# Distributed Control of Large-Scale Networks: From Theory to Applications

Siavash Alemzadeh

A dissertation submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

University of Washington

2020

Reading Committee:

Mehran Mesbahi, Chair

Behcet Acikmese

Steven Brunton

Marco Salviato

Program Authorized to Offer Degree:
Aeronautics and Astronautics

University of Washington

**Abstract**

Distributed Control of Large-Scale Networks:
From Theory to Applications

Siavash Alemzadeh

Chair of the Supervisory Committee:
Professor Mehran Mesbahi
William E. Boeing Department of Aeronautics and Astronautics

Networked systems analysis has been on the forefront of interdisciplinary research over the past few years with applications spanning from a wide variety of research literature, ranging across control systems, learning, power grids, social network, and economics. Control of large-scale networked systems often necessitates the availability of complex models for the interactions amongst the agents. This dissertation tackles some of the aspects of control and learning in such complex interconnected systems from both theoretic and applied viewpoints. We consider networks as "systems" and study the advantage they bring into control, estimation, and optimization of various applications such as antagonistic networks, network games, and layered networks. Next, we generalize our viewpoint towards networks as a "system-of-systems" where each system contains a subsystem and consider the distributed control problem on such hierarchy. This part considers methods for synthesis of linear quadratic controllers for large-scale systems that consist of interactive agents. Lastly, we put forward ideas on online iterative synthesis when the control horizon is limited or data collection is expensive. This part suggests potential remedy to some shortcomings that are incurred due to potential faulty assumptions in the preceding chapters of the thesis. An underlying theme to most parts of the manuscript is how well a network's structure facilitates the machinery of control and optimization of systems that are considered in this category. The functionality of methods in each part is demonstrated via illustrative simulations.

## TABLE OF CONTENTS

Page

## LIST OF FIGURES

## DEDICATION

To my parents,

*Saeed* and *Fariba*

who relentlessly strived for my success

and encouraged me for excellence

## ACKNOWLEDGMENTS

I would like to first thank Mehran Mesbahi for being such a caring adviser, mentor, and instructor. I appreciate his vigilant support, patience, and encouragement throughout my Ph.D. and giving me the freedom to learn from my mistakes while setting high the standards. His support made it possible for me to publish, attend, and present at several conferences and symposiums, meet the best in the community, and feel the joy of collaborative research. I would also like to thank my Ph.D. dissertation committee members Profs. *Behcet Acikmese, Steven Brunton, Marco Salviato* and *Selim Tuncel* for their insightful comments on this dissertation, which helped enormously in my growth as a student in control theory and machine learning.

I am incredibly grateful to be a part of the Robotics, Aerospace, Information, and Networks (RAIN) research lab. I especially would like to thank Mathias Hudoba de Badyn, Shahriar Talebi, Niyousha Rahimi, Afshin Mesbahi, Dillon Foight, Taylor Reynolds, Jingjing Bu, Bijan Barzgaran, Mengyuan Wang, Dian-Jing Chen, Aditya Deole, Tom Miesen, Unsik Lee, Saghar Hosseini, Marzieh Nabi, Airlie Chapman, and Eric Schoof for their heartwarming presence, insightful discussions, and their uplifting characteristics. Special thanks to Shahriar Talebi for being such an inspiring research fellow and friend. My research got much more assorted with him joining the lab leading to all of our joint work on distributed control, data-guided iterative control, and game networks. I also appreciate, Mathias Hudoba de Badyn, my first collaborator, for all the fantastic fruitful time we spent working on signed networks. His creativity and upbeat personality has been a source of inspiration to me. I would also like to acknowledge my collaborators *Afshin Mesbahi, Niyousha Rahimi,* and *Lillian Ratliff* at the University of Washington, *Leonardo Duenas-Osorio* and *Hesam Talebiyan* at Rice University and *Atiye Alaeddini*, from whom I learned a lot. I further thank *Ramin Moslemi* and *Ratnesh Sharma* at NEC Laboratories America Inc., and Nawid

Jamali at Honda Research Institute for their guidance during my internships.

I would also like to express my sincere gratitude to staff at The University of Washington, *Wanwisa Kisalang*, the Mechanical Engineering Graduate Student Adviser for facilitating my transition to the grad school and also *Ed Connery* and *Danyel Hacker* at the William E. Boeing Department of Aeronautics and Astronautics for their kind support and helping me to get better adapted to the life as a grad student. I also should thank Ricardo Hidalgo at the Hall Health mental clinic for giving me all the positive vibes during the most difficult times in my academic journey.

Last but not least, I would like to thank my dear parents, *Saeed* and *Fariba*, and my brother *Sepehr* for their unconditional support and consistent encouragement throughout my life and giving me strength to reach for the stars and chase my dreams. Also, I wholeheartedly thank my partner in crime, *Behnoosh*, who stood by me through all my travails and impatience and for all her absolute love and support.

# Chapter 1
# INTRODUCTION

## *1.1 Distributed Control*

Today's technology has faced unprecedented amounts of hardware, volumes of data, lines of source code, and number of end users. The unexpected scale of these systems and their interconnections gives rise to many problems. For example, while such complex systems keep evolving continuously, human or system failures on both software and hardware will gradually become a norm. This revolution, in the same time, has resulted in the development of even more effective computational tools that utilize the data generated by the system to reason about reduced order representations, subsequently utilized for classification or prediction on the underlying model of the system. Such techniques have been particularly useful when the derivation of models from first principles is restrictively complex or infeasible, and hence, it seems natural to leverage data-driven methods for high-dimensional complex systems. Nevertheless, when the scale of the problems grows even higher, the possibility of design and analysis boils down to quantifying the data based on potential underlying structures of the system.

After hitting some capacity threshold, large-scale systems and control require an infrastructure that permits distributed resources to work in parallel to develop, deploy, and evolve system components. This research on this area investigates integration of designs and platforms that support decentralized nature of the applications with a concentration on technologies, methods, and theories that will enable them to be developed in coordination with other parts of the same system. Such distributed nature could be embedded within applications in a natural or manual manner. For instance, humans, by nature, band together and form social groups which, in turn, becomes only a part of a possibly very large social network. Or say, hand-crafted interconnected systems of automatic drones that work collectively for a common human-unreachable goal such as putting out a blaze in forests or

satellites collecting information from the orbits is an example of hand-crafted engineering designs that easily leverage the sparsity within the system to communicate signals efficiently.

In the meantime, the emergence of the knowledge, cheap measurement tools, and the capability of local communications have raised a number of new system-level questions concerning how such systems should be coordinated and controlled. Now, one of the main challenges facing this new field of research is how should the interaction and control protocols be structured to ensure that a given task is achieved by the system? Furthermore, even if the solutions are viable for rudimentary systems, when the structural complexity increases, how do the traditional solutions address the new demands? Or how can data-guided system theoretic methods be utilized to connect the observations from data directly to the proposed solutions? The goal of this dissertation is to investigate a general framework for studying interactions of dynamic agents on networks–how the structure of the network affects the dynamics and behavior of the systems, processes, and information propagating over the networks.

### *1.2 Why Networks?*

Networked systems analysis has been on the cutting edge of multi-disciplinary research over the past few years with applications spanning from robotic swarms to biological networks. Some well-studied examples of networked systems are social networks and dynamics of opinions [81], flocking [165], autonomous robotics [152], quantum networks [1], autonomous flight [60], traffic control [175] and gene networks [176]. Often in practice, a real system is comprised of a number of components that are linked together—naturally or manually—to form a functioning unit. The dynamics for each separate component could be known or unknown and a connection between two of these units implies that the output of one component relies on the information it receives from its neighbors. We call such a system a *networked dynamical system* and those components are referred to as *agents* or *subsystems*. The structure of such interconnections can be represented in a communication graph, where the nodes correspond to the agents and the edges refer to the connections or edges among those agents. Graphs provide natural abstractions for how information is shared between agents in a network. Also, common to many of the networked systems, a global objective is

achieved based on local interactions—which in turn—require local decision-making that is inherently restricted by limited information exchange and prescribed set of admissible policies. As such, many computationally efficient optimization algorithms when implemented on a network have scaling issues when the network size grows. Not surprisingly, there has been an extensive literature on exploiting the structure of a information-exchange graph in order to reduce the complexity of distributed methods built upon different optimization and control methods. In the meantime, while these works mainly focus on a cooperative information sharing, many of the real-world systems exhibit non-cooperative behaviors due to a plethora of reasons including intrusions/attacks, greedy agents, competition for constrained resources, misaligned incentives, or the adversarial nature of environments that necessitate a game-theoretical model, for instance. Moreover, from a practical perspective, employment of graph models in real-world applications is becoming inevitable. From telecommunication, biology, data mining to inferential modeling and robotics, graph-based network applications are ubiquitous in today's technological world. Hence, we also use graph theory as a well-established tool to model network systems and apply it in different examples that we study in this dissertation. Some basic standard definitions of graphs used in the text are provided in Section 1.4.

### *1.3 Overview and Contributions of the Thesis*

In this thesis, we address a variety of questions that emerge naturally when facing network control design. In particular, we go over three different types of question that will be addressed in three parts.

#### Part I: Networks as Systems

In this part, we consider networks as systems with input and output where a collection of nodes in the network assume control and sensing roles, whereas the remaining nodes of the graph execute a local, agreement-like protocol. We call this protocol consensus which will be the cornerstone of our analysis throughout this part. On each chapter we tackle a shortcoming or application with the aim of identifying graph theoretic implications for the

system theoretic properties of such networked systems.

First, we focus on antagonistic network and show how the symmetry structure of a network with a single control node directly relates to the controllability and observability of the corresponding input-output system. We also take one further step by introducing network equitable partitions as means by which such controllability and observability characterizations can be extended to networks with multiple inputs and outputs. We will discuss that the key feature of structurally balanced graphs that allows this analysis is that they admit a gauge transformation that allows the permutation matrix corresponding to the graph symmetry be extended to a signed permutation matrix.

Second, we turn our attention to application of network games where we use a dual-averaging optimization approach to propose a distributed no-regret learning algorithm in network games. Specifically, with only locally available observations, we consider the scenario where each player optimizes a global objective, formed by local objective functions on the nodes of a given communication graph.

Lastly, we study the guaranteed-cost control design problem of layered network, constructed by graph products of several subsystems. The idea behind this work is that model uncertainty becomes more pronounced for large-scale networks and for certain classes of systems, the layering structure allows a compositional approach. Hence, we present such an approach to determine performance guarantees on layered networks with inherent model uncertainties.

**Part II: Networks as a System-of-Systems**

In the second part of the thesis, not only we assume that networks form a system, but the allow agents to run their own dynamical systems that continuously exchange information with neighboring systems. The distributed control schemes are presented mainly in this part of the thesis. Besides, in this part we assume that decisions are going to be made directly from data and there is no knowledge of the underlying model that generates such data. To that end, we propose a distributed $Q$-learning algorithm to design a feedback mechanism. In the first section of this part, we introduce a decentralized control scheme

where each agents tries to optimize it own objective while interacting with other agents in the network. This interaction is shown to be available due to some knowledge of neighbors' objectives.

The second section of this part contrasts decentralized control to distributed control ideas where instead of following a consensus-based methodology, we propose an algorithm that introduces a structured controller where this sparsity structure comes from the underlying graph structure parameterizing the agents'communication. To ensure stability and convergence, our algorithm requires temporary "auxiliary" links to boost information exchange of a small portion of the graph during the learning phase.

**Part III: Iterative Data-Driven Control**

In the last technical part of the dissertation, we shift focus from core network control design and analysis to some more general aspects of data-driven iterative control from a non-asymptotic viewpoint. Meanwhile, we address some of the drawbacks that we face in the prior chapters of the thesis. First, we focus on linear model regression on time-series data that are collected from a linear dynamical system where we focus on modal properties of the underlying dynamics. We provide deterministic error bounds for fitting a linear model to observed time-series data with a particular attention to the role of symmetry and eigenvalue multiplicity in the underlying system matrix.

In the second section of this part, we introduce a similar setup that tackles the online control of a linear system from a different angle. Specifically, we examine online regulation of (possibly unstable) partially unknown linear systems with no *a priori* assumptions on the initial controller—which is a standard assumption in most of the iterative control literature. we introduce and characterize the notion of "regularizability" for linear systems that gauges the capacity of a system to be regulated in finite-time in contrast to its asymptotic behavior (commonly characterized by stabilizability/controllability). Next, having access only to the input matrix, we propose the DGR synthesis that—as its name suggests—regulates the underlying states while also generating informative data that can subsequently be used for data-driven stabilization or system identification.

## *1.4 Background and Notations*

Here we will go over some of the common mathematical background and notations that the reader might encounter throughout the text.

