{aligned}\dot{W}_{j,k}(t) &= -2pW_{j,k}(t) + sW_{(j-1) \bmod n,k}(t) + sW_{(j+1) \bmod n,k}(t)\\ &\quad + sW_{j,(k-1) \bmod n}(t) + sW_{j,(k+1) \bmod n}(t) + \sum_{v_a \in \mathcal{D}} \delta_{j,a}\delta_{k,a} \qquad \text{(D.3.1)}\\ W_{j,k}(0) &= 0, \quad \forall j,k \in \{0,1,\dots,n-1\}\end{aligned}$$

The system of differential equations in Eq. (D.3.1) is first converted into a system of algebraic equations with a Laplace transform.

$$\begin{aligned}&\tilde{W}_{j,k}(\epsilon) - \rho\tilde{W}_{(j-1) \bmod n,k}(\epsilon) - \rho\tilde{W}_{(j+1) \bmod n,k}(\epsilon)\\ &\quad - \rho\tilde{W}_{j,(k-1) \bmod n}(\epsilon) - \rho\tilde{W}_{j,(k+1) \bmod n}(\epsilon) = \frac{1}{\epsilon(\epsilon+2p)} \sum_{v_a \in \mathcal{D}} \delta_{j,a}\delta_{k,a}\end{aligned} \qquad \text{(D.3.2)}$$

The parameter $\rho = \frac{s}{\epsilon+2p}$. A discrete Fourier transform (DFT), defined as,

$$\hat{\tilde{W}}_{\hat{j},\hat{k}}(\epsilon) = \sum_{\hat{j},\hat{k}=0}^{n-1} e^{-i\frac{2\pi}{n}j\hat{j}} e^{-i\frac{2\pi}{n}k\hat{k}} \tilde{W}_{j,k}(\epsilon) \qquad \text{(D.3.3)}$$

which has inverse,

$$\tilde{W}_{j,k}(\epsilon) = \frac{1}{n^2} \sum_{j,k=0}^{n-1} e^{i\frac{2\pi}{n}j\hat{j}} e^{i\frac{2\pi}{n}k\hat{k}} \hat{\tilde{W}}_{\hat{j},\hat{k}}(\epsilon) \qquad \text{(D.3.4)}$$

The DFT in Eq. (D.3.3) is applied to the system of equations in Eq. (D.3.2) by multiplying it by $e^{-i\frac{2\pi}{n}j\hat{j}}e^{-i\frac{2\pi}{n}k\hat{k}}$ and summing over all pairs of integers $j, k \in \{0, 1, \ldots, n-1\}$,

$$\left(1 - 2\rho\cos\frac{2\pi}{n}\hat{j} - 2\rho\cos\frac{2\pi}{n}\hat{k}\right)\hat{\tilde{W}}_{\hat{j},\hat{k}}(\epsilon) = \frac{1}{\epsilon(\epsilon+2p)}\sum_{v_a\in\mathcal{D}} e^{-i\frac{2\pi}{n}a\hat{j}}e^{-i\frac{2\pi}{n}a\hat{k}} \tag{D.3.5}$$

Solving for $\hat{\tilde{W}}_{\hat{j},\hat{k}}(\epsilon)$ in Eq. (D.3.5) and applying the inverse DFT in Eq. (D.3.4) yields the terms of the controllability Gramian in the frequency domain.

$$\tilde{W}_{j,k}(\epsilon) = \frac{1}{n^2}\frac{1}{\epsilon(\epsilon+2p)}\sum_{v_a\in\mathcal{D}}\sum_{\hat{j},\hat{k}=0}^{n-1}\frac{e^{-i\frac{2\pi}{n}(a-j)\hat{j}}e^{-i\frac{2\pi}{n}(a-k)\hat{k}}}{1-2\rho\cos\frac{2\pi}{n}\hat{j}-2\rho\cos\frac{2\pi}{n}\hat{k}} \tag{D.3.6}$$

For finite time $t$, the evolution of the original elements of the controllability Gramian of bidirectional ring can be written using the inverse Laplace transform,

$$\begin{aligned} W_{j,k}(t) &= \sum_{v_a\in\mathcal{D}}\int_0^t e^{-2p\tau}\left[\frac{1}{n}\sum_{\hat{j}=0}^{n-1}e^{2s\tau\cos\frac{2\pi}{n}\hat{j}}e^{-i\frac{2\pi}{n}(a-j)\hat{j}}\right]\left[\frac{1}{n}\sum_{\hat{k}=0}^{n-1}e^{2s\tau\cos\frac{2\pi}{n}\hat{k}}e^{-i\frac{2\pi}{n}(a-k)\hat{k}}\right]d\tau \\ &= \sum_{v_a\in\mathcal{D}}\int_0^t e^{-2p\tau}\left[\frac{1}{n}\sum_{\hat{j}=0}^{n-1}e^{2s\tau\cos\frac{2\pi}{n}\hat{j}}\cos\frac{2\pi}{n}(a-j)\hat{j}\right]\left[\frac{1}{n}\sum_{\hat{k}=0}^{n-1}e^{2s\tau\cos\frac{2\pi}{n}\hat{k}}\cos\frac{2\pi}{n}(a-k)\hat{k}\right]d\tau \end{aligned} \tag{D.3.7}$$

On the other hand, if $p > 2s$, then we can compute the steady state controllability Gramian using the final value theorem with Eq. (D.3.6),

$$W_{j,k} = \frac{1}{2n^2}\sum_{v_a\in\mathcal{D}}\sum_{\hat{j},\hat{k}=0}^{n-1}\frac{\cos\left(\frac{2\pi}{n}(a-j)\hat{j}\right)\cos\left(\frac{2\pi}{n}(a-k)\hat{k}\right)}{p-s\cos\frac{2\pi}{n}\hat{j}-s\cos\frac{2\pi}{n}\hat{k}} \tag{D.3.8}$$

In the $n \to \infty$ limit, the bidirectional ring becomes the bidirectional path described previously in this appendix. The relation can be seen by noting the similarity between Eq. (D.1.9) (after applying the result of Eq. (D.1.10)) and the last line of Eq. (D.3.7) for finite time $t$ or for Eq. (D.1.12) and Eq. (D.3.8) for the steady state case.

## D.4 Proof of the Gramian of Unidirectional Ring

The unidirectional ring graph consists of $n$ nodes, $\mathcal{V} = \{v_0, v_1, \ldots, v_{n-1}\}$, and edges $(v_k, v_{k+1 \bmod n})$ for $k = 0, \ldots, n-1$. The graph has uniform edge weights $s$ and self-loops $-p$ for $p > s$. The algebraic Lyapunov equation of the steady state controllability Gramian of the unidirectional ring graph is,

$$\begin{aligned} W_{0,0} - \rho W_{n-1,0} - \rho W_{0,n-1} &= \alpha \\ W_{j,0} - \rho W_{j-1,0} - \rho W_{j,n-1} &= 0, \quad j = 1, \ldots, n-1 \\ W_{0,k} - \rho W_{0,k-1} - \rho W_{n-1,k} &= 0, \quad k = 1, \ldots, n-1 \\ W_{j,k} - \rho W_{j-1,k} - \rho W_{j,k-1} &= 0, \quad j, k = 1, \ldots, n-1 \end{aligned} \tag{D.4.1}$$

where $\rho = \frac{s}{2p}$ and $\alpha = \frac{1}{2p}$. In Chapter 5, the solution of this system of this system of equations is proposed to be,

$$W_{j,k} = \alpha \frac{\sum_{\ell \geq 0} a_{j,k}^{(\ell)} \rho^{\ell n + j + k}}{\sum_{\ell \geq 0} b_\ell \rho^{\ell n}} \tag{D.4.2}$$

The rational form in Eq. (D.4.2) can be proven by plugging it into the system of equations in Eq. (D.4.1) and equating coefficients of $\rho^x$ for powers of $x$. The first line of Eq. (D.4.1) yields the equation,

$$a_{0,0}^{(0)} - b_0 + \sum_{\ell \geq 1} \left( a_{0,0}^{(\ell)} - 2a_{n-1,0}^{(\ell-1)} - b_\ell \right) \rho^{\ell n} = 0 \tag{D.4.3}$$

The second and third lines of Eq. (D.4.1) are equivalent due to the symmetry of the controllability Gramian and so only the second line is applied,

$$\left( a_{j,0}^{(0)} - a_{j-1,0}^{(0)} \right) \rho^j + \sum_{\ell \geq 1} \left( a_{j,0}^{(\ell)} - a_{j-1,0}^{(\ell)} - a_{j,n-1}^{(\ell-1)} \right) \rho^{\ell n + j} = 0 \tag{D.4.4}$$

Finally, the last line of Eq. (D.4.1) is used to determine the remaining relations,

$$\sum_{\ell\geq 0}\left(a_{j,k}^{(\ell)}-a_{j-1,k}^{(\ell)}-a_{j,k-1}^{(\ell)}\right)\rho^{\ell n+j+k}=0 \tag{D.4.5}$$

Putting the system of equations in Eqs. (D.4.3), (D.4.4), and (D.4.5) together yields,

$$\begin{aligned}
a_{0,0}^{(0)}&=b_0\\
a_{j,0}^{(0)}-a_{j-1,0}^{(0)}&=0,\quad j=1,\dots,n-1\\
a_{0,0}^{(\ell)}-2a_{n-1,0}^{(\ell-1)}&=b_\ell,\quad \ell=1,2,\dots\\
a_{j,0}^{(\ell)}-a_{j-1,0}^{(\ell)}-a_{j,n-1}^{(\ell-1)}&=0,\quad j=1,\dots,n-1,\qquad \ell=1,2,\dots\\
a_{j,k}^{(\ell)}-a_{j-1,k}^{(\ell)}-a_{j,k-1}^{(\ell)}&=0,\quad j,k=1,\dots,n-1,\ \ \ell=0,1,\dots
\end{aligned} \tag{D.4.6}$$

As $0<\rho<1$, the first layer, $\ell=0$, can be found by taking the $n\to\infty$ limit,

$$\lim_{n\to\infty}W_{j,k}=\alpha\frac{a_{j,k}^{(0)}\rho^{j+k}}{b_0}=\alpha\rho^{j+k}\binom{j+k}{k}\quad\Rightarrow\quad a_{j,k}^{(0)}=\binom{j+k}{k},\quad b_0=1 \tag{D.4.7}$$

which is the controllability Gramian of the unidirectional path graph derived previously. The solution to the system of equations in Eq. (D.4.6) using Eq. (D.4.7) as the initial values is,

$$a_{j,k}^{(\ell)}=\sum_{p=0}^{\ell}\sum_{q=0}^{p}\binom{np+j+k}{nq+j}b_{\ell-p}=\sum_{p=0}^{\ell}b_{\ell-p}c_{j,k}^{(p)} \tag{D.4.8}$$

where the coefficient in the final form in Eq. (D.4.8) is,

$$c_{j,k}^{(p)}=\sum_{q=0}^{p}\binom{np+j+k}{nq+j}$$

With the values of $a_{j,k}^{(\ell)}$ now expressed in terms of only the values $b_\ell$, $\ell=0,1,\dots$, all that remains is to determine them. To determine the values of $b_\ell$ to ensure that the polynomials in $\rho$ that make up the numerator and denominator in Eq. (5.3.24) have a finite number of non-zero coefficients, we construct the following system of equations.