### *1.4.1 Linear Algebra*

We denote by $\mathbb{R}$ and $\mathbb{N}$ the set of real and natural numbers respectively, $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$, and $\mathbb{S}$ refers to the set of symmetric matrices. A column vector with $n$ elements is referred to as $\boldsymbol{v} \in \mathbb{R}^n$, where $\boldsymbol{v}_i$ represents the $i$th element in $\boldsymbol{v}$. The matrix $M \in \mathbb{R}^{p\times q}$ contains $p$ rows and $q$ columns with $M_{ij}$ denoting the element in the $i$th row and $j$th column of $M$. The square matrix $N \in \mathbb{S}^{n\times n}$ is *symmetric* if $N^\top = N$, where $N^\top$ denotes the *transpose* of the matrix $N$. The operator *diag(.)* makes a square diagonal matrix out of the elements of its argument and the operator *vech*(.) takes a square matrix and stacks the upper right triangular half (including the diagonal) into a single vector. The zero and identity matrices are denoted by $\mathbf{0}$ and $\mathrm{I}_n$. The unit vector $e_i$ is the column vector with all zero entries except $[e_i]_i = 1$. We let $N \succ 0$ ($\succeq 0$) when $N$ is a positive-(semi)definite matrix, i.e., $\boldsymbol{x}^\top N\boldsymbol{x} > 0$ ($\geq 0$) for all $\boldsymbol{x} \neq 0$. A matrix is positive-definite if and only if all of its leading principle minors are positive. The $i$th eigenvalue and spectral radius of $M$ are denoted by $\lambda_i(M)$ and $\rho(M)$ and $M$ is Schur stable if $\rho(M) < 1$. To simplify the vector notation, we use semicolon (;) to concatenate column vectors, *e.g.*, $[\boldsymbol{v}^\top\ \boldsymbol{w}^\top]^\top = [\boldsymbol{v};\ \boldsymbol{w}]$. For a block matrix $\mathbf{F}$, by $[\mathbf{F}]_{rk}$ we imply the $r$th row and $k$th column "block" component with appropriate dimensions. We denote by $\mathbb{I}_{\mathcal{E}}$ the *indicator function* of event $\mathcal{E}$ defined as $\mathbb{I}_{\mathcal{E}} = 1$ if $\mathcal{E}$ holds and $\mathbb{I}_{\mathcal{E}} = 0$ otherwise. The *cardinality* of a set $S$ is denoted as $|S|$. The *annihilator* of a set $S$ is defined as,

$$S^\perp = \{v^* \in \mathbb{R}^n : \langle v, v^*\rangle = 0 \text{ for all } v \in S\},$$

where $\langle\cdot,\cdot\rangle$ is the inner product in the Euclidean space. The column space and null space of a matrix $M$ are denoted by $\mathcal{R}(M)$ and $\mathcal{N}(M)$ respectively. We define $\mathcal{R}(P)$ to be $A$-*invariant* if there exists $C$ such that $AP = PC$. A doubly stochastic matrix $P \in \mathbb{R}^{n\times n}$ is defined as a non-negative square matrix such that $\sum_j P_{ij} = \sum_j P_{jk} = 1$ for all $i$ and $k$, and $\sigma_2(P)$

indicates the second largest singular value of $P$. We define $[n] = \{1, \ldots, n\}$. The *Euclidean norm* of a vector $x \in \mathbb{R}^n$ is defined as $\|x\| = (x^\top x)^{1/2} = (\sum_{i=1}^n x_i^2)^{1/2}$ and the *dual norm* to $\|x\|$ is denoted by $\|x\|_* := \sup_{\|u\|=1} \langle x, u \rangle$. Also, the *1-norm* is defined as $\|x\|_1 = \sum_{i=1}^n |x_i|$. A function $f$ is convex if $f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y)$ for all $\theta \in (0,1)$ and for all $x, y$ in its convex domain, and $g$ is a subgradient of $f$ at point $z$ if $f(y) \geq f(z) + g^\top (y - z)$ for all $y$. If $f$ is convex and differentiable, the gradient of $f$, $\nabla f(x)$, is also a subgradient of $f$ at $x$. The set of all subgradients of $f$ at $x$ is called subdifferential and denoted by $\partial f(x)$.

### *1.4.2 Graph Theory*

A network system with $n$ agents is characterized by a *graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of edges, and $W \in \mathbb{R}^{n \times n}$ consists of weights assigned to edges. We call $\mathcal{A} \in \mathbb{R}^{n \times n}$ the *adjacency matrix* where $\mathcal{A}_{ij} = W_{ij} \neq 0$. The *degree matrix* $D \in \mathbb{R}^{n \times n}$ is a square diagonal matrix where $D_{ii} = \sum_{j \in \mathcal{N}(i)} \mathcal{A}_{ij}$. The *graph Laplacian* is then defined as $\mathcal{L} = \mathcal{D} - \mathcal{A}$. Then the continuous-time *consensus dynamics* is defined as $\dot{x} = -\mathcal{L}x$ with $x \in \mathbb{R}^n$ the state vector. A *path* of length $r$ in $\mathcal{G}$ is given by a sequence of different nodes $v_{i_0}, v_{i_1}, \ldots, v_{i_r}$ such that for $k = 0, 1, \ldots, r-1$, the nodes $v_{i_k}$ and $v_{i_{k+1}}$ are neighbors. When the terminal nodes are equal, the path is called a *cycle*. The reader is referred to [116] for more details on system theoretic-related notion of graphs.

### *1.4.3 Graph Products*

Cartesian product is an effective way to synthesize large-scale networks from smaller graphs, which we refer to as *factors* [88]. The Cartesian product of $m$ graphs $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i, \mathcal{W}_i)$ for $i = 1, 2, \ldots, m$ is denoted by $\mathcal{G} = \square_{i=1}^m \mathcal{G}_i$. The vertex set of $\mathcal{G}$ has the form $\mathcal{V}_\mathcal{G} = \mathcal{V}_1 \times \mathcal{V}_2 \times \cdots \times \mathcal{V}_m$ and the nodes $P_v = (v_1, v_2, \ldots, v_m)$ and $P_u = (u_1, u_2, \ldots, u_m)$ are connected if and only if there exists some $i$ such that $(v_i, u_i) \in \mathcal{E}_i$ and $v_j = u_j$ for $j \neq i$. The Kronecker product of $A \in \mathbb{R}^{p_1 \times q_1}$ and $B \in \mathbb{R}^{p_2 \times q_2}$, denoted by $A \otimes B$. The Kronecker product is not commutative as $A \otimes B \neq B \otimes A$ but rather permutation equivalent, i.e., there exist $P$ and $Q$ such that $A \otimes B = P(B \otimes A)Q$. Other important properties of the Kronecker product include *mixed-product*, $(A \otimes B)(C \otimes D) = AC \otimes BD$, *distributivity*,

$A \otimes (B + C) = (A \otimes B) + (A \otimes C)$, and *associativity*, $A \otimes (B \otimes C) = (A \otimes B) \otimes C$. Moreover, $\left(\bigotimes_{i=1}^{m} R_i\right)^\top = \bigotimes_{i=1}^{m} R_i^\top$ and $\left(\bigotimes_{i=1}^{m} T_i\right)^{-1} = \bigotimes_{i=1}^{m} T_i^{-1}$ if $T_i \in \mathbb{R}^{n\times n}$ is invertible for all $i \in \{1, 2, \ldots, m\}$. The Kronecker sum is defined on square matrices $M \in \mathbb{R}^{m\times m}$ and $N \in \mathbb{R}^{n\times n}$ as $M \oplus N = M \otimes I_n + I_m \otimes N$. For $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$, if $(\lambda_i, u_i)$ and $(\mu_j, v_j)$ are eigenvalue-eigenvector pairs of $M$ and $N$ respectively, then $(\lambda_i\mu_j, u_i \otimes v_j)$ are eigenvalue-eigenvector pairs of $M \otimes N$. This also implies that the Kronecker product preserves positive definiteness [84].

### *1.4.4 Linear Quadratic Regulator*

Linear Quadratic Regulator (LQR) is a classical method in optimal control theory where the states evolve based on a linear equation that correlates the states and control and the evaluation is based on a quadratic index measure. Assume the dynamics follows,

$$x(k+1) = Ax(k) + Bu(k), \qquad x(0) = x_0$$

in a discrete-time setup. The dynamics can also be deemed as continuous-time in the form of,

$$\dot{x} = Ax + Bu, \qquad x(0) = x_0.$$

Here, $x \in \mathbb{R}^n$ is the state vector of the system with the given initial state $x_0$, and $u \in \mathbb{R}^m$ is the control signal used to regulate the system. The pair $(A, B)$ are the system parameters that dictate the dynamics. We call the pair $(A, B)$ *controllable*, if and only if the controllability matrix $\mathcal{C} = [B \quad AB \quad \ldots \quad A^{n-1}B]$ has full-rank, where $n$ is the size of the system. We assume that this pair is fixed for every analysis presented in this thesis. The objective is to find $u$ that drives the state from $x_0$ towards zero in an optimal way. In order to quantify optimality a quadratic cost is specified as,

$$J(x, u) = \sum_{k=0}^{\infty} x(k)^\top Q x(k) + u(k)^\top R u(k),$$

where $Q$ and $R$ are the penalty parameters on the states and control signals respectively. Within the framework of LQR, this cost is minimized subject to the linear dynamics introduced above. The solution to the discrete-time LQR is shown to rely on the solution to the

discrete Algebraic Riccati Equation (ARE),

$$P = A^\top PA + Q - A^\top PB \left(R + B^\top PB\right)^{-1} B^\top PA.$$

Then the optimal control law in the discrete-time case is linear and defined by,

$$u^*(k) = -\left(R + B^\top PB\right)^{-1} B^\top PAx(k) = -Kx(k).$$

The policy derived above is also referred to as linear feedback control. For any stabilizing linear control law $u(k) = -Kx(k)$, the value of the cost can be computed by finding the solution to the discrete-time Lyapunov equation,

$$P = (A - BK)^\top P(A - BK) + Q + K^\top RK,$$

whose solution can be expressed as an infinite sum,

$$P = \sum_{k=0}^{\infty} \left((A - BK)^\top\right)^k Q(A - BK)^k,$$

where $A_{\text{cl}} = A - BK$ is referred to as the closed-loop system from $x(k+1) = (A - BK)x(k)$. Note that depending on how the policy is updated ($u = -Kx$ or $u = Kx$) we switch the closed-loop system to $A_{\text{cl}} = A + BK$. Also, when the linear quadratic system as described above follows the law $u = -Kx$, then the cost function can be determined as,

$$J(x, u) = x_0^\top Px_0.$$

### *1.4.5 Nonlinear Dynamical Systems*

We follow the same conventions as in [3]. Consider the controlled dynamical system

$$\dot{x} = F(x, u) \tag{1.1}$$

where $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is a smooth mapping. The *accessible set* $\mathcal{A}(x_0, T)$ of the system (1.1) from $x_0$ at time $T$ is the set of all end-points $\theta(T)$ where $\theta : [0, T] \to \mathbb{R}^n$ is a trajectory of (1.1). The accessible set of (1.1) from $x_0$ up to $T$ is defined as $\mathcal{A}(x_0, \leq T) \coloneqq \cup_{0 \leq \tau \leq T} \mathcal{A}(x_0, \tau)$. Then, the nonlinear dynamical system (1.1) is said to be *accessible* from the initial point $x_0$ if for every $T > 0$ the set $\mathcal{A}(x_0, \leq T)$ contains a non-empty interior.

### *1.4.6 Koopman Operator*

Consider the dynamical system in (1.1) without control. The *Koopman operator* $\mathcal{K}$ acts on functions of state space (called *observables*) $\psi$ by the action $\mathcal{K}\psi = \psi \circ F$. The function $\varphi(x)$ is an *eigenfunction* of $\mathcal{K}$ corresponding to the *eigenvalue* $\mu$ if $\mathcal{K}\varphi(x) = e^{\mu t}\varphi(x)$. For an observable function $g$ in the span of Koopman eigenfunctions, one can write $g(x) = \sum_{k=1}^{\infty} v_k \varphi_k(x)$, where the $v_k$'s are the *Koopman modes*. For the case of full-state observable $g(x) = x$ the states can be reconstructed as, $x(t) = g(x(t)) = \sum_{k=1}^{\infty} e^{\mu_k t}\varphi_k(x(0))v_k$, where we refer to $\{\mu_k, \varphi_k, v_k\}$ as the *Koopman triple*. The Koopman operator is linear, but can be infinite-dimensional. To make this operator computationally usable, a numerical approximation of Koopman operator can be obtained by EDMD, resulting in a finite-dimensional approximation $K$ [171].

## *1.5 Included Publications*

Some of the ideas and figures have appeared previously in the following publications:

[8] Siavash Alemzadeh, Mathias Hudoba de Badyn, and Mehran Mesbahi. "Controllability and stabilizability analysis of signed consensus networks." 2017 IEEE Conference on Control Technology and Applications (CCTA). IEEE, 2017.

[46] Mathias Hudoba de Badyn, Siavash Alemzadeh, and Mehran Mesbahi. "Controllability and data-driven identification of bipartite consensus on nonlinear signed networks." 2017 IEEE 56th Annual Conference on Decision and Control (CDC). IEEE, 2017.

[10] Siavash Alemzadeh, and Mehran Mesbahi. "Influence models on layered uncertain networks: A guaranteed-cost design perspective." 2018 IEEE Conference on Decision and Control (CDC). IEEE, 2018.

[7] Atiye Alaeddini, Siavash Alemzadeh, Afshin Mesbahi, and Mehran Mesbahi. "Linear model regression on time-series data: non-asymptotic error bounds and applications." 2018

IEEE Conference on Decision and Control (CDC). IEEE, 2018.

[9] Siavash Alemzadeh, and Mehran Mesbahi. "Distributed $Q$-learning for dynamically decoupled systems." 2019 American Control Conference (ACC). IEEE, 2019.

[157] Shahriar Talebi, Siavash Alemzadeh, Lillian J. Ratliff, and Mehran Mesbahi. "Distributed Learning in Network Games: a Dual Averaging Approach." 2019 IEEE 58th Conference on Decision and Control (CDC). IEEE, 2019.

[11] Siavash Alemzadeh, Hesam Talebiyan, Shahriar Talebi, Leonardo Duenas-Osorio, and Mehran Mesbahi. "Resource Allocation for Infrastructure Resilience using Artificial Neural Networks." *In Proc. of 32th International Conference on Tools with Artificial Intelligence* (ICTAI), 2020.

[155] Shahriar Talebi, Siavash Alemzadeh, Niyousha Rahimi, and Mehran Mesbahi. "On Regularizability and its Application to Online Control of Unstable LTI Systems." (2020).

# Chapter 2

# NETWORKS AS SYSTEMS

In this chapter, we tackle a few applications on networks with the assumption that the network is basically an input-output system. Although, in principle, one can designate network inputs and outputs at distinct nodes, we will be primarily concerned with the situation when the input and output nodes are identical. Having considered the system in this way, we focus on three applications of networks: controllability of antagonistic networks, distributed learning in network games, guaranteed-cost control of layered networks.

## *2.1 Mathematical Preliminaries*

### *2.1.1 Consensus Protocol*

Consensus (or agreement) is one of the fundamental problems in multiagent coordination. This protocol refers to the state of the system where a collection of agents are to agree on a joint state value. In an $n$ agent dynamic setup, the consensus protocol states that the rate of change of each unit's state is assumed to be governed by the (weighted) sum of its relative states with respect to a subset of its neighbors in the network. Let $x_i \in \mathbb{R}$ denote the scalar state of agent $i$, then

$$\dot{x}_i(t) = \sum_{j \in \mathcal{N}_i} \left(x_j(t) - x_i(t)\right), \qquad i = 1, \cdots, n$$

and putting this into the matrix formation,

$$\dot{x}(t) = \mathcal{L}(\mathcal{G})x(t),$$

where $x$ now collect the entirety of states and the interconnections are encoded with the Laplacian matrix $\mathcal{L}(\mathcal{G})$. In simple terms, each agent in the network averages the value of the parameter that they hold with that of their neighbours.

### 2.1.2 Network Partitioning

We fix the input (and output) nodes in the network and denote them by subscript $i$, whereas the other agents in the network, the floating nodes shown with $f$, continue to abide by the ambient agreement protocol. The graphs containing this parts of the network are respectively denoted by $\mathcal{G}_i$ and $\mathcal{G}_f$, defined after removing the input/output nodes as well as the edges in between. The partitioning implies that the Laplacian can be form into,

$$\mathcal{L}(\mathcal{G}) = \begin{bmatrix} A_f & B_f \\ B_f^\top & A_i \end{bmatrix}.$$

Based on this partitioning of the node, the resulting system becomes a standard linear time-invariant (LTI) system. We thereby study the agreement within this framework in this context when the floating nodes evolve as,

$$\begin{aligned} \dot{x}_f(t) &= -A_f x_f(t) - B_f u(t), \\ y(t) &= -B_f^\top x_f(t), \end{aligned}$$

where $u$ denotes the (exogenous) control signal injected at the input nodes.

### 2.1.3 Signed graphs

A *signed graph* $\mathcal{G}_s$ is a graph that admits negative weights. We define the *signed graph Laplacian* as $\mathcal{L}_s = \mathcal{D}_s - \mathcal{A}_s$, where $\mathcal{A}_s$ can also contain negative element and the degree matrix is given by $\mathcal{D}_{ii} = \sum_{j\in\mathcal{N}(i)} |\mathcal{A}_{ij}| = \sum_{j\in\mathcal{N}(i)} |\mathcal{W}_{ij}|$. The generalization of consensus dynamics to signed graphs was introduced in [13], and the corresponding dynamical system is given by $\dot{x} = -\mathcal{L}_s x$, i.e.,

$$\dot{x}_i = -\sum_{j\in\mathcal{N}(i)} |\mathcal{W}_{ij}| \left(x_i - \operatorname{sgn}(\mathcal{W}_{ij}) x_j\right), \tag{2.1}$$

where sgn represents the sign function.

### 2.1.4 Automorphisms/Interlacing/Equitable Partitions

An *automorphism* of the graph $\mathcal{G}$ is a permutation $\psi$ of its nodes such that $\psi(i)\psi(j) \in \mathcal{E}$ if and only if $ij \in \mathcal{E}$. Let the permutation matrix $\Psi$ be such that $[\Psi]_{ij} = 1$ if $\psi(i) = j$ and

zero otherwise. Then $\psi$ is an automorphism of $\mathcal{G}$ if and only if $\Psi A(\mathcal{G}) = A(\mathcal{G})\Psi$ (see [116]). Suppose $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{m\times m}$ are both symmetric and $m \leq n$. Then the eigenvalues of $B$ *interlace* the eigenvalues of $A$ if for $i = 1, 2, \ldots, m$, $\lambda_{n-m+i}(A) \leq \lambda_i(B) \leq \lambda_i(A)$ where $\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_n(A)$ are the eigenvalues of $A$ in a non-increasing order [73]. The *cell* $C$ is a subset of the graph nodes $\mathcal{V}$. A *nontrivial cell* is a cell with more than one node. A *partition* is a grouping of $\mathcal{V}$ into different cells. An *r-partition* $\pi$ of $\mathcal{V}$ with cells $\{C_i\}_{i=1}^r$ is *equitable* if each node in $C_j$ has the same number of neighbors in $C_i$, for all $i, j$. We call $\pi$ a Nontrivial Equitable Partition (NEP) if it contains at least one nontrivial cell. Let $b_{ij}$ be the number of neighbors in $C_j$ of a node in $C_i$. The *quotient* of $\mathcal{G}$ over $\pi$, denoted by $\mathcal{G}/\pi$, is the directed graph with the cells of an equitable *r-partition* $\pi$ as its nodes and $b_{ij}$ edges directed from $C_i$ to $C_j$. The adjacency matrix of the quotient is specified by $[A(\mathcal{G}/\pi)]_{ij} = b_{ij}$. A *characteristic vector* $p_i \in \mathbb{R}^n$ of a nontrivial cell $C_i$ has 1's in components associated with $C_i$ and 0's elsewhere. A *characteristic matrix* $P \in \mathbb{R}^{n\times r}$ of a partition $\pi$ of $\mathcal{V}$ is defined as $[p_i]_{i=1}^r$.

### *2.2 Signed Consensus Networks*

Consensus algorithms on networks with antagonistic interactions were first considered by Altafini [13,14]. The network property of *structural balance*, first considered in the study of social networks ([4,34]) was identified in Altafini's works as the property inducing bipartite consensus in which the agents converge to two disjoint clusters instead of a uniform consensus. Graph-theoretic properties of signed Laplacian dynamics were studied by Pan *et al.* [129]. Further research by Pan *et al.* has looked at identifying the bipartite structure of structurally balanced graphs using data from signed Laplacian dynamics and dynamic mode decomposition Dynamic Mode Decomposition (DMD) [130], adding to the works done by Harary and Kabell [79] and Facchetti *et al* [61]. Recent contributions by Clark *et al.* have studied the leader selection problem in signed consensus [41].

On the other hand, the generalization to nonlinear consensus algorithms has been studied in numerous settings. Behaviour of nonlinear consensus protocols was considered by Srivastava *et al.* [149]. The extension of these consensus protocols to signed networks was studied by Altafini [14]. Moreover, the generalization of symmetry arguments for control-

lability was first looked at by Aguilar and Gharesifard [3]. There has been a recent interest in applying data-driven methods to control of networks, such as the Koopman operator. The Koopman operator is a dynamical framework in which one considers the propagation of observables of the state, rather than the state itself. The Koopman operator is linear, even for a nonlinear system, but the trade-off is that the observable space might be infinite dimensional [31]. This formalism lends itself well to a data-driven approach, allowing one to approximate the Koopman operator by collecting data [171]. Research by Pan *et al.* has looked at identifying the bipartite structure of signed networks using data-driven methods [130], furthering work done by Facchetti *et al* [61], and Harary and Kabell [79].

The contributions of this part is two-folds. First, we conduct a controllability analysis of signed Laplacian consensus using symmetry arguments developed by Rahmani *et al.* [135], for the individual Single-Input-Single-Output (SISO) and Multiple-Input-Multiple-Output (MIMO) cases of consensus dynamics with leader nodes. In particular, we identify the property of structural balance that when combined with symmetry causes uncontrollability of signed consensus dynamics. The key feature of structurally balanced graphs that allows this analysis is that they admit a *gauge transformation* that allows the permutation matrix corresponding to the graph symmetry to be extended to a signed permutation matrix, which we show leads to uncontrollability when corresponding to a symmetry about input nodes. When then use tools developed by Chapman and Mesbahi in [38] to derive controllability and stablilizability conditions for influenced consensus dynamics. In the second half of this section, we show that the property of structural balance, when combined with symmetries in the underlying graph, as well as certain symmetries of the nonlinear dynamics, causes uncontrollability in the context of the accessibility problem. In particular, we consider the same network flows studied in [3, 14, 149], however we extend the controllability analysis to unsigned dynamics. We then extend the bipartite identification problem considered by Pan *et al.* in [130] to the case of nonlinear consensus signed networks. In particular, we use a Koopman operator-theoretic approach alongside Extended Dynamic Mode Decomposition (EDMD) to extract a "Koopman mode" whose sign structure will be shown to contain the bipartite structure.

### *2.2.1 Linear Antagonistic Consensus Networks*

The analysis of the linear signed consensus consists of two main parts, which examine two different notions of influencing control on networks. In the first part, we examine the notion of uncontrollability due to symmetry and interlacing which was initially derived in [135] for signed consensus networks. The notion of control in this case is taking over the state of one or several nodes in the network, and using their edges to inject signals into the system. In the second part, we consider the case where the nodes are controlled by injecting a single-integrator signal to some nodes of the graph. Signed consensus networks are of interest because the negative weights induce a phenomenon known as *clustering*, where agents will not converge to an agreement subspace, but rather converge to opposite equilibria. The condition that causes clustering was identified by [14] as *structural balance*. Surprisingly, we show that this topological feature is exactly the condition that produces uncontrollability of signed networks.

**Lemma 1.** *([14]) The following statements are equivalent:*

1. *The signed graph $\mathcal{G}$ is structurally balanced;*
2. *There exists a gauge $G_t$ such that $G_t\mathcal{A}_sG_t$ has only positive entries (i.e. the graph becomes* unsigned*);*
3. *All cycles in $\mathcal{G}$ are positive;*
4. *The signed Laplacian $\mathcal{L}_s$ has a zero eigenvalue;*
5. *There exists a bipartition of $\mathcal{V}$ such that the edge weights on the edges within the same set are positive, and the edges connecting the two sets are negative.*

It is known that the unsigned consensus dynamics have an agreement subspace spanned by the eigenvector $\mathbb{1}$ corresponding to the zero eigenvalue [116]. For signed consensus, the zero eigenvalue explicitly corresponds to disagreement. Then one may think that structural balance is therefore not a desirable quality for signed consensus, but it turns out that if the

graph is not structurally balanced, the consensus dynamics become trivial in some sense, and converge to 0.