We set $m = \lceil \frac{n}{2} \rceil$ and set,

$$a_{j,j}^{(m)} = 0, \quad j = 1, \ldots, m$$

The resulting system of equations can be written $M\boldsymbol{b} = \boldsymbol{h}$ where the matrix $M \in \mathbb{Z}^{m\times m}$ and vector $\boldsymbol{h} \in \mathbb{Z}^m$ have elements found by equating coefficients of each $b_\ell$,

$$M_{j,k} = c_{j,j}^{(m-k)}, \quad \boldsymbol{h}_j = -c_{j,j}^{(m)} \tag{D.4.9}$$

The main benefit of this approach is that the coefficients $a_{j,k}^{(\ell)}$ and $b_\ell$ need only be computed once for each value of $n$, independent of the edge weights $s$ and self-loop $p$. Using multi-precision floating-point values, the coefficients for the first few values of $n$ are computed using the linear system in Eq. (D.4.9) and collected in Table 5.1 in Chapter 5.

## D.5 Derivation of the Gramian with Redundancy

This section provides the derivation of the role of redundancy in the control energy presented in Sec. 5.4. The unidirectional balloon graph is the simplest model which captures the both the role of distance and the role of redundancy in a graph whose controllability Gramian can be solved analytically. We assume that there are $b$ redundant paths of length $d$ connecting nodes $v_0$ and $v_d$. The quotient graph of the unidirectional balloon graph is a unidirectional path graph except that edge $(v_{d-1}, v_d)$ has weight $bs$. The controllability Gramian $W_{j,k}$ for elements $j, k = 0, \ldots, d-1$ are equal to the of the unidirectional path graph and it is only when $j = d$ or $k = d$ or both do the controllability Gramian elements change. As such, we solve for these elements directly. First, we can solve for,

$$W_{d,0} = \frac{bs}{2p} W_{d-1,0} = \frac{b}{2p}\left(\frac{s}{2p}\right)^d$$

The general interface elements, for $j = d$ and $k < d$, are found as,

$$W_{d,k} = \frac{s}{2p} W_{d,k-1} + \frac{bs}{2p} W_{d-1,k}, \quad 0 < k < d \tag{D.5.1}$$

From before, we know $W_{d-1,k} = \frac{1}{2p}\left(\frac{s}{2p}\right)^{d+k-1}\binom{d-1+k}{k}$, and we propose that,

$$W_{d,k} = \frac{b}{2p}\left(\frac{s}{2p}\right)^{d+k}\binom{d+k}{k} \tag{D.5.2}$$

which is verified by plugging it into Eq. (D.5.1),

$$\begin{aligned} W_{d,k} &= \frac{1}{2p}\frac{bs}{2p}\left(\frac{s}{2p}\right)^{d+k-1}\binom{d+k-1}{k-1} + \frac{b}{2p}\frac{s}{2p}\left(\frac{s}{2p}\right)^{d+k-1}\binom{d-1+k}{k} \\ &= \frac{b}{2p}\left(\frac{s}{2p}\right)^{d+k}\left[\binom{d+k-1}{k-1} + \binom{d-1+k}{k}\right] \\ &= \frac{b}{2p}\left(\frac{s}{2p}\right)^{d+k}\binom{d+k}{k} \end{aligned}$$

Finally, the term of interest is the diagonal element $W_{d,d}$, which is determined using Eq. (D.5.2),

$$\begin{aligned} W_{d,d} = 2W_{d,d-1} &= 2\frac{b}{2p}\frac{bs}{2p}\left(\frac{s}{2p}\right)^{2d-1}\binom{d+d-1}{d} \\ &= \frac{b^2}{2p}\left(\frac{s}{2p}\right)^{2d}\binom{2d}{d} \end{aligned} \tag{D.5.3}$$

By taking the inverse of Eq. (D.5.3), the control energy is clearly a function of both the distance $d$ and the redundancy $b$, which is written using gamma functions so that we make take derivatives in $d$.

$$E_d = \frac{2p}{b^2}\left(\frac{2p}{s}\right)^{2d}\frac{\Gamma^2(d+1)}{\Gamma(2d+1)} \tag{D.5.4}$$

To better explore the growth in log scale, we take the logarithm of Eq. (D.5.4),

$$\log E_d = \log\frac{2p}{b^2} + 2d\log\frac{2p}{s} + 2\log\Gamma(d+1) - \log\Gamma(2d+1) \tag{D.5.5}$$

With the expression in Eq. (D.5.5), we can compute the rate of increase of the control energy with respect to distance and the redundancy.

$$\begin{aligned}\frac{d}{d(d)}\log E_d &= 2\log\frac{2p}{s}+\frac{2}{d!}\Gamma'(d+1)-\frac{1}{(2d)!}\Gamma'(2d+1)\\ &= 2\log\frac{2p}{s}+2\sum_{k=1}^{d}\frac{1}{k}-2\sum_{k=1}^{2d}\frac{1}{k}\end{aligned} \tag{D.5.6}$$

To determine the asymptotic behavior of Eq. (D.5.6), note the summations can be expressed as the difference of Harmonic numbers,

$$2H_{2d}-2H_d = 2H_{2d}-2\log(2d)-2(H_d-\log d)+2\log 2 \tag{D.5.7}$$

where the additional terms were included to use the identity,

$$\lim_{n\to\infty}\left(H_n-\log(n)\right)=\gamma \tag{D.5.8}$$

where $\gamma$ is the Euler-Mascheroni constant. Applying the identities in Eqs. (D.5.7) and (D.5.8), the asymptotic limit is,

$$\begin{aligned}\lim_{d\to\infty}\frac{d}{d(d)}\log E_d &= 2\log p-2\log s+2\log 2-2\lim_{d\to\infty}\left(H_{2d}-H_d\right)\\ &= 2\log p-2\log s+2\log 2-2\left(c-c+\log 2\right)\\ &= 2\log p-2\log s\end{aligned} \tag{D.5.9}$$

which agrees with the previous results derived for the unidirectional chain using the integral form and is independent of the number of branches. Turning to the number of branches, the rate of change of the log control energy is,

$$\frac{d}{d(b)}\log E_d=-\frac{2}{b} \tag{D.5.10}$$

Note that these relations can only hold for $p > bs$, at which value the adjacency matrix is no longer Hurwitz and the steady state controllability Gramian does not exist. The expressions in Eqs. (D.5.9) and (D.5.10) are used in Sec. 5.4 of the main text to explain the numerical results contained therein.

## D.6 Derivation of the Gramian of Hypercubic Lattices

An $m$-dimensional hypercubic lattice is an undirected graph with nodes, $v_{\boldsymbol{j}}$ at every point $\boldsymbol{j} \in \mathbb{Z}^m$. Let $\boldsymbol{e}_k \in \mathbb{Z}^m$ be the $k$'th unit vector of dimension $m$. The edges of the hypercubic lattice are undirected and consist of,

$$\mathcal{E} = \{(v_{\boldsymbol{j}}, v_{\boldsymbol{j}\pm\boldsymbol{e}_k})|\forall \boldsymbol{j} \in \mathbb{Z}^m, \forall k = 1, \ldots, m\} \tag{D.6.1}$$

Edge weights are defined for each unit vector, $s_k$, $k = 1, \ldots, m$ so that,

$$w(v_{\boldsymbol{j}}, v_{\boldsymbol{j}\pm\boldsymbol{e}_k}) = s_k \tag{D.6.2}$$

The controllability Gramian of the hypercubic lattice, using the connectivity defined in Eq. (D.6.1) and the weights in Eq. (D.6.2), can be written,

$$\begin{aligned}\dot{W}_{\boldsymbol{j},\boldsymbol{k}}(t) = -2pW_{\boldsymbol{j},\boldsymbol{k}}(t) + \sum_{\ell=1}^{m} s_\ell \left(W_{\boldsymbol{j},\boldsymbol{j}+\boldsymbol{e}_\ell}(t) + W_{\boldsymbol{j},\boldsymbol{j}-\boldsymbol{e}_\ell}(t) + W_{\boldsymbol{k},\boldsymbol{k}+\boldsymbol{e}_\ell}(t) + W_{\boldsymbol{k},\boldsymbol{k}-\boldsymbol{e}_\ell}(t)\right) \\ + \sum_{v_{\boldsymbol{a}}\in\mathcal{D}} \delta_{\boldsymbol{j},\boldsymbol{a}}\delta_{\boldsymbol{k},\boldsymbol{a}}, \quad \forall \boldsymbol{j}, \boldsymbol{k} \in \mathbb{Z}^m\end{aligned} \tag{D.6.3}$$

Applying the Laplace transform to Eq. (D.6.3), and defining $\rho_\ell = \frac{s_\ell}{\epsilon+2p}$,

$$\begin{aligned}\tilde{W}_{\boldsymbol{j},\boldsymbol{k}}(\epsilon) - \sum_{\ell=1}^{m} \rho_\ell \left(\tilde{W}_{\boldsymbol{j},\boldsymbol{j}+\boldsymbol{e}_\ell}(\epsilon) + W_{\boldsymbol{j},\boldsymbol{j}-\boldsymbol{e}_\ell}(\epsilon) + W_{\boldsymbol{k},\boldsymbol{k}+\boldsymbol{e}_\ell}(\epsilon) + W_{\boldsymbol{k},\boldsymbol{k}-\boldsymbol{e}_\ell}(\epsilon)\right) \\ = \frac{1}{\epsilon(\epsilon+2p)} \sum_{v_{\boldsymbol{a}}\in\mathcal{D}} \delta_{\boldsymbol{j},\boldsymbol{a}}\delta_{\boldsymbol{k},\boldsymbol{a}}, \quad \forall \boldsymbol{j}, \boldsymbol{k} \in \mathbb{Z}^m\end{aligned} \tag{D.6.4}$$