**Theorem 1.** *([14]) If $\mathcal{L}_s$ is structurally unbalanced in a signed graph $\mathcal{G}_s$, then $\lim_{t\to\infty} x(t) = 0$ where $x \in \mathbb{R}^n$ is the state vector with the corresponding dynamics $\dot{x} = -\mathcal{L}_s x$. Otherwise, if $\mathcal{L}_s$ is structurally balanced, then $\lim_{t\to\infty} x(t) = (1/n)\left(\mathbb{1}^\top G_t x(0)\right) G_t \mathbb{1}$.*

Previous work by Rahmani *et al.* in [135] showed that symmetry with respect to a single input and interlacing for multiple input are sufficient for uncontrollability, and this was generalized by Chapman and Mesbahi in [38] to show that fractional symmetry with respect to the inputs is sufficient and necessary for uncontrollability. In this part of the thesis, we aim to show that structural balance is the property that combined with symmetry and interlacing leads to uncontrollability. There are examples of signed networks that are input symmetric but controllable, and we will discuss these in upcoming sections. Before proceeding to our main results, we summarize the various dynamics considered in this part.

There are several variants of consensus dynamics with respect to how inputs are injected into nodes. Given a connected signed graph $\mathcal{G}_s$, we can select one node and use that specific node to inject our input signal $u$. This corresponds to partitioning the Laplacian as follows,

$$\mathcal{L}_s = \left[\begin{array}{c|c} \mathcal{A}_s^f & B_s^f \\ \hline {B_s^f}^\top & \mathcal{A}_s^i \end{array}\right], \tag{2.2}$$

where $f$ and $i$ denote the *floating* and *input* parts of the network respectively. Then, the dynamical system for signed consensus networks is,

$$\dot{x} = -\mathcal{A}_s^f x - B_s^f u \ , \tag{2.3}$$

where for a SISO system we have $\mathcal{A}_s^f \in \mathbb{R}^{(n-1)\times(n-1)}$, $B_s^f \in \mathbb{R}^{(n-1)\times 1}$, and $\mathcal{A}_s^i$ is a scalar. Similarly, one can define the same dynamics as (2.3) for the MIMO system. In that case, $\mathcal{A}_s^f \in \mathbb{R}^{n_f \times n_f}$ and $B_s^f \in \mathbb{R}^{n_f \times n_i}$ where $n_f$ and $n_i$ are the numbers of the nodes in floating and input subgraphs respectively. The *floating signed graph* is denoted by $\mathcal{G}_s^f$.

**Lemma 2.** *[**Popov-Belevitch-Hautus (PBH) Test**] The system described in (2.3) is controllable if and only if none of the eigenvectors of $\mathcal{A}_s^f$ are simultaneously orthogonal to all columns of $B_s^f$.*

The second variant of controlling consensus networks is to simply inject signals into nodes, without taking over the state of the node. We can therefore define the *influenced signed consensus dynamics* with $q$ inputs and $p$ outputs as

$$\dot{x} = -\mathcal{L}_s x + B(I)u, \qquad y = C(O)x, \tag{2.4}$$

where $I \subseteq N$ is the set of input nodes, $O \subseteq N$ the set of output nodes, and $B(I)$ and $C(I)$ are defined as,

$$B(I) = \begin{bmatrix} | & & | \\ e_{i_1} & \cdots & e_{i_q} \\ | & & | \end{bmatrix}, \qquad C(O) = \begin{bmatrix} | & & | \\ f_{j_1} & \cdots & f_{j_p} \\ | & & | \end{bmatrix}^\top, \tag{2.5}$$

in which $e_i$ is the indicator vector for nodes $i \in I$, and $f_j$ for nodes $j \in O$. The control signal vector is $u \in \mathbb{R}^q$.

*The SISO Case*

A standard result by [135] shows that for a SISO consensus network, a symmetry about the input node is sufficient for uncontrollability. We extend this result for the signed consensus networks considered by [14] and [130]. In particular, we show that structural balance and input symmetry for the unsigned graph is sufficient for uncontrollability.

**Remark 1.** *This result shows signed Laplacian is in some sense more robust to symmetries about the input nodes. In particular, there are examples of unsigned graphs that are symmetric about an input and therefore uncontrollable, but whose signed counterparts exhibit controllability despite the symmetry. Therefore, there is not enough conditions to claim the uncontrollability of the system.*

Now, we state and prove a lemma which helps us provide the main result of this section.

**Lemma 3.** *Assume the unsigned graph $\mathcal{G}$ enjoys input symmetry and the signed network $\mathcal{G}_s$ is structurally balanced. Then, the following statements hold*

1. *There exists $J'$ such that $J'\mathcal{A}_s^f = \mathcal{A}_s^f J'$*

2. *For the same $J'$ as part 1, $J'^{\top} B_s^f = B_s^f$*

3. *If $v$ is the eigenvector corresponding to the eigenvalue $\lambda$ of $\mathcal{A}_s^f$. Then, $J'v$ (and hence $v - J'v$) would also be the eigenvectors corresponding to $\lambda$*

*Proof.* From the second statement of lemma 1 there exists $G_t$ such that

$$G_t \mathcal{L}_s G_t = \mathcal{L}, \tag{2.6}$$

and hence,

$$\mathcal{A}_s^f = G' A^f G', \qquad B_s^f = \sigma_n G' B^f.$$

Therefore

$$G_t \left[\begin{array}{c|c} \mathcal{A}_s^f & B_s^f \\ \hline {B_s^f}^{\top} & \mathcal{A}_s^i \end{array}\right] G_t = \left[\begin{array}{c|c} G'\mathcal{A}_s^f G' & G' B_s^f \sigma_n \\ \hline \sigma_n {B_s^f}^{\top} G' & \sigma_n \mathcal{A}_s^i \sigma_n \end{array}\right] = \left[\begin{array}{c|c} A^f & B^f \\ \hline {B^f}^{\top} & A^i \end{array}\right],$$

where $G' = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{n-1})$. W.l.o.g., we can assume $\sigma_n = 1$ since (2.6) also holds for $-G$. This gives,

$$B^f = G' B_s^f.$$

From [135] we know that if $\mathcal{G}$ has the input symmetry structure, then there exists the permutation matrix $J$ such that,

$$J A^f = A^f J.$$

Let $J' = G' J G'$ and,

$$J' \mathcal{A}_s^f = G' J G' G' A^f G' = G' J A^f G' = G' A^f J G' = G' A^f G' G' J G' = \mathcal{A}_s^f J',$$

which completes the proof of part (1). Note,

$$J'^{\top} B_s^f = J'^{\top} G' B^f = G' J^{\top} G' G' B^f = G' J^{\top} B^f = -G' J^{\top} A^f \mathbb{1} = -G' A^f J^{\top} \mathbb{1} = B_s^f,$$

which, in turn, gives the result of part (2). To show that last part, from definition $\mathcal{A}_s^f v = \lambda v$. Then,

$$\mathcal{A}_s^f J' v = J' \mathcal{A}_s^f v = \lambda J' v,$$

implying that $J'v$ is also an eigenvector for the same eigenvalue. Hence, assuming orthonormal eigenvectors for $\mathcal{A}_s^f$, then $v - J'v$ would also be an eigenvector corresponding to $\lambda$. □

The matrix $J'$ introduced for a signed floating graph in lemma Lemma 3 is in some sense correspondent to the permutation matrix $J$ in the unsigned case. In fact, the only difference between $J$ and $J'$ are the elements with the same rows or columns as the negative elements in $G'$. For example, for the graph of Figure 2.1, and considering node 5 to be the input node, we get the following matrices,

$$J = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad G' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \quad J' = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}.$$

Therefore, the first two parts of lemma 3 demonstrate the correlation between the signed and unsigned consensus networks using the gauge transformation $G_t$. Now we are well-equipped to provide the main theorem of this part.

**Theorem 2.** *$\mathcal{G}_s$ is uncontrollable if it is input symmetric and structurally balanced.*

*Proof.* We use the results of lemma 3 and the PBH test to show the uncontrollability of $\mathcal{G}_s$. Let $(\lambda, v)$ be a pair of eigenvalue and eigenvector for $\mathcal{A}_s^f$ so that $\mathcal{A}_s^f v = \lambda v$. Then, from part 3 of lemma 3 we know that $v - J'v$ is also and eigenvector for $\mathcal{A}_s^f$. Then, from part 2 of lemma 3,

$$(v - J'v)^\top B_s^f = v^\top B_s^f - v^\top J'^\top B_s^f = v^\top B_s^f - v^\top B_s^f = 0,$$

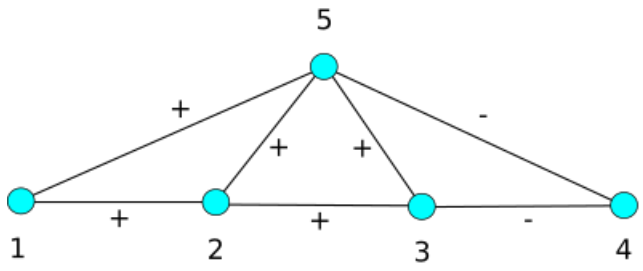



Figure 2.1: A network topology with input symmetry

which implies that the system is not controllable according to the PBH test of controllability.

□

**Remark 2.** *A signed symmetry implies the existence an unsigned symmetry of $\mathcal{G}$. The converse is true when $\mathcal{G}$ is structurally balanced.*

*The MIMO Case*

In this section, we generalize the uncontrollability result for multiple input systems from [135] to signed networks. To this end, we leverage the machinery of interlacing and equitable partitions on graphs. First, we extend the following two lemmas from [73] to the signed case.

**Lemma 4.** *Let $\pi$ be a partition of the structurally balanced signed graph $\mathcal{G}_s$, with adjacency matrix $\mathcal{A}_s$ and characteristic matrix $P$, and let $P' = G_t P$ with $G_t$ defined as in (2.6). Then $\pi$ is equitable if and only if the column space of $P'$ is $\mathcal{A}_s$-*invariant*.*

*Proof. Necessity:* assume $\pi$ is equitable. From lemma 9.3.1 in [73] if $\pi$ is equitable then $AP = P\hat{A}$ with $\hat{A} = A(\mathcal{G}_s/\pi)$. Then, it follows from the second statement in lemma 1,

$$P\hat{A} = AP = G_t\mathcal{A}_s G_t P \quad \Rightarrow \quad \mathcal{A}_s P' = P'\hat{A}.$$

*Sufficiency:* From lemma 9.3.2 in [73], $\pi$ is equitable if there exists $B$ such that $AP = PB$. Then, if such $B$ exists,

$$PB = AP = G_t\mathcal{A}_s G_t P \quad \Rightarrow \quad \mathcal{A}_s P' = P'B.$$

□

**Lemma 5.** *([108]) Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$, let $S$ be a subspace of $\mathbb{R}^n$. Then, $S^\perp$ is $A$-*invariant *if and only if $S$ is $A$-*invariant*.*

**Remark 3.** *([135]) Let $|\mathcal{V}| = n$, $|C_i| = n_i$, and $|\pi| = r$. Then we can find an orthogonal decomposition of $\mathbb{R}^n$ as,*

$$\mathbb{R}^n = \mathcal{R}(P') \oplus \mathcal{R}(Q'),$$

*where* $\mathcal{R}(Q') = \mathcal{R}(P')^{\perp}$. *Then, we can form an orthonormal basis for* $\mathbb{R}^n$ *as,*

$$T = [\ \bar{P}' \mid \bar{Q}'\ ], \tag{2.7}$$

*where* $\bar{P}'$ *and* $\bar{Q}'$ *are normalized* $P$ *and* $Q$ *respectively and satisfy* $\bar{P}'^{\top}\bar{Q}' = 0$ *and* $\bar{Q}'^{\top}\bar{Q}' = I_{n-r}$.

**Lemma 6.** *([73] Theorem 9.5.1) Let* $\Phi \in \mathbb{R}^{n\times n}$ *be a real symmetric matrix, and let* $R \in \mathbb{R}^{n\times m}$ *be such that* $R^{\top}R = I_m$. *Set* $\Theta = R^{\top}\Phi R$ *and let* $\nu_1, \nu_2, \ldots, \nu_m$ *be an orthogonal set of eigenvectors for* $\Theta$ *such that* $\Theta\nu_i = \lambda_i(\Theta)\nu_i$, *where* $\lambda_i(\Theta) \in \mathbb{R}$ *is an eigenvalue of* $\Theta$. *Then,* $\Phi R = R\Theta$ *if the interlacing between* $\Phi$ *and* $\Theta$ *is tight.*

**Lemma 7.** *Given a connected signed graph* $\mathcal{G}_s$, *the system* (2.3) *is uncontrollable if and only if* $\mathcal{L}_s$ *and* $\mathcal{A}_s^f$ *share at least one common eigenvalue.*

*Proof. Necessity:* Assume the system is uncontrollable. Then, by the PBH test there is an eigenvector of $\mathcal{A}_s^f$, say $\nu$, with $B_s^{f^{\top}}\nu = 0$. Let $\mathcal{A}_s^f\nu = \lambda\nu$ for some $\lambda$. Then,

$$\mathcal{L}_s \begin{bmatrix} \nu \\ \mathbf{0} \end{bmatrix} = \left[\begin{array}{c|c} \mathcal{A}_s^f & B_s^f \\ \hline B_s^{f^{\top}} & \mathcal{A}_s^i \end{array}\right] \begin{bmatrix} \nu \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathcal{A}_s^f\nu \\ B_s^{f^{\top}}\nu \end{bmatrix} = \lambda \begin{bmatrix} \nu \\ \mathbf{0} \end{bmatrix}.$$

*Sufficiency:* Suppose $\mathcal{L}_s$ and $\mathcal{A}_s^f$ share an eigenvalue, say $\lambda$. Define $P_f = [I_{n_f}, \mathbf{0}]^{\top}$ be the $n \times n_f$ matrix such that $\mathcal{A}_s^f = P_f^{\top}\mathcal{L}_s P_f$. From Lemma 6 we get $\mathcal{L}_s P_f = P_f\mathcal{A}_s^f$. Let $\mathcal{A}_s^f\nu_f = \lambda\nu_f$ then $\mathcal{L}_s P_f\nu_f = P_f\mathcal{A}_s^f\nu_f$, which results in,

$$\mathcal{L}_s \begin{bmatrix} \nu_f \\ \mathbf{0} \end{bmatrix} = \lambda \begin{bmatrix} \nu_f \\ \mathbf{0} \end{bmatrix}.$$

This implies,

$$\left[\begin{array}{c|c} \mathcal{A}_s^f & B_s^f \\ \hline B_s^{f^{\top}} & \mathcal{A}_s^i \end{array}\right] \begin{bmatrix} \nu_f \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathcal{A}_s^f\nu_f \\ B_s^{f^{\top}}\nu_f \end{bmatrix} = \lambda \begin{bmatrix} \nu_f \\ \mathbf{0} \end{bmatrix},$$

which gives $B_s^{f^{\top}}\nu_f = 0$. □

Lemma 7 is the main tool we use to show the final MIMO uncontrollability result of this section. In the following lemmas, we use linear algebra techniques to provide a criteria in which $\mathcal{L}_s$ and $\mathcal{L}_s^f$ share a common eigenvalue and discuss on the system uncontrollability.

**Lemma 8.** *Suppose a structurally balanced signed graph $\mathcal{G}_s$ has an NEP, $\pi$ with characteristic matrix $P$, then the adjacency matrix $\mathcal{A}_s$ is similar to a block diagonal matrix.*

*Proof.* From (2.24) Let $T = [\bar{P}'|\bar{Q}']$. Define,

$$\bar{A}_s = T^\top \mathcal{A}_s T = \begin{bmatrix} \bar{P}'^\top \mathcal{A}_s \bar{P}' & \bar{P}'^\top \mathcal{A}_s \bar{Q}' \\ \bar{Q}'^\top \mathcal{A}_s \bar{P}' & \bar{Q}'^\top \mathcal{A}_s \bar{Q}' \end{bmatrix}$$

Since $\bar{P}'$ and $\bar{Q}'$ are $\mathcal{A}_s$-*invariant*, there exist $B$ and $C$ such that,

$$\mathcal{A}_s \bar{P}' = \bar{P}' B, \qquad \mathcal{A}_s \bar{Q}' = \bar{Q}' C,$$

from which,

$$\bar{A}_s = \begin{bmatrix} \bar{P}'^\top \mathcal{A}_s \bar{P}' & \bar{P}'^\top \bar{Q}' C \\ \bar{Q}'^\top \bar{P}' B & \bar{Q}'^\top \mathcal{A}_s \bar{Q}' \end{bmatrix} = \begin{bmatrix} A_{P'} & \mathbf{0} \\ \mathbf{0} & A_{Q'} \end{bmatrix},$$

where $A_{P'} = \bar{P}'^\top \mathcal{A}_s \bar{P}'$ and $A_{Q'} = \bar{Q}'^\top \mathcal{A}_s \bar{Q}'$. □

**Lemma 9.** *With the same condition as lemma 8, $\mathcal{L}_s$ is similar to a block diagonal matrix.*

*Proof.* Let $\mathcal{D} = \mathbf{Diag}([d_i \mathbb{1}_{n_i}]_{i=1}^r)$ be the degree matrix with $d_i$ denoting the degree of each node in $C_i$. Then $\mathcal{D}P' = P'\hat{\mathcal{D}}$ where $\hat{\mathcal{D}} = \mathbf{Diag}([d_i]_{i=1}^r)$. Hence, $\mathcal{R}(P')$ is $\mathcal{D}$-*invariant*. Also, from Lemma 5, $\mathcal{R}(Q') = \mathcal{R}(P')^\perp$ is also $\mathcal{D}$-*invariant*. Since from Lemma 4, $\mathcal{R}(P')$ and $\mathcal{R}(Q')$ are also $\mathcal{A}_s$-*invariant*, from $\mathcal{L}_s = \mathcal{D}_s - \mathcal{A}_s$, $\mathcal{R}(P')$ and $\mathcal{R}(Q')$ are $\mathcal{L}_s$-*invariant*. Then following Lemma 8,

$$\bar{L}_s = T^\top \mathcal{L}_s T = \begin{bmatrix} \mathcal{L}_{P'} & \mathbf{0} \\ \mathbf{0} & \mathcal{L}_{Q'} \end{bmatrix},$$

where $\mathcal{L}_{P'} = \bar{P}'^\top \mathcal{L}_s \bar{P}'$ and $\mathcal{L}_{Q'} = \bar{Q}'^\top \mathcal{L}_s \bar{Q}'$. □

**Lemma 10.** *Let $\mathcal{G}_s^f$ be a signed floating graph, and let $\mathcal{A}_s^f$ be defined as in (2.3). If there exists an NEP, $\pi_f$ in $\mathcal{G}_s^f$ and a $\pi$ in the original structurally balanced signed graph $\mathcal{G}_s$ such that all the nontrivial cells in $\pi_f$ are also cells in $\pi$, then $\mathcal{L}_s^f$ is similar to a block diagonal matrix.*

*Proof.* Define $\bar{P}'_f$ and $\bar{Q}'_f$ as in (2.24) for the floating graph so that $\mathcal{R}(\bar{P}'_s) \oplus \mathcal{R}(\bar{Q}'_s) = \mathbb{R}^{n_f}$. Let $\mathcal{D}^f_s$ be the diagonal degree matrix for the floating section of the signed graph $\mathcal{G}_s$. Since $\mathcal{L}^f_s = \mathcal{D}^f_s - \mathcal{A}^f_s$ and all the nontrivial cells in $\pi_f$ are also cells in $\pi$, pursuing lemmas 8 and 9 it follows,

$$\bar{L}^f_s = T_f^\top \mathcal{L}^f_s T_f = \begin{bmatrix} \mathcal{L}^f_{P'} & \mathbf{0} \\ \mathbf{0} & \mathcal{L}^f_{Q'} \end{bmatrix},$$

with $\mathcal{L}^f_{P'} = \bar{P}'^\top_f \mathcal{L}^f_s \bar{P}'_f$ and $\mathcal{L}^f_{Q'} = \bar{Q}'^\top_f \mathcal{L}^f_s \bar{Q}'_f$. □

We are now well-equipped to address the main result of this section. From Lemmas 9 and 10 we know that $\mathcal{L}_s$ and $\mathcal{L}^f_s$ are similar to block diagonal matrices. In the following theorem, we show that depending on the graph partition $\pi$ and structural balance, the two block diagonal matrices have identical blocks and therefore common eigenvalues. This leads to the uncontrollability of the system by Lemma 7.

**Theorem 3.** *Given a connected structurally balanced signed graph $\mathcal{G}_s$ with the floating graph $\mathcal{G}^f_s$, the system (2.3) is uncontrollable if there exist NEP on $\mathcal{G}_s$ and $\mathcal{G}^f_s$, say $\pi$ and $\pi_f$, such that $\pi_f$ contains all nontrivial cells of $\pi$.*

*Proof.* Let $\pi \cap \pi_f = \{C_1, C_2, \ldots, C_{r_1}\}$ with $|C_i| \geq 2,\ i = 1, 2, \ldots, r_1$. Let the nontrivial cells contain the first $n_1$ nodes. Hence $n_1 = \sum_{i=1}^{r_1} |C_i| \leq n_f < n$. Given that $\pi_f$ contains all nontrivial cells of $\pi$, it follows

$$P' = \begin{bmatrix} P'_1 & \mathbf{0} \\ \mathbf{0} & I_{n-n_1} \end{bmatrix}_{n \times r} \quad \text{and} \quad P'_f = \begin{bmatrix} P'_1 & \mathbf{0} \\ \mathbf{0} & I_{n_f - n_1} \end{bmatrix}_{n_f \times r_f},$$

where $P'_1 \in \mathbb{R}^{n_1 \times r_1}$ contains the nontrivial part of the characteristic matrices. Let $\bar{P}'$ and $\bar{P}'_f$ be the normalization of $P'$ and $P'_f$. Define $\bar{Q}'$ and $\bar{Q}'_f$ as in (2.24). Then

$$\bar{Q}' = \begin{bmatrix} Q'_1 \\ \mathbf{0} \end{bmatrix}_{n \times (n_1 - r_1)}, \qquad \bar{Q}'_f = \begin{bmatrix} Q'_1 \\ \mathbf{0} \end{bmatrix}_{n_f \times (n_1 - r_1)},$$

where $Q'_1 \in \mathbb{R}^{n_1 \times (n_1 - r_1)}$ matrix that satisfies $Q'_1 P_1 = 0$. It follows that $\bar{Q}'_f = R^\top \bar{Q}'$ with $R = [I_{n_f}, \mathbf{0}]^\top$. From the definitions of $\mathcal{L}_{Q'}$ and $\mathcal{L}^f_{Q'}$ in Lemmas 9 and 10,

$$L_{Q'} = \bar{Q}'^\top \mathcal{L}_s \bar{Q}' = \bar{Q}'^\top_f R^\top \mathcal{L}_s R \bar{Q}'_f = \bar{Q}'^\top_f \mathcal{L}^f_s \bar{Q}'_f = L^f_{Q'}$$

Thus one block is common between $\mathcal{L}_s$ and $\mathcal{L}_s^f$ leading them to have at least one equal eigenvalue. Therefore, by lemma 7 the system is uncontrollable. □

*Stabilizability and Output Controllability*

Recent developments in controllability have extended the idea of using symmetry to characterize controllability of linear systems [38]. In particular, a relationship between the existence of a solution to a particular convex optimization problem was used to construct the following controllability test.

**Theorem 4** (Symmetry Controllability Test [38])**.** *Consider the general linear dynamics ,*

$$\dot{x} = Ax + Bu, \qquad y = Cx,$$

*with $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{p\times n}$. For $A$ diagonalizable and $C$ full row rank, consider the conditions on a square matrix $P$,*

*a. $P \neq I, AP = PA$ and $PB = B$*

*b. $CP = ZC$ for some $Z \neq I$*

*c. $\frac{1}{2}(P + P^\top) \preceq I$ and $PA + (PA)^\top \preceq A + A^\top$*

*Then, there exists $P \in \mathbb{C}^{n\times n}$ such that*

1. *(a) $\iff$ $(A, B)$ uncontrollable*
2. *(a) & (b) $\iff$ $(A, B, C)$ is output uncontrollable*
3. *(a) & (c) $\iff$ $(A, B)$ is unstabilizable*
4. *(a) & (b) & (c) $\iff$ $(A, B, C)$ is output unstabilizable*

*Note that conditions (a)-(c) are convex, and so convex optimization may be performed to find the existence of such a $P$.*

Using Theorem 4, we can now characterize the controllability, output controllability, stabilizability and output stabilizability of the *influenced signed consensus dynamics*,

$$\dot{x} = -\mathcal{L}_s x + B(I)u, \qquad y = C(O)x,$$

where $I \subseteq N$ is the set of input nodes, $O \subseteq N$ the set of output nodes, and $B(I), C(I)$ are the matrices,

$$B(I) = \begin{bmatrix} | & & | \\ e_{i_1} & \cdots & e_{i_q} \\ | & & | \end{bmatrix}, \qquad C(O) = \begin{bmatrix} | & & | \\ f_{j_1} & \cdots & f_{j_p} \\ | & & | \end{bmatrix}^\top,$$

in which $e_i$ is the indicator vector for nodes $i \in I$, and $f_j$ for nodes $j \in O$. The following lemmas establish the equivalence of controllability under a gauge transformation of the $B$ and $C$ matrices for structurally balanced $\mathcal{L}_s$, and some useful identities that will elucidate the role of the gauge transformation in the main result. For clarity of exposition, we will prove the lemmas at the end of the section.