Define the $2m$-dimensional discrete time Fourier transform as,

$$\hat{\tilde{W}}_{\hat{\boldsymbol{j}},\hat{\boldsymbol{k}}}(\epsilon) = \sum_{\boldsymbol{j},\boldsymbol{k}\in\mathbb{Z}^m} e^{-i\boldsymbol{j}^T\hat{\boldsymbol{j}}} e^{-i\boldsymbol{k}^T\hat{\boldsymbol{k}}} \tilde{W}_{\boldsymbol{j},\boldsymbol{k}}(\epsilon) \tag{D.6.5}$$

with inverse transform,

$$\tilde{W}_{\boldsymbol{j},\boldsymbol{k}}(\epsilon) = \frac{1}{(2\pi)^{2m}} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} e^{i\boldsymbol{j}^T \hat{\boldsymbol{j}}} e^{i\boldsymbol{k}^T \hat{\boldsymbol{k}}} \hat{\tilde{W}}_{\hat{\boldsymbol{j}},\hat{\boldsymbol{k}}}(\epsilon) d\hat{\boldsymbol{j}} d\hat{\boldsymbol{k}} \tag{D.6.6}$$

where each integral is taken over the $m$-dimensional hypercube, centered at the origin, of side length $2\pi$. Applying the DTFT in Eq. (D.6.5) to the controllability Gramian expression in the frequency domain in Eq. (D.6.4).

$$\left[1 - \sum_{\ell=1}^{m} (2\rho_\ell \cos \hat{j}_\ell + 2\rho_\ell \cos \hat{k}_\ell)\right] \hat{\tilde{W}}_{\hat{\boldsymbol{j}},\hat{\boldsymbol{k}}}(\epsilon) = \frac{1}{\epsilon(\epsilon + 2p)} \sum_{v_{\boldsymbol{a}} \in \mathcal{D}} e^{-i\boldsymbol{a}^T \hat{\boldsymbol{j}}} e^{-i\boldsymbol{a}^T \hat{\boldsymbol{k}}} \tag{D.6.7}$$

Solving Eq. (D.6.7) for the transformed controllability Gramian elements,

$$\hat{\tilde{W}}_{\hat{\boldsymbol{j}},\hat{\boldsymbol{k}}}(\epsilon) = \frac{1}{\epsilon} \sum_{v_{\boldsymbol{a}} \in \mathcal{D}} \frac{e^{-i\boldsymbol{a}^T \hat{\boldsymbol{j}}} e^{-i\boldsymbol{a}^T \hat{\boldsymbol{k}}}}{\epsilon + 2p - \sum_{\ell=1}^{m} \left(2s_\ell \cos \hat{j}_\ell + 2s_\ell \cos \hat{k}_\ell\right)} \tag{D.6.8}$$

Applying the inverse DTFT in Eq. (D.6.6) to the expression in Eq. (D.6.8),

$$\tilde{W}_{\boldsymbol{j},\boldsymbol{k}}(\epsilon) = \frac{1}{\epsilon} \sum_{v_{\boldsymbol{a}} \in \mathcal{D}} \frac{1}{(2\pi)^{2m}} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \frac{e^{-i(\boldsymbol{a}-\boldsymbol{j})^T \hat{\boldsymbol{j}}} e^{-i(\boldsymbol{a}-\boldsymbol{k})^T \hat{\boldsymbol{k}}}}{\epsilon + 2p - \sum_{\ell=1}^{m} \left(2s_\ell \cos \hat{j}_\ell + 2s_\ell \cos \hat{k}_\ell\right)} d\hat{\boldsymbol{j}} d\hat{\boldsymbol{k}} \tag{D.6.9}$$

For finite time $t$, the controllability Gramian can be found from the inverse Laplace transform,

$$\begin{aligned} W_{\boldsymbol{j},\boldsymbol{k}}(t) &= \sum_{v_{\boldsymbol{a}} \in \mathcal{D}} \int_0^t e^{-2p\tau} \prod_{\ell=1}^{m} \left[\frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-i(a_\ell - j_\ell)\hat{j}_\ell} e^{2s_\ell \tau \cos \hat{j}_\ell} d\hat{j}_\ell\right] \\ &\quad \times \left[\frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-i(a_\ell - k_\ell)\hat{k}_\ell} e^{2s_\ell \tau \cos \hat{k}_\ell} d\hat{k}_\ell\right] d\tau \\ &= \sum_{v_{\boldsymbol{a}} \in \mathcal{D}} \int_0^t e^{-2p\tau} \prod_{\ell=1}^{m} I_{a_\ell - j_\ell}(2s_\ell \tau) I_{a_\ell - k_\ell}(2s_\ell \tau) d\tau \end{aligned} \tag{D.6.10}$$

where $I_n(z)$ is the modified Bessel function of the first kind of integer order defined in Eq. (D.1.11). Note that for $m = 1$, the controllability Gramian in Eq. (D.6.10) reduces back to the form for the bidirectional path graph as expressed in Eq. (D.1.12). Also, if $p > \sum_{\ell=1}^{m} 2s_\ell$, then we can compute the unique steady state controllability

Gramian using the final value theorem applied to Eq. (D.6.9).

$$W_{\boldsymbol{j},\boldsymbol{k}} = \sum_{v_{\boldsymbol{a}} \in \mathcal{D}} \frac{1}{(2\pi)^{2m}} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \frac{e^{-i(\boldsymbol{a}-\boldsymbol{j})^T \hat{\boldsymbol{j}}} e^{-i(\boldsymbol{a}-\boldsymbol{k})^T \hat{\boldsymbol{k}}}}{2p - \sum_{\ell=1}^{m} \left(2s_\ell \cos \hat{j}_\ell + 2s_\ell \cos \hat{k}_\ell\right)} d\hat{\boldsymbol{j}} d\hat{\boldsymbol{k}} \tag{D.6.11}$$

Again, note that for $m = 1$ in Eq. (D.6.11), we recover the steady state controllability Gramian of the bidirectional chain in Eq. (D.1.18).

## D.7 Derivation of the Gramian for General Lattices

For the general lattice in $m \geq 1$ dimesions, let us define the lattice operator $\Delta$ which defines the connectivity pattern,

$$\Delta W_{\boldsymbol{j},\boldsymbol{k}}(t) = \sum_{\boldsymbol{\ell} \in \mathcal{I}} s_{\boldsymbol{\ell}} W_{\boldsymbol{j}+\ell,\boldsymbol{k}}(t) + s_{\boldsymbol{\ell}} W_{\boldsymbol{j},\boldsymbol{k}+\boldsymbol{\ell}}(t) \tag{D.7.1}$$

The differential Lyapunov equation to solve, in terms of the lattice operator in Eq. (D.7.1), is,

$$\dot{W}_{\boldsymbol{j},\boldsymbol{k}}(t) = -2pW_{\boldsymbol{j},\boldsymbol{k}}(t) + \Delta W_{\boldsymbol{j},\boldsymbol{k}}(t) + \sum_{v_{\boldsymbol{a}} \in \mathcal{D}} \delta_{\boldsymbol{a},\boldsymbol{j}} \delta_{\boldsymbol{a},\boldsymbol{k}} \tag{D.7.2}$$

First, taking the Laplace transform yields,

$$\tilde{W}_{\boldsymbol{j},\boldsymbol{k}}(\epsilon) - \rho(\epsilon) \Delta \tilde{W}_{\boldsymbol{j},\boldsymbol{k}}(\epsilon) = \frac{1}{\epsilon(\epsilon + 2p)} \sum_{v_{\boldsymbol{a}} \in \mathcal{D}} \delta_{\boldsymbol{a},\boldsymbol{j}} \delta_{\boldsymbol{a},\boldsymbol{k}}$$

Applying the discrete time Fourier transform to the Gramian equation in the Laplace domain yields,

$$\left(1 - \rho(\epsilon)\mathcal{F}(\Delta)\right) \hat{\tilde{W}}_{\hat{\boldsymbol{j}},\hat{\boldsymbol{k}}}(\epsilon) = \frac{1}{\epsilon(\epsilon + 2p)} \sum_{v_{\boldsymbol{a}} \in \mathcal{D}} e^{-i\boldsymbol{a}\hat{\boldsymbol{j}}} e^{-i\boldsymbol{a}\hat{\boldsymbol{k}}}$$

where the DTFT of the lattice operator is found to be,

$$\mathcal{F}(\Delta) = \sum_{\boldsymbol{\ell} \in \mathcal{I}} s_{\boldsymbol{\ell}} \left(e^{i\boldsymbol{\ell}\hat{\boldsymbol{j}}} + e^{i\boldsymbol{\ell}\hat{\boldsymbol{k}}}\right) = \phi(\hat{\boldsymbol{j}}) + \phi(\hat{\boldsymbol{k}}) = \sigma(\hat{\boldsymbol{j}}) + \sigma(\hat{\boldsymbol{k}}) + i\omega(\hat{\boldsymbol{j}}) + i\omega(\hat{\boldsymbol{k}})$$

Splitting this term into real and imaginary components for the individual points, $\hat{\boldsymbol{j}}$ and $\hat{\boldsymbol{k}}$, will be useful in the coming derivation. The inverse DTFT is applied to find the Gramian in the Laplace domain,

$$\tilde{W}_{\boldsymbol{j},\boldsymbol{k}}(\epsilon) = \sum_{v_{\boldsymbol{a}} \in \mathcal{D}} \frac{1}{(2\pi)^{2m}} \frac{1}{\epsilon(\epsilon + 2p)} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \frac{e^{i(\boldsymbol{j}-\boldsymbol{a})\boldsymbol{x}} e^{i(\boldsymbol{k}-\boldsymbol{a})\boldsymbol{y}}}{1 - \rho(\epsilon)\mathcal{F}(\Delta)} d\boldsymbol{x} d\boldsymbol{y}$$

where each integral is over the $m$-dimensional cube of side length $2\pi$ centered at the origin. For finite time, the inverse Laplace transform yields,

$$W_{\boldsymbol{j},\boldsymbol{k}}(t) = \sum_{v_{\boldsymbol{a}} \in \mathcal{D}} \int_0^t e^{-2p\tau} A_{\boldsymbol{j}-\boldsymbol{a}}(\tau) A_{\boldsymbol{k}-\boldsymbol{a}}(\tau) d\tau \tag{D.7.3}$$

where the separate functions,

$$\begin{aligned} A_{\boldsymbol{j}}(t) &= \frac{1}{(2\pi)^m} \int_{-\pi}^{\pi} e^{i\boldsymbol{j}\boldsymbol{x}} e^{\phi(\boldsymbol{x})t} d\boldsymbol{x} \\ &= \frac{1}{(2\pi)^m} \int_{-\pi}^{\pi} e^{\sigma(\boldsymbol{x})t} \cos\left(\omega(\boldsymbol{x})t + \boldsymbol{j}\boldsymbol{x}\right) d\boldsymbol{x} \end{aligned} \tag{D.7.4}$$