**Lemma 11.** *Let $(Ls, B(I))$ be the pair in the dynamics (2.4), and let $B_s(I) = G_t B(I)$ for any gauge transformation $B$ (regardless of whether $\mathcal{L}_s$ is structurally balanced). Then, $(-\mathcal{L}_s, B(I))$ is controllable if and only if $(-\mathcal{L}_s, B_s(S))$ is controllable. Furthermore, letting $C_s(O) = C(O)G_t$, we have that $(-\mathcal{L}_s, B(I), C(O))$ is output controllable if and only if $(-L, B_s(I), C_s(O))$ is output controllable.*

**Lemma 12.** *Consider the dynamics in (2.4). Suppose $\mathcal{L}_s$ is structurally balanced with gauge $G_t$.*

1. *Suppose that there is an automorphism $J$ such that $J\mathcal{L}_s = \mathcal{L}_s J$. Then, $J_s\mathcal{L}_s = \mathcal{L}_s J_s$, where $J_s = G_t J G_t$.*

2. *Suppose $J$ is input symmetric. Then, $J_s B_s = B_s$.*

3. *Suppose that there exists $Z \neq I$ such that $ZC(O) = C(O)J$. Then, $ZC_s(O) = C(O)J_s$.*

The following theorem identifies structural balance as the key feature for uncontrollability of the dynamics (2.4), on top of symmetry of the underlying unsigned graph. The crux of the argument is that the gauge transformation allows a signed symmetric automorphism to satisfy the criteria in Theorem 4, allowing a variant of the result in Corollary 5 of [38].

**Theorem 5.** *Let $J$ be a non-trivial fractional automorphism of $L$. Suppose further that $\mathcal{L}_s$ is structurally balanced with gauge $G_t$. Consider dynamics (2.4), and the following conditions on $J$ with $B_s := G_t B(I)$ and $C_s := C(O)G_t$,*

*a.* $J_s B_s = B_s$

*b.* $C_s(R) J_s C_s(N \setminus R)^\top = 0$, $C_s(R) J_s C_s(R)^\top = Z \neq I$.

*c.* $J_s v_i = v_i$ *for all* $v_i \sim \lambda_i(\mathcal{L}_s) > 0$.

*Then,*

1. *(a)* $\iff$ $(-\mathcal{L}_s, B)$ *uncontrollable*
2. *(a) & (b)* $\iff$ $(-\mathcal{L}_s, B, C)$ *is output uncontrollable*
3. *(a) & (c)* $\iff$ $(-\mathcal{L}_s, B, C)$ *is output unstabilizable*
4. *(a) & (b) & (c)* $\iff$ $(-\mathcal{L}_s, B, C)$ *is output unstabilizable*

*The results hold when $B \to B_s$ or $C \to C_s$.*

*Proof.* By Lemma 11, $J_s B_s = B_s$ is equivalent to Theorem 4(a). Suppose that there exists $Z \neq I$ such that $ZC(O) = C(O)J$ to establish condition Theorem 4(b). Note that $C(O)C(O)^\top = I$ and $C(O)C(N \setminus O)^\top = 0$, and $[C(O)^\top,\ C(N \setminus O)^\top]$ is unitary. We can

compute,

$$0 = C(O)J - ZC(N \setminus O) \tag{2.8}$$
$$= (C(O)J - ZC(N \setminus O))[C(O)^\top,\ C(N \setminus O)^\top] \tag{2.9}$$
$$= [C(O)JC(O)^\top - ZC(R)C(R)^\top, \tag{2.10}$$
$$C(O)JC(N \setminus O)^\top - ZC(R)C(N \setminus O)^\top] \tag{2.11}$$
$$= [C(O)JC(O)^\top - Z,\ C(O)JC(N \setminus O)^\top], \tag{2.12}$$

yielding $C(R)JC(N \setminus R)^\top = 0$, $C(R)JC(R)^\top = Z \neq I$. By Lemma 12(3), this is equivalent if we interchange $J \to J_s$ and $C \to C_s$. Now, lets assume 5(a) and that,

$$-J_s\mathcal{L}_s - (\mathcal{L}_s J_s)^\top \succeq -\mathcal{L}_s - \mathcal{L}_s^\top \tag{2.13}$$
$$\implies \frac{1}{2}\left(J_s\mathcal{L}_s + (J_s\mathcal{L}_s)^\top\right) \preceq \mathcal{L}_s, \tag{2.14}$$

in order to establish Theorem 4(c). Since $(J_s + J_s^\top)/2 \preceq I$,

$$\frac{\lambda_i(\mathcal{L}_s)}{2} v_i^\top (J_s + J_s^\top) v_i \geq v_i^\top \mathcal{L}_s v_i = \lambda_i(\mathcal{L}_s), \tag{2.15}$$

with equality if $J_s v_i = v_i$, and hence $\lambda_i(\mathcal{L}_s) > 0$, which holds for the stable modes of $-\mathcal{L}_s$. □

We now prove Lemma 11.

*Proof.* Note that the column space of the controllability matrix of $(-\mathcal{L}_s, B)$

$$\mathcal{C}(B(I)) := \begin{bmatrix} B & -\mathcal{L}_s B & \cdots & (-\mathcal{L}_s)^{n-1} B \end{bmatrix} \tag{2.16}$$

is spanned by columns of the form $(-\mathcal{L}_s)^m e_i$ for $0 \leq m \leq n-1$. The action of a gauge $G_t$ on $B(I)$ is to multiply each column of $B(I)$ by $\pm 1$, and so the column space of the controllability matrix of $(-\mathcal{L}_s, B_s)$,

$$\mathcal{C}(B(I)) := \begin{bmatrix} G_t B & \cdots & (-\mathcal{L}_s)^{n-1} G_t B \end{bmatrix}, \tag{2.17}$$

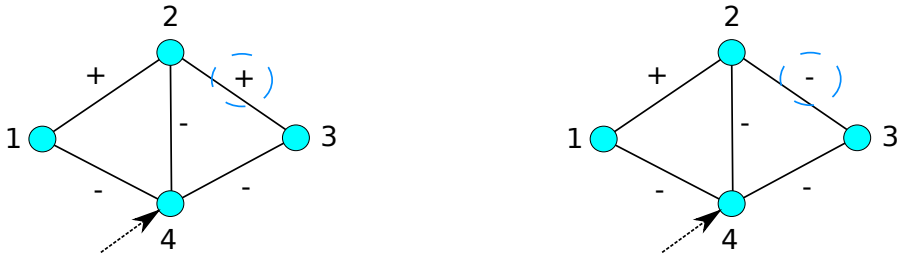
is spanned by columns of the form $\sigma_i(-\mathcal{L}_s)^m e_i$ for $0 \leq m \leq n-1$, and $\sigma_i = \pm 1$. Clearly,

$$\text{span}\{(-\mathcal{L}_s)^m e_i\} = \text{span}\{\sigma_i(-\mathcal{L}_s)^m e_i\} \tag{2.18}$$

and so rank$[\mathcal{C}(B(I))]$ = rank$[\mathcal{C}(B_s(I))]$. The same argument applies for output controllability, considering the output controllability matrix

$$\mathcal{C}(B(I), C(O)) := \begin{bmatrix} CB & \cdots & C(-\mathcal{L}_s)^{n-1}B \end{bmatrix}. \tag{2.19}$$

□




(a) (b)

Figure 2.2: (a) Uncontrollable and structurally balanced graph with an input symmetry about node 4. (b) Controllable and structurally unbalanced graph with an input symmetry about node 4.

We now prove Lemma 12.

*Proof.* 1) $J_s\mathcal{L}_s = G_tJG_tG_t\mathcal{L}G_t = G_tJ\mathcal{L}G_t = G_t\mathcal{L}JG_t = G_t\mathcal{L}G_tG_tJG_t = \mathcal{L}_sJ_s$.

2) We know $JB = B$. Hence, $J_sB_s = G_sJG_sG_sB = G_sJB = G_sB = B_s$.

3) $ZC(O)G_t = C(O)JG_t = C(O)G_tG_tJG_t = C_s(O)J_s$. □

*Examples*

In this section, we provide examples pertaining to the discussions in this part of the thesis. In particular, we show the difference between signed and unsigned consensus.

**Bipartite Consensus.** Signed Laplacian considered in this section exhibit clustering when

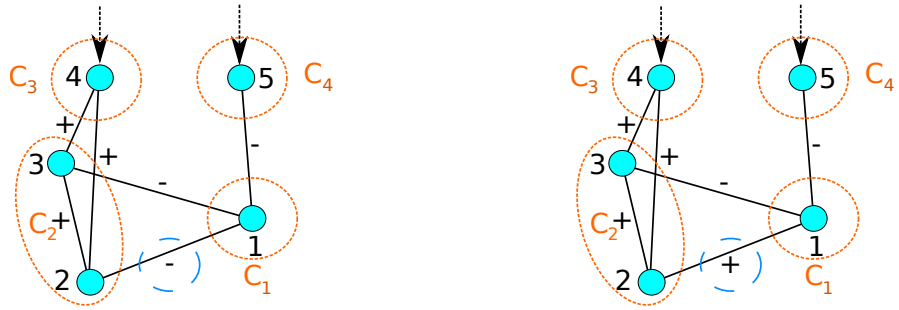



Figure 2.3: (a) Uncontrollable and structurally balanced graph with $\pi_f$ containing nontrivial cells in $\pi$. (b) The same system with one change in the signs. No structural balance and controllable.

structurally balanced. Consider the signed network in Figure 2.5a, and its unsigned counterpart. Note that the signed network is structurally balanced, since the product of the edge weights over all three cycles is positive.

**MIMO Perturbed Laplacian.** In this example, we consider the MIMO case with two input signals injected onto nodes 4 and 5. The partition is equitable and $\pi = \{C_1, C_2, C_3, C_4\}$ and $\pi_f = \{C_1, C_2\}$. Moreover,

$$A = \begin{bmatrix} 0 & -1 & -1 & 0 & -1 \\ -1 & 0 & 1 & 1 & 0 \\ -1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad P' = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A(\mathcal{G}/\pi) = \begin{bmatrix} 0 & 2 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 2 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

The system in figure 2.3a is structurally balanced and $\pi_f$ contains the nontrivial cell $C_2$ in $\pi$ and thus the system is uncontrollable. However, changing one negative sign to positive as depicted in figure 2.3 makes the system structurally unbalanced and the system becomes controllable.

**Influenced Consensus.** Previously, we discussed that symmetry alone is not sufficient for

uncontrollability. In this section we show an example of a symmetric signed network that is controllable. Consider the following influenced consensus networks in Figure 3.2. As one can see, the network in Figure 2.5a is structurally balanced, but the one in Figure 2.5b is not. The Laplacians for these two networks are, respectively,

$$\mathcal{L}_1 = \begin{bmatrix} 2 & -1 & 0 & 1 \\ -1 & 3 & 1 & 1 \\ 0 & 1 & 2 & 1 \\ 1 & 1 & 1 & 3 \end{bmatrix}, \qquad \mathcal{L}_2 = \begin{bmatrix} 2 & -1 & 0 & 1 \\ -1 & 3 & -1 & 1 \\ 0 & -1 & 2 & 1 \\ 1 & 1 & 1 & 3 \end{bmatrix}.$$

The controllability matrices for these two networks are, respectively:

$$\mathcal{C}_1 = \begin{bmatrix} 0 & 1 & 4 & 16 \\ 0 & 1 & 4 & 16 \\ 0 & 1 & 4 & 16 \\ 1 & 3 & 12 & 48 \end{bmatrix}, \qquad \mathcal{C}_2 = \begin{bmatrix} 0 & 1 & 4 & 14 \\ 0 & 1 & 6 & 32 \\ 0 & 1 & 6 & 30 \\ 1 & 3 & 12 & 52 \end{bmatrix}.$$

As one can see, the network in Figure 2.5a is uncontrollable since $\mathcal{C}_1$ is rank-deficient, but the network in Figure 2.5b is controllable, and hence one can conclude that unsigned symmetry is not sufficient for uncontrollability.

### *2.2.2 Nonlinear Antagonistic Bipartite Consensus Networks*

In this part, we tackle two problems regarding nonlinear signed consensus networks. In the first section, we extend the result in [3] to the signed networks. We examine the necessary conditions of uncontrollability in nonlinear signed network systems due to input and dynamics symmetry. In particular, we will show how the additional topological property of structural balance in signed networks plays a key role in driving uncontrollability. In the second section, we show that a particular *Koopman mode* from the EDMD approximation of the Koopman operator contains the sign structure corresponding to the bipartition in Lemma 1(5). This extends work by Pan *et al.* who considered the equivalent problem for linear signed consensus [130]. Before we proceed, in this section we make a slight change of notation and suppose that an automorphism of the graph $\mathcal{G}$ is a permutation $\phi : \mathcal{V} \to \mathcal{V}$ of its nodes such that $\phi(i)\phi(j) \in \mathcal{E}$ if and only if $ij \in \mathcal{E}$. The permutation $\phi$ *induces* a

mapping $\varphi : \mathbb{R}^n \to \mathbb{R}^n$ such that $[\varphi(x)]_i = x_{\phi(i)}$. Let the permutation matrix $J$ be such that $[J]_{ij} = 1$ if $\phi(i) = j$ and zero otherwise. Therefore, the permutation matrix $J$ is simply the Jacobian matrix of $\varphi$, in that $J = D\varphi$. Thus, $\phi$ *represents the automorphism* of $\mathcal{G}$ if and only if $JA(\mathcal{G}) = A(\mathcal{G})J$.

*Nonlinear Controllability of Signed Networks*

In this section, we extend previous work [3] to analyze the controllability of nonlinear consensus protocols to the case where these protocols run on a signed network. We consider three types of nonlinear consensus protocols, following the nomenclature in [3, 14, 149],

- **Absolute Nonlinear Flow**

$$\dot{x}_i = -\sum_{j \in N_i} [f(x_i) - \text{sgn}(a_{ij}) f(x_j)] \tag{2.20}$$

- **Relative Nonlinear Flow**

$$\dot{x}_i = -\sum_{j \in N_i} f\left(x_i - \text{sgn}(a_{ij}) x_j\right) \tag{2.21}$$

- **Disagreement Nonlinear Flow**

$$\dot{x}_i = -f\left(\sum_{j \in N_i} x_i - \text{sgn}(a_{ij}) x_j\right) \tag{2.22}$$

To make the manuscript self-contained, we provide two main theorems from [3] which we use later to demonstrate uncontrollability.

**Theorem 6.** *([3]) Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and $F : \mathbb{R}^n \to \mathbb{R}^n$ be a flow on $\mathcal{G}$. Assume $\varphi$ is a non-identity symmetry on $F$. Then, for any leader $l$, the leader-follower network flow on $\mathcal{G}$ induced by $l$ is not accessible from the origin in $\mathbb{R}^{n-1}$.*

**Theorem 7.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and let $F : \mathbb{R}^n \to \mathbb{R}^n$ be the dynamics in any of (2.20)-(2.22). Also, assume $\varphi$ be an automorphism of $\mathcal{G}$. Then $F$ is $\varphi$-invariant.*

**Remark 4.** *The more general case of* (2.20)-(2.22) *holds when each node has its own smooth nonlinear function* $f_i : \mathbb{R} \to \mathbb{R}$. *However, as shown in [3],* $f_i = f_{\phi(i)}$ *is a necessary and sufficient condition for all the subsequent controllability analysis to work. Since we assume the general function* $f$ *for all nodes, this condition is automatically satisfied.*

The behavior of the dynamics (2.20)-(2.22) clearly depends on the choice of $f : \mathbb{R} \to \mathbb{R}$. In [14], several classes of functions were considered. First, the class of *translated positive, infinite sector nonlinearities* $\mathcal{S}$ is defined as,

$$\mathcal{S} := \Big\{ f : [f(x) - f(x^*)](x - x^*) > 0 \text{ for } x \neq x^*, f(0) = 0, \int_{x^*}^{x} f(t)dt \to \infty \text{ as } |x| \to \infty \Big\}. \tag{2.23}$$

See [93] for properties of this class of functions. A subset $\mathcal{S}_0 \subset \mathcal{S}$ of these functions that will be used later is the *untranslated* ($x^* = 0$) *positive, infinite sector nonlinearities* given by,

$$\mathcal{S}_0 := \Big\{ f : \ f(x^*)x > 0 \text{ for } x \neq 0, f(0) = 0, \int_{0}^{x} f(t)dt \to \infty \text{ as } |x| \to \infty \Big\}. \tag{2.24}$$

The reason these classes of functions are interesting is that when combined with the dynamics introduced in (2.20)-(2.22), *clustering* occurs in a structurally balanced graph. This is summarized in the following theorem.

**Theorem 8.** *(*Theorems 3 & 4 in [14]*) Consider a graph* $\mathcal{G}$. *Assume either the dynamics* (2.20) *with* $f \in \mathcal{S}$ *or the dynamics* (2.21) *with* $f \in \mathcal{S}_0$ *running on* $\mathcal{G}$. *Then,*

$$\lim_{t\to\infty} x(t) = \frac{1}{n} \left( \mathbb{1}^{top} G_t x(0) \right) G_t \mathbb{1}, \tag{2.25}$$

*if and only if* $\mathcal{G}$ *is structurally balanced (with gauge transformation* $G_t$*).*

According to this theorem, for certain classes of functions, the dynamics will converge to two different clusters. These clusters are exactly those corresponding to the bipartite consensus condition in Lemma 1(5).

In the following subsections, we elaborate on the controllability of the dynamics (2.20)-(2.22) and show that a notion of symmetry about the input node, as well as structural balance, lead to uncontrollability. For each case we consider both even (symmetric) and

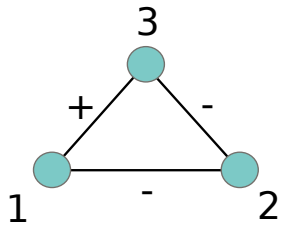



Figure 2.4: Example of a signed automorphism

odd (anti-symmetric) functions $f$. From [14] we know that if the underlying signed graph $\mathcal{G}$ is structurally balanced, then there exists a gauge $G_t$ that acts as a similarity transformation on the adjacency matrix of $\mathcal{G}$ in that $G_t A_s(\mathcal{G}) G_t = A$ where $A$ is the adjacency matrix of unsigned $\mathcal{G}$. We will show that $G_t$ defines a useful coordinate transformation that allows an immediate application of the uncontrollability test derived by Aguilar and Gharesifard [3]. Before that, we need the following definition which extends the notion of a graph symmetry for signed graphs.

**Definition 1.** *Let $\varphi$ be a non-identity automorphism on graph $\mathcal{G}$. Suppose that this graph is structurally balanced, with gauge transformation $G_t$ induced by the function $g : \mathbb{R}^n \to \mathbb{R}^n$ defined by $[g(x)]_i = \sigma_i x_i$. Then, we define* signed automorphism *operator as $\varphi' = g \circ \varphi \circ g$. Moreover, assume that $J$ is the matrix representation of the permutation operator $\varphi$, in that $J = D\varphi$. Then the analogous matrix $J' = G_t J G_t$ is the matrix representation of the signed permutation operator $\varphi'$, in that $J' = D(g \circ \varphi \circ g)$.*

Definition 1 implies that unlike the unsigned graph, the signed automorphism contains sign alterations of edge weights as well as permutations of nodes. Hence, if $\varphi(x_i) = x_r$, then $\varphi'(x_i) = \sigma_i \sigma_r x_r$. For example, consider the graph in Figure 2.4. The corresponding gauge transformation and automorphisms are defined as follows:

$$G_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad J = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad J' = G_t J G_t = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

One can note that while the structures of $J$ and $J'$ are similar, the signs of the elements are

different. This also implies,

$$\varphi'(x_1, x_2, x_3) = (-x_2, -x_1, x_3) \;\Rightarrow\; \begin{cases} \varphi'(x_1) = \sigma_1\sigma_2 x_2 \\ \varphi'(x_2) = \sigma_1\sigma_2 x_1 \\ \varphi'(x_3) = \sigma_3\sigma_3 x_3 \end{cases} \tag{2.26}$$

Recall that we use $\phi(i)$ as the action of the automorphism on the index of a node rather than the more obscure notation $\varphi(v_i)$. For example, in Figure 2.4 we have that $\phi(1) = 2$ and $\phi(2) = 1$.

### *2.2.3 Absolute Nonlinear Flow*

The original definition of absolute nonlinear flow allows the function $f$ to vary across the nodes [149]. In our problem setup, however, all such functions are assumed to be equal, and thus the dynamics resemble linear consensus in that we can write $\dot{x} = -L_s f(x)$, where $f(x)$ is the function $f$ applied entry-wise to the vector $x$. In the following theorem, we will show that for absolute nonlinear flow with odd functions $f$, structural balance directly generalizes the uncontrollability conditions in [3]. For the case of even functions, we need to impose additional topological structure on the edge weights of the underlying graph.

**Theorem 9.** *Consider a structurally balanced graph $\mathcal{G}$ with gauge transformation $G_t$ and* absolute nonlinear flow *dynamics* (2.20)*. Further suppose $\mathcal{G}$ has a non-trivial signed automorphism $\varphi'$. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a smooth odd function (for example, odd $f \in \mathcal{S}_0$ with $f$ smooth). Then, for any vertex $j \in \mathrm{Fix}(\varphi')$ chosen as the leader, the leader-follower network is not accessible from the origin in $\mathbb{R}^{n-1}$. Moreover, the same results holds for smooth even functions $f$ if $\varphi'$ preserves edge signs, in that $\mathrm{sgn}(a_{ij}) = \mathrm{sgn}(a_{\phi(i)\phi(j)})$.*

*Proof.* Following Equation (1.1), let $F$ denote the network flow and assume the dynamics in (2.20). We will first note that for smooth odd functions $f$, the dynamics of the system can become unsigned by a convenient coordinate transformation. Let $z = G_t x$, or $z_i = \sigma_i x_i$.

Then, following [13]

$$\begin{aligned}\dot{z}_i &= -\sigma_i \sum_{j\in\mathcal{N}_i} f(\sigma_i z_i) - \operatorname{sgn}(a_{ij}) f(\sigma_j z_j) \\ &= -\sigma_i \sum_{j\in\mathcal{N}_i} \sigma_i f(z_i) - \operatorname{sgn}(a_{ij})\sigma_j f(z_j) \\ &= -\sum_{j\in\mathcal{N}_i} f(z_i) - f(z_j),\end{aligned}$$

which is defined as the unsigned absolute nonlinear flow. unsigned nonlinear dynamical system. Then, from Theorems 6 and 7 we conclude the system is inaccessible.

Now, suppose $f$ is an even function. On one hand, from (2.20) we have

$$\begin{aligned}F_i(\varphi'(x)) &= -\sum_{l\in N_r} f(\sigma_i\sigma_r x_r) - \operatorname{sgn}(a_{ij}) f(\sigma_j\sigma_l x_l) \\ &= -\sum_{l\in N_r} f(x_r) - \operatorname{sgn}(a_{ij}) f(x_l).\end{aligned}$$

On the other hand we have

$$F_{\phi(i)}(x) = F_r(x) = -\sum_{l\in N_r} f(x_r) - \operatorname{sgn}(a_{rl}) f(x_l).$$

Hence, $F$ is $\varphi'$-invariant if $\operatorname{sgn}(a_{ij}) = \operatorname{sgn}(a_{rl})$. □

The condition assumed in Theorem 9 for the case of even functions can be interpreted as the sign symmetry of the graph. This means that if two nodes are connected antagonistically, their connection will remain negative even after the signed automorphism. This is in addition to the topological property of structural balance.

### *2.2.4 Relative Nonlinear Flow*

The following result shows that unlike the case of absolute nonlinear flow, structural balance on top of the signed variant of the conditions in Theorem 7 imply that the network flow is not accessible from the origin for both even and odd functions, without any additional constraints on edge weight permutations.

**Theorem 10.** *Consider a structurally balanced graph $\mathcal{G}$ with gauge transformation $G_t$ and* relative nonlinear flow *dynamics. Further suppose $\mathcal{G}$ has a non-trivial automorphism $\varphi'$.*

*Let $f : \mathbb{R}^n \to \mathbb{R}$ be a smooth odd or even function (for example, odd $f \in \mathcal{S}_0$ with $f$ smooth). Then, for any vertex $j \in \mathrm{Fix}(\varphi')$ chosen as the leader, the leader-follower network is not accessible from the origin in $\mathbb{R}^{n-1}$.*

*Proof.* Let the same notions as in the proof of Theorem 9. We will show that both cases of odd and even functions $f$ lead to $\varphi'$-invariance of the flow $F$ and therefore the inaccessibility from the origin.