To prove Eq. (D.7.3) solves Eq. (D.7.2), we differentiate Eq. (D.7.3) with respect to time (remembering that the temporal integral is a convolution with the driver node term) and apply the definition in Eq. (D.7.4),

$$\begin{aligned} \frac{d}{dt} W_{\boldsymbol{j},\boldsymbol{k}}(t) &= \sum_{v_{\boldsymbol{a}} \in \mathcal{D}} \Bigg[ A_{\boldsymbol{j}-\boldsymbol{a}}(0) A_{\boldsymbol{k}-\boldsymbol{a}}(0) + \int_0^t \frac{\partial}{\partial t} e^{-2p(t-\tau)} A_{\boldsymbol{j}-\boldsymbol{a}}(t-\tau) A_{\boldsymbol{k}-\boldsymbol{a}}(t-\tau) d\tau \Bigg] \\ &= \sum_{v_{\boldsymbol{a}} \in \mathcal{D}} \Bigg[ \frac{1}{(2\pi)^m} \Bigg[ \int_{-\pi}^{\pi} \cos((\boldsymbol{j} - \boldsymbol{a}\boldsymbol{x}) d\boldsymbol{x} \Bigg] \Bigg[ \frac{1}{(2\pi)^m} \int_{-\pi}^{\pi} \cos((\boldsymbol{k} - \boldsymbol{a})\boldsymbol{y}) d\boldsymbol{y} \Bigg] \\ &- 2p \int_0^t e^{-2p(t-\tau)} A_{\boldsymbol{j}-\boldsymbol{a}}(t-\tau) A_{\boldsymbol{k}-\boldsymbol{a}}(t-\tau) d\tau \\ &+ \int_0^t e^{-2p(t-\tau)} A'_{\boldsymbol{j}-\boldsymbol{a}}(t-\tau) A_{\boldsymbol{k}-\boldsymbol{a}}(t-\tau) d\tau \\ &+ \int_0^t e^{-2p(t-\tau)} A_{\boldsymbol{j}-\boldsymbol{a}}(t-\tau) A'_{\boldsymbol{k}-\boldsymbol{a}}(t-\tau) d\tau \Bigg] \end{aligned}$$

The derivative of the integrals,

$$\begin{aligned}
A'_{\boldsymbol{j}}(t-\tau) &= \frac{1}{(2\pi)^m}\int_{-\pi}^{\pi}\phi(\boldsymbol{x})e^{\phi(\boldsymbol{x})t}e^{i\boldsymbol{j}\boldsymbol{x}}d\boldsymbol{x} \\
&= \frac{1}{(2\pi)^m}\int_{-\pi}^{\pi}\sum_{\ell\in\mathcal{I}} e^{i\boldsymbol{\ell}\boldsymbol{x}}e^{\phi(\boldsymbol{x})(t-\tau)}e^{i\boldsymbol{j}\boldsymbol{x}}d\boldsymbol{x} \\
&= \sum_{\ell\in\mathcal{I}}\frac{1}{(2\pi)^m}\int_{-\pi}^{\pi}e^{i(\boldsymbol{j}+\boldsymbol{\ell})\boldsymbol{x}}e^{\phi(\boldsymbol{x})t}d\boldsymbol{x} \\
&= \sum_{\ell\in\mathcal{I}}A'_{\boldsymbol{j}+\boldsymbol{\ell}}(t-\tau)
\end{aligned}$$

The initial integrals can be rewritten,

$$A_{\boldsymbol{j}}(0) = \frac{1}{(2\pi)^m}\int_{-\pi}^{\pi}e^{i\boldsymbol{j}\boldsymbol{x}}d\boldsymbol{x} = \frac{1}{(2\pi)^m}\prod_{\ell=1}^{m}\int_{-\pi}^{\pi}e^{i\boldsymbol{j}_\ell\boldsymbol{x}_\ell}d\boldsymbol{x}_\ell = \frac{1}{(2\pi)^m}\prod_{\ell=1}^{m}\frac{2\sin(\pi\boldsymbol{j}_\ell)}{\boldsymbol{j}_\ell} = \begin{cases}1 & \boldsymbol{j}=\boldsymbol{0}\\ 0 & \text{otherwise}\end{cases}$$

With these results, the derivative of the Gramian can be rewritten as,

$$\begin{aligned}
\frac{d}{dt}W_{\boldsymbol{j},\boldsymbol{k}}(t) &= -2pW_{\boldsymbol{j},\boldsymbol{k}}(t)\sum_{v_{\boldsymbol{a}}\in\mathcal{D}}\Bigg[\delta_{\boldsymbol{j},\boldsymbol{a}}\delta_{\boldsymbol{k},\boldsymbol{a}} \\
&\quad+\sum_{\boldsymbol{\ell}\in\mathcal{I}}s_{\boldsymbol{\ell}}\int_0^t e^{-2p(t-\tau)}A_{\boldsymbol{j}+\boldsymbol{\ell}-\boldsymbol{a}}(t-\tau)A_{\boldsymbol{k}-\boldsymbol{a}}(t-\tau)d\tau \\
&\quad+\sum_{\boldsymbol{\ell}\in\mathcal{I}}s_{\boldsymbol{\ell}}\int_0^t e^{-2p(t-\tau)}A_{\boldsymbol{j}-\boldsymbol{a}}(t-\tau)A_{\boldsymbol{k}+\boldsymbol{\ell}-\boldsymbol{a}}(t-\tau)d\tau\Bigg] \\
&= -2pW_{\boldsymbol{j},\boldsymbol{k}}(t)+\sum_{\boldsymbol{\ell}\in\mathcal{I}}s_{\boldsymbol{\ell}}\left(W_{\boldsymbol{j}+\boldsymbol{\ell},\boldsymbol{k}}(t)+W_{\boldsymbol{j},\boldsymbol{k}+\boldsymbol{\ell}}(t)\right)+\sum_{v_{\boldsymbol{a}}\in\mathcal{D}}\delta_{\boldsymbol{j},\boldsymbol{a}}\delta_{\boldsymbol{k},\boldsymbol{a}} \\
&= -2pW_{\boldsymbol{j},\boldsymbol{k}}(t)+\Delta W_{\boldsymbol{j},\boldsymbol{k}}(t)+\sum_{v_{\boldsymbol{a}}\in\mathcal{D}}\delta_{\boldsymbol{j},\boldsymbol{a}}\delta_{\boldsymbol{k},\boldsymbol{a}}
\end{aligned}$$

which is the original expression in Eq. (D.7.2) thus completing the proof.

# References

[1] Isaac Klickstein, Afroza Shirin, and Francesco Sorrentino. “Energy scaling of targeted optimal control of complex networks”. In: *Nature Communications* 8 (Apr. 2017), p. 15145.

[2] Afroza Shirin, Isaac Klickstein, and Francesco Sorrentino. “Optimal control of complex networks: Balancing accuracy and energy of the control action”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 27.4 (Apr. 2017), p. 041103.

[3] Isaac Klickstein, Afroza Shirin, and Francesco Sorrentino. “Locally Optimal Control of Complex Networks”. In: *Physical Review Letters* 119.26 (Dec. 2017), p. 268301.

[4] Isaac Klickstein et al. “Energy Scaling with Control Distance in Complex Networks”. In: *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, May 2018, pp. 1–5.

[5] Isaac Samuel Klickstein and Francesco Sorrentino. “Control Distance and Energy Scaling of Complex Networks”. In: *IEEE Transactions on Network Science and Engineering* (2018).

[6] Isaac Klickstein and Francesco Sorrentino. “Control Energy of Lattice Graphs”. In: *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, Dec. 2018, pp. 6132–6138.

[7] Ishan Kafle et al. “Optimal control of networks in the presence of attackers and defenders”. In: *Chaos* 28.5 (2018).

[8] Isaac Samuel Klickstein and Francesco Sorrentino. “Generating Graphs with Symmetry”. In: *IEEE Transactions on Network Science and Engineering* (2018).

[9] Isaac Klickstein and Francesco Sorrentino. “Generating symmetric graphs”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 28.12 (Dec. 2018), p. 121102.

[10] Afroza Shirin et al. “Prediction of Optimal Drug Schedules for Controlling Autophagy”. In: *Scientific Reports* 9.1 (Dec. 2019), p. 1428.

[11] Afroza Shirin et al. “Optimal regulation of blood glucose level in Type I diabetes using insulin and glucagon”. In: *PLOS ONE* 14.3 (Mar. 2019). Ed. by Abhyudai Singh, e0213665.

[12] Isaac Klickstein, Louis Pecora, and Francesco Sorrentino. “Symmetry induced group consensus”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29.7 (July 2019), p. 073101.

[13] Afroza Shirin, Isaac S. Klickstein, and Francesco Sorrentino. "Stability analysis of reservoir computers dynamics via Lyapunov functions". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29.10 (Oct. 2019), p. 103147.

[14] Isaac Klickstein and Francesco Sorrentino. "The controllability Gramian of lattice graphs". In: *Automatica* 114 (Apr. 2020), p. 108833.

[15] Fabio Della Rossa et al. "Symmetries and Cluster Synchronization in Multilayer Networks". In: *Nature Communications* ACCEPTED (2020).

[16] John E. Hopcroft and Richard M. Karp. "An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs". In: *SIAM Journal on Computing* 2.4 (Dec. 1973), pp. 225–231.

[17] Mark E J Newman. *Networks: an introduction*. Oxford university press, 2010.

[18] Elliott W. Montroll and George H. Weiss. "Random Walks on Lattices. II". In: *Journal of Mathematical Physics* 6.2 (Feb. 1965), pp. 167–181.

[19] Shigetoshi Katsura et al. "Lattice Green's Function. Introduction". In: *Journal of Mathematical Physics* 12.5 (May 1971), pp. 892–895.

[20] Paul Erdős and Alfréd Rényi. "On random graphs I". In: *Publ. Math. Debrecen* 6 (1959), pp. 290–297.

[21] Duncan J Watts and Steven H Strogatz. "Collective dynamics of 'small-world' networks". In: *nature* 393.6684 (1998), p. 440.

[22] Albert-László Barabási and Réka Albert. "Emergence of scaling in random networks". In: *Science* 286.5439 (1999), pp. 509–512.

[23] K.-I. Goh, Byungnam Kahng, and Doochul Kim. "Spectra and eigenvectors of scale-free networks". In: *Physical Review E* 64.5 (2001), p. 51903.

[24] Sergey N Dorogovtsev and José Ferreira F Mendes. "Scaling properties of scale-free evolving networks: Continuous approach". In: *Physical Review E* 63.5 (Apr. 2001), p. 056125.

[25] Romualdo Pastor-Satorras and Alessandro Vespignani. "Epidemic spreading in scale-free networks". In: *Physical review letters* 86.14 (2001), p. 3200.

[26] N Schwartz et al. "Percolation in directed scale-free networks". In: *Physical Review E* 66.1 (2002), p. 15104.

[27] Aaron Clauset, Cosma Rohilla Shalizi, and Mark E J Newman. "Power-law distributions in empirical data". In: *SIAM review* 51.4 (2009), pp. 661–703.

[28] Ivan Voitalov et al. "Scale-free networks well done". In: *Physical Review Research* 1.3 (Oct. 2019), p. 033034.

[29] Anna D. Broido and Aaron Clauset. "Scale-free networks are rare". In: *Nature Communications* 10.1 (Dec. 2019), p. 1017.

[30] Albert-Laszlo Barabasi. *Love is all you need: Clauset's fruitless search for scale-free networks*. 2018.

[31] Michael Molloy and Bruce Reed. "A critical point for random graphs with a given degree sequence". In: *Random structures & algorithms* 6.2-3 (1995), pp. 161–180.

[32] Michael Molloy and Bruce Reed. “The size of the giant component of a random graph with a given degree sequence”. In: *Combinatorics, probability and computing* 7.3 (1998), pp. 295–305.

[33] Josef Lauri and Raffaele Scapellato. *Topics in Graph Automorphisms and Reconstruction.* Cambridge University Press, 2016.

[34] Igor V. Belykh and Martin Hasler. “Mesoscale and clusters of synchrony in networks of bursting neurons”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 21.1 (2011), p. 16106.

[35] Ronald C. Read and Derek G. Corneil. “The graph isomorphism disease”. In: *Journal of Graph Theory* 1.4 (1977), pp. 339–363.

[36] Brendan D McKay. *Practical graph isomorphism.* 1981. arXiv: 1301.1493.

[37] Brendan D. McKay and Adolfo Piperno. “Practical graph isomorphism, II”. In: *Journal of Symbolic Computation* 60 (Jan. 2014), pp. 94–112.

[38] Réka Albert and Albert-László Barabási. “Statistical mechanics of complex networks”. In: *Reviews of Modern Physics* 74.1 (Jan. 2002), pp. 47–97.

[39] Mark E J Newman, Albert-László Barabási, and Duncan J Watts. *The structure and dynamics of networks.* Princeton University Press, 2006.

[40] Stefano Boccaletti et al. “Complex networks: Structure and dynamics”. In: *Physics Reports* 424.4-5 (Feb. 2006), pp. 175–308.

[41] Albert-László Barabási. *Network Science.* Cambridge university press, 2016.

[42] Michelle Girvan and Mark E J Newman. “Community structure in social and biological networks”. In: *Proceedings of the national academy of sciences* 99.12 (2002), pp. 7821–7826.

[43] Hawoong Jeong et al. “The large-scale organization of metabolic networks”. In: *Nature* 407.6804 (2000), p. 651.

[44] Réka Albert. “Scale-free networks in cell biology”. In: *Journal of cell science* 118.21 (2005), pp. 4947–4957.

[45] Danielle S Bassett and Olaf Sporns. “Network neuroscience”. In: *Nature Neuroscience* 20.3 (Mar. 2017), pp. 353–364.

[46] Sarah Feldt Muldoon et al. “Stimulation-Based Control of Dynamic Brain Networks”. In: *PLOS Computational Biology* 12.9 (Sept. 2016). Ed. by Claus C. Hilgetag, e1005076.

[47] Gang Yan et al. “Network control principles predict neuron function in the Caenorhabditis elegans connectome”. In: *Nature* 550.7677 (2017), p. 519.

[48] Giuliano Andrea Pagani and Marco Aiello. “The power grid as a complex network: a survey”. In: *Physica A: Statistical Mechanics and its Applications* 392.11 (2013), pp. 2688–2700.

[49] Réka Albert, István Albert, and Gary L Nakarado. “Structural vulnerability of the North American power grid”. In: *Physical Review E* 69.2 (Feb. 2004), p. 025103.

[50] Sergio Arianos et al. “Power grid vulnerability: A complex network approach”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 19.1 (Mar. 2009), p. 013119.

[51] Chengcheng Shao et al. “The spread of low-credibility content by social bots”. In: *Nature Communications* 9.1 (Nov. 2018), p. 4787.

[52] Haewoon Kwak et al. “What is Twitter, a social network or a news media?” In: *Proceedings of the 19th international conference on World wide web - WWW '10*. New York, New York, USA: ACM Press, 2010, p. 591. arXiv: `0809.1869v1`.

[53] Alessandro Vespignani. “Modelling dynamical processes in complex socio-technical systems”. In: *Nature physics* 8.1 (2012), p. 32.

[54] Cristopher Moore and Mark E J Newman. “Epidemics and percolation in small-world networks”. In: *Physical Review E* 61.5 (2000), p. 5678.

[55] Cameron Nowzari, Victor M Preciado, and George J Pappas. “Analysis and control of epidemics: A survey of spreading processes on complex networks”. In: *IEEE Control Systems* 36.1 (2016), pp. 26–46.

[56] Vito Latora and Massimo Marchiori. “Is the Boston subway a small-world network?” In: *Physica A: Statistical Mechanics and its Applications* 314.1-4 (Nov. 2002), pp. 109–113.

[57] Gang Yan et al. “Efficient routing on complex networks”. In: *Physical Review E* 73.4 (Apr. 2006), p. 046108.

[58] Alain Barrat, Marc Barthelemy, and Alessandro Vespignani. *Dynamical processes on complex networks*. Cambridge university press, 2008.

[59] René Thomas, Denis Thieffry, and Marcelle Kaufman. “Dynamical behaviour of biological regulatory networks—I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state”. In: *Bulletin of Mathematical Biology* 57.2 (Mar. 1995), pp. 247–276.

[60] John J Grainger and William D Stevenson Jr. *Power system analysis*. 2nd ed. McGraw-Hill, 2004.

[61] Steven H Strogatz. “From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators”. In: *Physica D: Nonlinear Phenomena* 143.1-4 (Sept. 2000), pp. 1–20.

[62] Dennis S Bernstein. *Matrix Mathematics: Theory, Facts, and Formulas*. 2nd ed. Princeton university press, 2009.

[63] Joao P Hespanha. *Linear systems theory*. Princeton university press, 2009.

[64] R. Kalman. “On the general theory of control systems”. In: *IRE Transactions on Automatic Control* 4.3 (Dec. 1959), pp. 110–110.

[65] K. Murota and S. Poljak. “Note on a graph-theoretic criterion for structural output controllability”. In: *IEEE Transactions on Automatic Control* 35.8 (1990), pp. 939–942.

[66] R. E. Kalman. “Mathematical Description of Linear Dynamical Systems”. In: *Journal of the Society for Industrial and Applied Mathematics Series A Control* 1.2 (Jan. 1963), pp. 152–192.

[67] M.L.J. Hautus. “Stabilization controllability and observability of linear autonomous systems”. In: *Indagationes Mathematicae (Proceedings)* 73 (1970), pp. 448–455.

[68] Ching-Tai Lin. “Structural controllability”. In: *IEEE Transactions on Automatic Control* 19.3 (June 1974), pp. 201–208.

[69] R. Shields and J. Pearson. “Structural controllability of multiinput linear systems”. In: *IEEE Transactions on Automatic Control* 21.2 (Apr. 1976), pp. 203–212.

[70] Jean-Michel Dion, Christian Commault, and Jacob van der Woude. “Generic properties and control of linear structured systems: a survey”. In: *Automatica* 39.7 (July 2003), pp. 1125–1144.

[71] Hirokazu Mayeda and Takashi Yamada. “Strong Structural Controllability”. In: *SIAM Journal on Control and Optimization* 17.1 (Jan. 1979), pp. 123–138.

[72] Vasileios Tzoumas, Ali Jadbabaie, and George J Pappas. “Minimal reachability problems”. In: *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE. IEEE, Dec. 2015, pp. 4220–4225.

[73] Yang-Yu Liu and Albert-László Barabási. “Control principles of complex systems”. In: *Reviews of Modern Physics* 88.3 (2016), p. 35006.

[74] Adilson E Motter. “Networkcontrology”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 25.9 (Sept. 2015), p. 97621.

[75] Yang-Yu Liu, Jean-Jacques Slotine, and Albert-László Barabási. “Controllability of complex networks”. In: *Nature* 473.7346 (May 2011), pp. 167–173.

[76] Franz-Josef Müller and Andreas Schuppert. “Few inputs can reprogram biological networks”. In: *Nature* 478.7369 (Oct. 2011), E4–E4.

[77] Marc Mézard and Giorgio Parisi. “The Bethe lattice spin glass revisited”. In: *The European Physical Journal B-Condensed Matter and Complex Systems* 20.2 (2001), pp. 217–233.

[78] Lenka Zdeborová and Marc Mézard. “The number of matchings in random graphs”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2006.05 (May 2006), P05003–P05003.

[79] Lenka Zdeborová and Florent Krzakala. “Statistical physics of inference: Thresholds and algorithms”. In: *Advances in Physics* 65.5 (2016), pp. 453–552.

[80] Christian Commault and Jean-Michel Dion. “Input addition and leader selection for the controllability of graph-based systems”. In: *Automatica* 49.11 (Nov. 2013), pp. 3322–3328.

[81] Noah J Cowan et al. “Nodal Dynamics, Not Degree Distributions, Determine the Structural Controllability of Complex Networks”. In: *PLoS ONE* 7.6 (2012), e38398.

[82] Sergio Pequito, Soummya Kar, and A Pedro Aguiar. “A structured systems approach for optimal actuator-sensor placement in linear time-invariant systems”. In: *2013 American Control Conference*. IEEE, June 2013, pp. 6108–6113.

[83] Alexander Olshevsky. “Minimum input selection for structural controllability”. In: *2015 American Control Conference (ACC)*. IEEE. IEEE, July 2015, pp. 2218–2223.

[84] Sergio Pequito, Soummya Kar, and A Pedro Aguiar. "A Framework for Structural Input/Output and Control Configuration Selection in Large-Scale Systems". In: *IEEE Transactions on Automatic Control* 61.2 (Feb. 2016), pp. 303–318.

[85] Sérgio Pequito, Soummya Kar, and A Pedro Aguiar. "Minimum cost input/output design for large-scale linear structural systems". In: *Automatica* 68 (June 2016), pp. 384–391.

[86] Sérgio Pequito et al. "The robust minimal controllability problem". In: *Automatica* 82 (2017), pp. 261–268.

[87] Christian Commault and Jean-Michel Dion. "The single-input Minimal Controllability Problem for structured systems". In: *Systems & Control Letters* 80 (June 2015), pp. 50–55.

[88] Wen-Xu Wang et al. "Optimizing controllability of complex networks by minimum structural perturbations". In: *Physical Review E* 85.2 (2012), p. 26115.

[89] Airlie Chapman and Mehran Mesbahi. "On strong structural controllability of networked systems: A constrained matching approach". In: *2013 American Control Conference*. Washington, DC, USA: IEEE, June 2013, pp. 6126–6131.

[90] Márton Pósfai and Philipp Hövel. "Structural controllability of temporal networks". In: *New Journal of Physics* 16.12 (2014), p. 123055.

[91] Aming Li et al. "The fundamental advantages of temporal networks". In: *Science* 358.6366 (2017), pp. 1042–1046.

[92] Tamás Nepusz and Tamás Vicsek. "Controlling edge dynamics in complex networks". In: *Nature Physics* 8.7 (2012), pp. 568–573.

[93] Shima Sadat Mousavi, Mohammad Haeri, and Mehran Mesbahi. "On the Structural and Strong Structural Controllability of Undirected Networks". In: *IEEE Transactions on Automatic Control* 63.7 (July 2018), pp. 2234–2241.

[94] Zhengzhong Yuan et al. "Exact controllability of complex networks". In: *Nature Communications* 4.1 (Dec. 2013), p. 2447.

[95] Marzieh Nabi-Abdolyousefi and Mehran Mesbahi. "On the Controllability Properties of Circulant Networks". In: *IEEE Transactions on Automatic Control* 58.12 (Dec. 2013), pp. 3179–3184.

[96] Alexander Olshevsky. "Minimal controllability problems". In: *IEEE Transactions on Control of Network Systems* 1.3 (2014), pp. 249–258.

[97] Zhengzhong Yuan et al. "Exact controllability of multiplex networks". In: *New Journal of Physics* 16.10 (2014), p. 103036.

[98] Richard M. Karp. "Reducibility among Combinatorial Problems". In: *Complexity of Computer Computations*. Boston, MA: Springer US, 1972, pp. 85–103.

[99] Ali Jadbabaie et al. "Minimal reachability is hard to approximate". In: *IEEE Transactions on Automatic Control* (2018).

[100] Jianxi Gao et al. "Target control of complex networks". In: *Nature Communications* 5.1 (Dec. 2014), p. 5415.

[101] Francesco Lo Iudice, Franco Garofalo, and Francesco Sorrentino. “Structural permeability of complex networks to control signals”. In: *Nature Communications* 6.1 (Dec. 2015), p. 8349.

[102] Nima Monshizadeh, Kanat Camlibel, and Harry Trentelman. “Strong targeted controllability of dynamical networks”. In: *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, Dec. 2015, pp. 4782–4787.

[103] Henk J. van Waarde, M. Kanat Camlibel, and Harry L. Trentelman. “A Distance-Based Approach to Strong Target Control of Dynamical Networks”. In: *IEEE Transactions on Automatic Control* 62.12 (Dec. 2017), pp. 6266–6277.

[104] Amirreza Rahmani et al. “Controllability of multi-agent systems from a graph-theoretic perspective”. In: *SIAM Journal on Control and Optimization* 48.1 (2009), pp. 162–186.

[105] Airlie Chapman and Mehran Mesbahi. “On symmetry and controllability of multi-agent systems”. In: *53rd IEEE Conference on Decision and Control*. Los Angeles, CA, USA: IEEE, Dec. 2014, pp. 625–630.

[106] Airlie Chapman and Mehran Mesbahi. “State controllability, output controllability and stabilizability of networks: A symmetry perspective”. In: *2015 54th IEEE Conference on Decision and Control (CDC)*. Osaka, Japan: IEEE, Dec. 2015, pp. 4776–4781.

[107] Edward R Scheinerman and Daniel H Ullman. *Fractional graph theory: A rational approach to the theory of graphs*. John Wiley & Sons, 2008.

[108] Yang-Yu Liu, Jean-Jacques Slotine, and Albert-László Barabási. “Control Centrality and Hierarchical Structure in Complex Networks”. In: *PLoS ONE* 7.9 (Sept. 2012). Ed. by Yamir Moreno, e44459.

[109] S. Hosoe. “Determination of generic dimensions of controllable subspaces and its application”. In: *IEEE Transactions on Automatic Control* 25.6 (Dec. 1980), pp. 1192–1196.

[110] Bingbo Wang, Lin Gao, and Yong Gao. “Control range: a controllability-based index for node significance in directed networks”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2012.04 (2012), P04011.

[111] Tao Jia et al. “Emergence of bimodality in controlling complex networks”. In: *Nature communications* 4 (2013), p. 2002.

[112] Jose C Nacher and Tatsuya Akutsu. “Analysis on controlling complex networks based on dominating sets”. In: *Journal of Physics: Conference Series*. Vol. 410. 1. IOP Publishing. 2013, p. 12104.

[113] Jose C Nacher and Tatsuya Akutsu. “Analysis of critical and redundant nodes in controlling directed and undirected complex networks using dominating sets”. In: *Journal of Complex Networks* 2.4 (2014), pp. 394–412.

[114] Jose C Nacher and Tatsuya Akutsu. “Minimum dominating set-based methods for analyzing biological networks”. In: *Methods* 102 (2016), pp. 57–63.

[115] Tao Jia and Albert-László Barabási. “Control capacity and a random sampling method in exploring controllability of complex networks”. In: *Scientific reports* 3 (2013), p. 2354.

[116] Tao Jia and Márton Pósfai. “Connecting core percolation and controllability of complex networks”. In: *Scientific reports* 4 (2014), p. 5379.

[117] J. Ruths and Derek Ruths. “Control Profiles of Complex Networks”. In: *Science* 343.6177 (Mar. 2014), pp. 1373–1376.

[118] Thomas Kailath. *Linear systems*. Prentice hall Englewood Cliffs, NJ, 1980.

[119] Athanasios Antoulas. *Approximation of Large-Scale Dynamical Systems*. Hoston, Texas, 2005.

[120] Sven J Hammarling. “Numerical Solution of the Stable, Non-negative Definite Lyapunov Equation Lyapunov Equation”. In: *IMA Journal of Numerical Analysis* 2.3 (1982), pp. 303–323.

[121] Peter Benner et al. “SLICOT—A Subroutine Library in Systems and Control Theory”. In: *Applied and Computational Control, Signals, and Circuits*. Boston, MA: Birkhäuser Boston, 1999, pp. 499–539.

[122] Thilo Penzl. “A cyclic low-rank Smith method for large sparse Lyapunov equations”. In: *SIAM Journal on Scientific Computing* 21.4 (1999), pp. 1401–1418.

[123] Peter Benner, Jing-Rebecca Li, and Thilo Penzl. “Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems”. In: *Numerical Linear Algebra with Applications* 15.9 (Nov. 2008), pp. 755–777.

[124] Peter Benner and Jens Saak. “Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: a state of the art survey”. In: *GAMM-Mitteilungen* 36.1 (2013), pp. 32–52.

[125] Gang Yan et al. “Controlling complex networks: How much energy is needed?” In: *Physical review letters* 108.21 (2012), p. 218703.

[126] Gang Yan et al. “Spectrum of controlling and observing complex networks”. In: *Nature Physics* 11.9 (Sept. 2015), pp. 779–786.

[127] Torbjörn Granlund. *GNU MP: The GNU Multiple Precision Arithmetic Library*. 2020.

[128] *GNU MPFR: The Multiple Precision Floating-Point Reliable Library*. 2018.

[129] Andreas Enge, Philippe Theveny, and Paul Zimmermann. *GNU MPC: The Gnu Multiple Precision Complex Library*. 2019.

[130] Advanpix LLC. *Multiprecision Computing Toolbox for MATLAB*. 2016.

[131] Jie Sun and Adilson E Motter. “Controllability Transition and Nonlocality in Network Control”. In: *Physical Review Letters* 110.20 (May 2013), p. 208701.

[132] Fabio Pasqualetti, Sandro Zampieri, and Francesco Bullo. “Controllability Metrics, Limitations and Algorithms for Complex Networks”. In: *IEEE Transactions on Control of Network Systems* 1.1 (Mar. 2014), pp. 40–52.

[133] Shi Gu et al. “Controllability of structural brain networks”. In: *Nature Communications* 6.1 (Dec. 2015), p. 8414.

[134] Sérgio Pequito et al. “Trade-offs between driving nodes and time-to-control in complex networks”. In: *Scientific Reports* 7.1 (Dec. 2017), p. 39978.

[135] Tyler H Summers and John Lygeros. “Optimal sensor and actuator placement in complex dynamical networks”. In: *IFAC Proceedings Volumes* 47.3 (2014), pp. 3784–3789.

[136] Fabrizio L Cortesi, Tyler H Summers, and John Lygeros. “Submodularity of energy related controllability metrics”. In: *53rd IEEE Conference on Decision and Control*. IEEE. IEEE, Dec. 2014, pp. 2883–2888.

[137] Tyler Summers. “Actuator placement in networks using optimal control performance metrics”. In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE. IEEE, Dec. 2016, pp. 2703–2708.

[138] Tyler H Summers, Fabrizio L Cortesi, and John Lygeros. “On submodularity and controllability in complex dynamical networks”. In: *IEEE Transactions on Control of Network Systems* 3.1 (2016), pp. 91–101.

[139] Satoru Fujishige. *Submodular functions and optimization*. Elsevier, 2005.

[140] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. “An analysis of approximations for maximizing submodular set functions—I”. In: *Mathematical Programming* 14.1 (Dec. 1978), pp. 265–294.

[141] Marshall L Fisher et al. “An analysis of approximations for maximizing submodular set functions—II”. In: *Polyhedral combinatorics*. Vol. 14. 1. Springer, 1978, pp. 265–294.

[142] Airlie Chapman and Mehran Mesbahi. “System Theoretic Aspects of Influenced Consensus: Single Input Case”. In: *IEEE Transactions on Automatic Control* 57.6 (June 2012), pp. 1505–1511.

[143] Vasileios Tzoumas, Ali Jadbabaie, and George J Pappas. “Sensor placement for optimal Kalman filtering: Fundamental limits, submodularity, and algorithms”. In: *American Control Conference (ACC), 2016*. IEEE. 2016, pp. 191–196.

[144] Vasileios Tzoumas et al. “Selecting Sensors in Biological Fractional-Order Systems”. In: *IEEE Transactions on Control of Network Systems* 5.2 (June 2018), pp. 709–721.

[145] Andrew Clark, Linda Bushnell, and Radha Poovendran. “On leader selection for performance and controllability in multi-agent systems”. In: *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, Dec. 2012, pp. 86–93.

[146] Andrew Clark et al. “Minimizing Convergence Error in Multi-Agent Systems Via Leader Selection: A Supermodular Optimization Approach”. In: *IEEE Transactions on Automatic Control* 59.6 (June 2014), pp. 1480–1494.

[147] Andrew Clark, Linda Bushnell, and Radha Poovendran. “A Supermodular Optimization Framework for Leader Selection Under Link Noise in Linear Multi-Agent Systems”. In: *IEEE Transactions on Automatic Control* 59.2 (Feb. 2014), pp. 283–296.

[148] Vasileios Tzoumas et al. “Minimal actuator placement with optimal control constraints”. In: *2015 American Control Conference (ACC)*. IEEE. IEEE, July 2015, pp. 2081–2086.

[149] Vasileios Tzoumas et al. “Minimal actuator placement with bounds on control effort”. In: *IEEE Transactions on Control of Network Systems* 3.1 (2016), pp. 67–78.

[150] L. A. Wolsey. “An analysis of the greedy algorithm for the submodular set covering problem”. In: *Combinatorica* 2.4 (Dec. 1982), pp. 385–393.

[151] Guoqi Li et al. “Minimum-cost control of complex networks”. In: *New Journal of Physics* 18.1 (2016), p. 13012.

[152] Guoqi Li et al. “Optimal control of complex networks based on matrix differentiation”. In: *EPL (Europhysics Letters)* 115.6 (Sept. 2016), p. 68005.

[153] Leitao Gao et al. “Towards the minimum-cost control of target nodes in directed networks with linear dynamics”. In: *Journal of the Franklin Institute* 355.16 (Nov. 2018), pp. 8141–8157.

[154] Guoqi Li et al. “Minimum Cost Control of Directed Networks With Selectable Control Inputs”. In: *IEEE Transactions on Cybernetics* (2018), pp. 1–10.

[155] Guoqi Li et al. “Key-nodes selection problem for minimum cost control of directed networks”. In: *Optimal Control Applications and Methods* 39.1 (Jan. 2018), pp. 95–113.

[156] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[157] Guoqi Li et al. “Enabling Controlling Complex Networks with Local Topological Information”. In: *Scientific reports* 8.1 (2018), p. 4593.

[158] Donald E Kirk. *Optimal control theory: an introduction*. Courier Corporation, 2012.

[159] I Michael Ross. *A primer on Pontryagin’s principle in optimal control*. Collegiate publishers, 2015.

[160] Fariba Fahroo and I. Michael Ross. “Advances in Pseudospectral Methods for Optimal Control”. In: *AIAA Guidance, Navigation and Control Conference and Exhibit*. Reston, Virigina: American Institute of Aeronautics and Astronautics, Aug. 2008.

[161] Divya Garg et al. “A unified framework for the numerical solution of optimal control problems using pseudospectral methods”. In: *Automatica* 46.11 (Nov. 2010), pp. 1843–1851.

[162] I. Michael Ross and Mark Karpenko. “A review of pseudospectral optimal control: From theory to flight”. In: *Annual Reviews in Control* 36.2 (Dec. 2012), pp. 182–197.

[163] Michael A. Patterson and Anil V. Rao. “GPOPS-II”. In: *ACM Transactions on Mathematical Software* 41.1 (Oct. 2014), pp. 1–37.

[164] Andreas Wächter and Lorenz T Biegler. “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming”. In: *Mathematical programming* 106.1 (2006), pp. 25–57.

[165] Philip E. Gill, Walter Murray, and Michael A. Saunders. “SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization”. In: *SIAM Review* 47.1 (Jan. 2005), pp. 99–131.

[166] I. Michael Ross. “User’s manual for DIDO: a MATLAB application package for solving optimal control problems”. In: *Tomlab Optimization* (2004).

[167] Victor M Becerra. “Solving complex optimal control problems at no cost with PSOPT”. In: *Computer-Aided Control System Design (CACSD), 2010 IEEE International Symposium on*. IEEE. 2010, pp. 1391–1396.

[168] D. J. Klionsky. “Autophagy as a Regulated Pathway of Cellular Degradation”. In: *Science* 290.5497 (Dec. 2000), pp. 1717–1721.

[169] T. Shintani. “Autophagy in Health and Disease: A Double-Edged Sword”. In: *Science* 306.5698 (Nov. 2004), pp. 990–995.

[170] Gene H Golub and Charles F Van Loan. *Matrix computations*. Vol. 3. JHU Press, 2012.

[171] K.-I. Goh, B Kahng, and D Kim. “Universal Behavior of Load Distribution in Scale-Free Networks”. In: *Physical Review Letters* 87.27 (Dec. 2001), p. 278701.

[172] R. Milo. “Superfamilies of Evolved and Designed Networks”. In: *Science* 303.5663 (Mar. 2004), pp. 1538–1542.

[173] Vladimir Batagelj and Andrej Mrvar. “Pajek – program for large network analysis”. In: *Connections* 21 (1998), pp. 47–57.

[174] Kevin D Lafferty et al. “Food webs and parasites in a salt marsh ecosystem”. In: *Disease ecology: community structure and pathogen dynamics*. Ed. by Sharon K Collinge and Chris Ray. Oxford university press, 2006. Chap. 9, pp. 119–134.

[175] Neo D. Martinez. “Artifacts or Attributes? Effects of Resolution on the Little Rock Lake Food Web”. In: *Ecological Monographs* 61.4 (Feb. 1991), pp. 367–392.

[176] S.J. Hall and D. Raffaelli. “Food-Web Patterns: Lessons from a Species-Rich Web”. In: *The Journal of Animal Ecology* 60.3 (Oct. 1991), p. 823.

[177] *Air traffic control*.

[178] Rich Christie. *Power Systems Test Case Archive*. 1993.

[179] Peter J Menck et al. “How dead ends undermine power grid stability”. In: *Nature communications* 5 (2014), p. 3969.

[180] Vittoria Colizza, Romualdo Pastor-Satorras, and Alessandro Vespignani. “Reaction–diffusion processes and metapopulation models in heterogeneous networks”. In: *Nature Physics* 3.4 (Apr. 2007), pp. 276–282. arXiv: `0703129 [cond-mat]`.

[181] Haiyuan Yu et al. “Next-generation sequencing to generate interactome datasets”. In: *Nature Methods* 8.6 (June 2011), pp. 478–480.

[182] *Yeast Interactome Project*.

[183] Haiyuan Yu et al. “High-quality binary protein interaction map of the yeast interactome network”. In: *Science* (2008).

[184] R. Guimerà et al. “Self-similar community structure in a network of human interactions”. In: *Physical Review E* 68.6 (Dec. 2003), p. 065103.

[185] Tore Opsahl. “Triadic closure in two-mode networks: Redefining the global and local clustering coefficients”. In: *Social Networks* 35.2 (May 2013), pp. 159–167.

[186] Pablo M. Gleiser and Leon Danon. “Community structure in jazz”. In: *Advances in Complex Systems* 06.04 (Dec. 2003), pp. 565–573.

[187] Linton C Freeman, Cynthia M Webster, and Deirdre M Kirke. “Exploring social structure using dynamic three-dimensional color images”. In: *Social Networks* 20.2 (Apr. 1998), pp. 109–118.

[188] Tore Opsahl and Pietro Panzarasa. “Clustering in weighted networks”. In: *Social Networks* 31.2 (May 2009), pp. 155–163.

[189] Ginestra Bianconi. “The entropy of randomized network ensembles”. In: *EPL (Europhysics Letters)* 81.2 (Jan. 2008), p. 28005.

[190] Ginestra Bianconi. “Entropy of network ensembles”. In: *Physical Review E* 79.3 (Mar. 2009), p. 036114.

[191] Alan C. Hindmarsh et al. “SUNDIALS”. In: *ACM Transactions on Mathematical Software (TOMS)* 31.3 (Sept. 2005), pp. 363–396.

[192] Sean P Cornelius, William L Kath, and Adilson E Motter. “Realistic control of network dynamics”. In: *Nature Communications* 4.1 (Dec. 2013), p. 1942.

[193] Edward Ott, Celso Grebogi, and James A Yorke. “Controlling chaos”. In: *Physical review letters* 64.11 (1990), p. 1196.

[194] Ying-Cheng Lai. “Controlling complex, non-linear dynamical networks”. In: *National Science Review* 1.3 (2014), pp. 339–341.

[195] Bernold Fiedler et al. “Dynamics and Control at Feedback Vertex Sets. I: Informative and Determining Nodes in Regulatory Networks”. In: *Journal of Dynamics and Differential Equations* 25.3 (Sept. 2013), pp. 563–604.

[196] Atsushi Mochizuki et al. “Dynamics and control at feedback vertex sets. II: A faithful monitor to determine the diversity of molecular activities in regulatory networks”. In: *Journal of Theoretical Biology* 335 (Oct. 2013), pp. 130–146.

[197] Jorge Gomez Tejeda Zañudo, Gang Yang, and Réka Albert. “Structure-based control of complex networks with nonlinear dynamics”. In: *Proceedings of the National Academy of Sciences* 114.28 (July 2017), pp. 7234–7239.

[198] Xiang Li, Xiaofan Wang, and Guanrong Chen. “Pinning a Complex Dynamical Network to Its Equilibrium”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 51.10 (Oct. 2004), pp. 2074–2087.

[199] Guanrong Chen. “Pinning control and synchronization on complex dynamical networks”. In: *International Journal of Control, Automation and Systems* 12.2 (Apr. 2014), pp. 221–230.

[200] Francesco Sorrentino et al. “Controllability of complex networks via pinning”. In: *Physical Review E* 75.4 (Apr. 2007), p. 046103.

[201] H. P. Mirsky et al. “A model of the cell-autonomous mammalian circadian clock”. In: *Proceedings of the National Academy of Sciences* 106.27 (July 2009), pp. 11107–11112.

[202] Yoshihiko Susuki, Igor Mezic, and Takashi Hikihara. “Global swing instability of multimachine power systems”. In: *2008 47th IEEE Conference on Decision and Control*. Cancun, Mexico: IEEE, 2008, pp. 2487–2492.

[203] Edward Ott et al. “Scaling behavior of chaotic systems with riddled basins”. In: *Physical Review Letters* 71.25 (Dec. 1993), pp. 4134–4137.

[204] Louis M Pecora et al. “Cluster synchronization and isolated desynchronization in complex networks with symmetries”. In: *Nature communications* 5 (2014).

[205] Abu Bakar Siddique et al. “Symmetry-and input-cluster synchronization in networks”. In: *Physical Review E* 97.4 (2018), p. 42217.

[206] Francesco Sorrentino, Abu Bakar Siddique, and Louis M Pecora. “Symmetries in the time-averaged dynamics of networks: Reducing unnecessary complexity through minimal network models”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29.1 (Jan. 2019), p. 011101.

[207] Caitlin R S Williams et al. “Experimental Observations of Group Synchrony in a System of Chaotic Optoelectronic Oscillators”. In: *Physical Review Letters* 110.6 (Feb. 2013), p. 064104.

[208] Francesco Sorrentino and Louis Pecora. “Approximate cluster synchronization in networks with symmetries and parameter mismatches”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 26.9 (2016), p. 94823.

[209] Adolfo Piperno. “Search space contraction in canonical labeling of graphs”. In: *arXiv* (2011).

[210] Tommi Junttila and Petteri Kaski. “Engineering an Efficient Canonical Labeling Tool for Large and Sparse Graphs”. In: *2007 Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments (ALENEX)*. Philadelphia, PA: Society for Industrial and Applied Mathematics, Jan. 2007, pp. 135–149.

[211] Paul T. Darga et al. “Exploiting structure in symmetry detection for CNF”. In: *Proceedings of the 41st annual conference on Design automation - DAC ’04*. New York, New York, USA: ACM Press, 2004, p. 530.

[212] José Luis López-Presa, Luis Núñez Chiroque, and Antonio Fernández Anta. “Novel Techniques for Automorphism Group Computation”. In: 2013, pp. 296–307.

[213] S A Choudum. “A simple proof of the Erdos-Gallai theorem on graph sequences”. In: *Bulletin of the Australian Mathematical Society* 33.1 (1986), pp. 67–70.

[214] David Gale. “A theorem on flows in networks”. In: *Pacific J. Math* 7.2 (1957), pp. 1073–1082.

[215] Michael T Schaub et al. “Graph partitions and cluster synchronization in networks of oscillators”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 26.9 (2016), p. 94821.

[216] Satoru Kudose. “Equitable Partitions and Orbit Partitions”. In: *Acta Mathematica Sinica* (2009), pp. 1–9.

[217] Wei Ren and Randal W. Beard. *Distributed Consensus in Multi-vehicle Cooperative Control.* Communications and Control Engineering. London: Springer London, 2008.

[218] Anton V Proskurnikov, Alexey S Matveev, and Ming Cao. “Opinion dynamics in social networks with hostile camps: Consensus vs. polarization”. In: *IEEE Transactions on Automatic Control* 61.6 (2016), pp. 1524–1536.

[219] Jianping He et al. “SATS: Secure Average-Consensus-Based Time Synchronization in Wireless Sensor Networks”. In: *IEEE Transactions on Signal Processing* 61.24 (Dec. 2013), pp. 6387–6400.

[220] Peng Lin and Wei Ren. “Constrained Consensus in Unbalanced Networks With Communication Delays”. In: *IEEE Transactions on Automatic Control* 59.3 (Mar. 2014), pp. 775–781.

[221] Junyan Yu and Long Wang. “Group consensus of multi-agent systems with undirected communication graphs”. In: *2009 7th Asian Control Conference*. Hong Kong, China: IEEE, 2009.

[222] Junyan Yu and Long Wang. “Group consensus in multi-agent systems with switching topologies and communication delays”. In: *Systems & Control Letters* 59.6 (June 2010), pp. 340–348.

[223] Junyan Yu and Long Wang. “Group consensus of multi-agent systems with directed information exchange”. In: *International Journal of Systems Science* 43.2 (Feb. 2012), pp. 334–348.

[224] Jiahu Qin and Changbin Yu. “Group consensus of multiple integrator agents under general topology”. In: *52nd IEEE Conference on Decision and Control*. Florence, Italy: IEEE, Dec. 2013, pp. 2752–2757.

[225] Jiahu Qin et al. “On Group Synchronization for Interacting Clusters of Heterogeneous Systems”. In: *IEEE Transactions on Cybernetics* 47.12 (Dec. 2017), pp. 4122–4133.

[226] Jiahu Qin et al. “H infty group consensus for clusters of agents with model uncertainty and external disturbance”. In: *2015 54th IEEE Conference on Decision and Control (CDC)*. Osaka, Japan: IEEE, Dec. 2015, pp. 2841–2846.

[227] Hong-xiang Hu et al. “Reverse Group Consensus of Multi-Agent Systems in the Cooperation-Competition Network”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 63.11 (Nov. 2016), pp. 2036–2047.

[228] Hong-xiang Hu et al. “Couple-group consensus of multi-agent systems in the cooperation-competition network”. In: *2016 35th Chinese Control Conference (CCC)*. Chengdu, China: IEEE, July 2016, pp. 8302–8306.

[229] Yulan Gao et al. “Second-order group consensus in multi-agent systems with time-delays based on second-order neighbours’ information”. In: *2015 34th Chinese Control Conference (CCC)*. Hangzhou, China: IEEE, July 2015, pp. 7493–7498.

[230] Yuanzhen Feng, Shengyuan Xu, and Baoyong Zhang. “Group consensus control for double-integrator dynamic multiagent systems with fixed communication topology”. In: *International Journal of Robust and Nonlinear Control* 24.3 (Feb. 2014), pp. 532–547.

[231] Giovanni Russo and Jean-Jacques E. Slotine. “Symmetries, stability, and control in nonlinear systems and networks”. In: *Physical Review E* 84.4 (Oct. 2011), p. 041929.

[232] Giovanni Russo, Mario di Bernardo, and Jean-Jacques E. Slotine. "Convergence and cluster synchronization in networks of discrete-time and asynchronous systems". In: *52nd IEEE Conference on Decision and Control*. Florence, Italy: IEEE, Dec. 2013, pp. 5921–5926.

[233] Richard P. Stanley. *Enumerative Combinatorics*. Cambridge: Cambridge University Press, 2011.

[234] Marina Meilă. "Comparing clusterings-an information based distance". In: *Journal of Multivariate Analysis* (2007).

[235] Milton Abramowitz and Irene A Stegun. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*. Vol. 55. Courier Corporation, 1965.

[236] Jordi Duch and Alex Arenas. "Community detection in complex networks using extremal optimization". In: *Physical review E* 72.2 (2005), p. 27104.

[237] K.-I. Goh et al. "The human disease network". In: *Proceedings of the National Academy of Sciences* 104.21 (May 2007), pp. 8685–8690. arXiv: `NIHMS150003`.

[238] Wayne W. Zachary. "An Information Flow Model for Conflict and Fission in Small Groups". In: *Journal of Anthropological Research* 33.4 (Dec. 1977), pp. 452–473. arXiv: `NIHMS150003`.

[239] Ara Cho et al. "WormNet v3: a network-assisted hypothesis-generating server for Caenorhabditis elegans". In: *Nucleic Acids Research* 42.W1 (July 2014), W76–W82.

[240] Katrin Amunts et al. "BigBrain: An Ultrahigh-Resolution 3D Human Brain Model". In: *Science* 340.6139 (June 2013), pp. 1472–1475. arXiv: `NIHMS150003`.

[241] Mark E J Newman. "Finding community structure in networks using the eigenvectors of matrices". In: *Physical Review E* 74.3 (Sept. 2006), p. 036104.

[242] *Digital Library of Mathematical Functions*. 2019.

[243] I. S. Gradshteyn and I. M. Ryzhik. *Table of integrals, series, and products*. 4th ed. Academic press, inc, 2014.

[244] Shigetoshi Katsura and Sakari Inawashiro. "Lattice Green's Functions for the Rectangular and the Square Lattices at Arbitrary Points". In: *Journal of Mathematical Physics* 12.8 (Aug. 1971), pp. 1622–1630.

[245] Tohru Morita. "Useful Procedure for Computing the Lattice Green's Function-Square, Tetragonal, and bcc Lattices". In: *Journal of mathematical physics* 12.8 (1971), pp. 1744–1747.

[246] Mona Berciu. "On computing the square lattice Green's function without any integrations". In: *Journal of Physics A: Mathematical and Theoretical* 42.39 (2009), p. 395207.

[247] Chiara Dalla Man, Robert A. Rizza, and Claudio Cobelli. "Meal Simulation Model of the Glucose-Insulin System". In: *IEEE Transactions on Biomedical Engineering* 54.10 (Oct. 2007), pp. 1740–1749.

[248] Chiara Dalla Man et al. "GIM, Simulation Software of Meal Glucose—Insulin Model". In: *Journal of Diabetes Science and Technology* 1.3 (May 2007), pp. 323–330.

[249] Henry Wadsworth Gould. *Combinatorial Identities: A standardized set of tables listing 500 binomial coefficient summations.* Morgantown, W Va, 1972.

[250] Harry Bateman. *Tables of Integral Transforms Volume I.* Ed. by Arthur Erdélyi. McGraw-Hill Book Company, Inc, 1954.