Let $f$ be an odd function. Changing the coordinates by $z = G_t x$ yields,

$$\begin{aligned}
\dot{z}_i &= -\sigma_i \sum_{j \in \mathcal{N}_i} f(\sigma_i z_i - \mathrm{sgn}(a_{ij})\sigma_j z_j) \\
&= -\sigma_i \sum_{j \in \mathcal{N}_i} f(\sigma_i (z_i - \sigma_i \mathrm{sgn}(a_{ij})\sigma_j z_j)) \\
&= -\sum_{j \in \mathcal{N}_i} f(z_i - z_j),
\end{aligned}$$

which is the unsigned relative nonlinear flow. Hence $\varphi'$-invariance follows from Theorems 6 and 7.

For even function $f$, from (2.21)

$$F_i(\varphi'(x)) = -\sum_{l \in \mathcal{N}_r} f(\sigma_i \sigma_r x_r - \sigma_j \sigma_l \mathrm{sgn}(a_{ij}) x_l) \tag{2.27}$$

$$= -\sum_{l \in \mathcal{N}_r} f(\sigma_i \sigma_r (x_r - \sigma_r \sigma_l \sigma_i \sigma_j \mathrm{sgn}(a_{ij}) x_l)) \tag{2.28}$$

$$= -\sum_{l \in \mathcal{N}_r} f(x_r - \sigma_r \sigma_l x_l) \tag{2.29}$$

$$= -\sum_{l \in \mathcal{N}_r} f(x_r - \mathrm{sgn}(a_{rl}) x_l) \tag{2.30}$$

$$= F_r(x), \tag{2.31}$$

where we have used the fact that $\sigma_i \sigma_j \mathrm{sgn}(a_{ij}) > 0$ and hence $\sigma_i \sigma_j = \mathrm{sgn}(a_{ij})$ for all $i$ and $j \in \mathcal{N}_i$. □

*Disagreement Nonlinear Flow*

The next theorem shows that again for both even and odd smooth functions $f$, $\varphi'$-invariance and inaccessibility from the origin hold when considering structurally balanced graphs with

input symmetries.

**Theorem 11.** *Consider a structurally balanced graph $\mathcal{G}$ with gauge transformation $G_t$ and* disagreement nonlinear flow *dynamics. Further suppose $\mathcal{G}$ has a non-trivial automorphism $\varphi'$. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a smooth odd or even function (for example, odd $f \in \mathcal{S}_0$ with $f$ smooth). Then, for any vertex $j \in \mathrm{Fix}(\varphi')$ chosen as the leader, the leader-follower network is not accessible from the origin in $\mathbb{R}^{n-1}$.*

The proof is almost identical to that of Theorem 10 with the same change of coordinates for the odd function case, and the same sign analysis for the even function case.

**Remark 5.** *The analysis of the section demonstrates that for all of the three nonlinear flow structural balance in addition to $\varphi'$-invariance leads to the nonlinear system uncontrollability for some particular function $f$ (even or odd). The only exception was shown to be the case of an even function $f$ for the absolute nonlinear flow where we also need to have a edge-sign symmetry.*

### *2.2.5 Bipartite Identification with the Koopman Operator and EDMD*

In this section, we extend the data-driven approach by Pan *et al.* in using data-driven methods to identify the bipartite consensus for some of the nonlinear network flows considered in the preceding section. Our main result is that the Koopman mode corresponding to the zero eigenvalue of the Koopman operator contains the sign structure indicating the bipartite consensus.

**Theorem 12.** *Consider the dynamics (2.20) with $f \in \mathcal{S}$ and (2.21) with $f \in \mathcal{S}_0$. Recall that the full-state observable can be written in terms of the Koopman triple as,*

$$x(t) = \sum_{j=1}^{\infty} e^{\lambda_j t} \varphi_j(x_0) v_j. \tag{2.32}$$

*If the underlying graph is structurally balanced, then the following hold:*

1. *$\lambda_i \leq 0$ with a unique zero eigenvalue.*

2. *Let $\lambda_1 = 0$. Then, the sign structure of the corresponding Koopman mode $v_1$ corresponds to the bipartition of nodes denoted in Lemma 1(5).*

*Proof.* By Theorem 8, if $\mathcal{G}$ is structurally balanced, then we have that,

$$\lim_{t\to\infty} x(t) = \frac{1}{n}\left(\mathbb{1}^{top} G_t x_0\right) G_t \mathbb{1}. \tag{2.33}$$

Hence, $\lambda_i \leq 0$ since otherwise the RHS of Equation (2.4) does not converge. The sign structure of the vector $G_t\mathbb{1}$ corresponds to bipartition of nodes denoted in Lemma 1(5). Since $\lambda_1 = 0$ is unique, by setting $\alpha = (1/n)\mathbb{1}^{top} G_t x(0)$ we have that,

$$\lim_{t\to\infty} x(t) = \lim_{t\to\infty} \sum_{j=1}^{\infty} e^{\lambda_j t}\varphi_j(x_0)v_j = \varphi_1(x_0)v_1 = \alpha G_t \mathbb{1}. \tag{2.34}$$

Since both $\alpha$ and $\varphi_1(x_0)$ are constants, we conclude that $v_1 \propto G_t\mathbb{1}$ and the claim follows. □

Next, we provide an example where we use EDMD to approximate the first mode of the Koopman operator to obtain the sign structure corresponding to the bipartite consensus.

### *2.2.6 Examples*

In this section, we show some examples that highlight the necessity of combining $\varphi$-invariance, structural balance and leader-node symmetry for uncontrollability of signed networks. For examples that show the necessity of combining $\varphi$-invariance and leader-node symmetry for unsigned graphs, we refer the reader to [3]. We then show an example of using EDMD to obtain the bipartite consensus structure of a nonlinear flow on a structurally balanced graph.

**Unsigned Symmetry is Not Sufficient for Uncontrollability.** Here we show an example of a network flow on a signed graph which has a leader-node symmetry. We show that in one case, the graph is structurally balanced, and the induced flow is hence uncontrollable. By altering the sign on a single edge, structural balance is lost and the resulting network flow is controllable.

Consider the structurally balanced graph in Figure 2.5a, and the structurally unbalanced graph in Figure 2.5b. These have respective dynamics

$$\dot{x} = -L_1 x - \mathbf{1}u, \quad \dot{x} = -L_2 x - \mathbf{1}u,$$


(a) (b)

Figure 2.5: (a) Structurally balanced graph with a leader symmetry about node 4. (b) Structurally unbalanced graph with a leader symmetry about node 4.

with Laplacians,

$$\mathcal{L}_1 = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & 1 \\ 0 & 1 & 2 \end{bmatrix}, \qquad \mathcal{L}_2 = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix}.$$

The controllability matrices of these dynamics are respectively,

$$\mathcal{C}_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \qquad \mathcal{C}_2 = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 3 & 11 \\ 1 & 3 & 9 \end{bmatrix}.$$

**Identification of Bipartite Structure.** In this example, we consider a numerical method to identify the bipartite consensus structure of nonlinear dynamics on a structurally balanced graph, *i.e.*, obtaining the bipartition of nodes in Lemma 1(5). We do this by exploiting Theorem 12, and numerically approximating the Koopman mode corresponding to the zero eigenvalue. The goal here is to be able to, with no knowledge of the underlying connection structure between agents in a natural system, collect data from a natural system and numerically identify the two groups of agents between which only exist antagonistic interactions, in other words identify the bipartition of nodes in Lemma 1(5).

Consider the dynamics (2.21) with function $f(\cdot) = \sin(\cdot)$ with the underlying structurally balanced graph pictured in Figure 2.6.

**Algorithm 1** Extended Dynamic Mode Decomposition

*Initialize*

**Data:** $\{[X_i, Y_i]\}_{i=1}^{n-1} = \{[X_i, X_{i+1}]\}_{i=1}^{n-1}$

**Dictionary:** $\Psi_i = \psi_i(x_1, \ldots, x_6)$, $1 \leq i \leq N_K$

**Matrix:** $B \in \{0,1\}^{n \times N_k}$ st $\Psi_j = x_i \Leftrightarrow B_{ij} = 1$

*Compute*

$G = (n-1)^{-1} \sum_{m=1}^{n-1} \Psi(X_m)^* \Psi(X_m)$

$A = (n-1)^{-1} \sum_{m=1}^{n-1} \Psi(X_m)^* \Psi(Y_m)$

$K = G^{\dagger} A$

*Decompose $K$ Into*

**Eigenvalues:** $\mu_i$

**Right Eigenvectors:** $\xi_i$

**Left Eigenvectors:** $w_i$

*Compute*

**Koopman Modes:** $v_i = (w_i^* B)^T$, $1 \leq i \leq N_K$

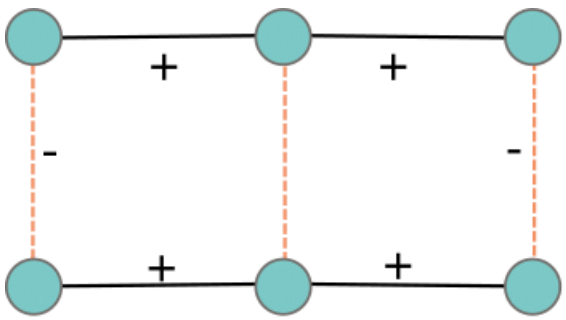



Figure 2.6: Underlying structurally balanced signed graph for EDMD example. Dashed edges indicate negative edges.

We use EDMD to approximate the first Koopman mode corresponding to the zero eigenvalue. Due to space constraints, we refer the reader to [171] for a detailed explanation of the EDMD algorithm; a summary of the relevant computations is shown in Algorithm 1.

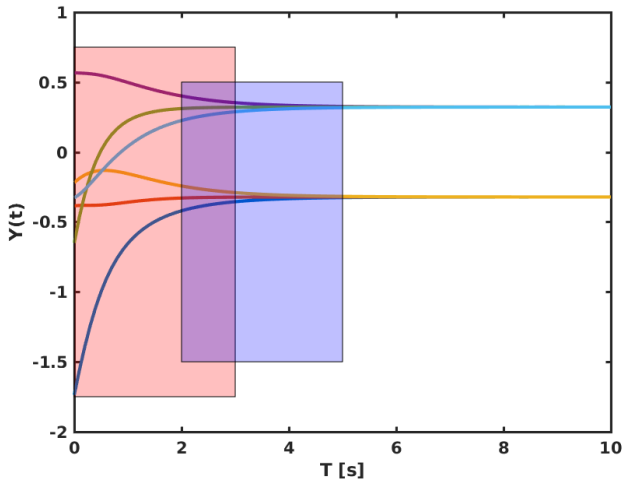



Figure 2.7: Evolution of Dynamics 2.21 with $f(\cdot) = \sin(\cdot)$ on the graph in Figure 2.6. Left (red) shaded region corresponds to $\bar{v}_1^2$ and the right (blue) shaded region corresponds to $\bar{v}_1^3$.

In particular, we use a dictionary of functions of the form $\psi_k = \prod_{i=1}^{6} H(x_i, j_i)$ where $H(x_i, j_i)$ is the $j_i$th order Hermite polynomial of $x_i$. We use all such possible $\psi_k$ for Hermite polynomials up to order 2. Simple combinatorics indicates that there are 729 such functions $\psi_k$ for a 6-node graph; hence the parameter $N_k$ in Algorithm 1 is 729. The dynamics are shown in Figure 2.7 for an initial condition of $x_0 = (-1.73, -0.38, -0.21, 0.56, -0.65, -0.32)$. The EDMD procedure in Algorithm 1 was applied to three sets of data with a time-step of 0.1, as depicted in Figure 2.7: the full data (from $0 \leq t \leq 10$), the data in the red shaded region ($0 \leq t \leq 3$) and the data in the blue shaded region $2 \leq t \leq 5$). The computed Koopman modes corresponding to $\bar{\lambda}_1 \approx 1$ for these regions respectively are ,

$$\bar{v}_1^1 = \begin{bmatrix} 0.018 \\ 0.026 \\ 0.016 \\ -0.020 \\ -0.020 \\ -0.011 \end{bmatrix}, \qquad \bar{v}_1^2 = \begin{bmatrix} 0.019 \\ 0.021 \\ 0.015 \\ -0.018 \\ -0.020 \\ -0.007 \end{bmatrix}, \qquad \bar{v}_1^3 = \begin{bmatrix} 0.016 \\ 0.023 \\ 0.015 \\ -0.018 \\ -0.016 \\ -0.012 \end{bmatrix}, \tag{2.35}$$

which all contain sign structure corresponding to the bipartition depicted in Figure 2.6.

### *2.3 Game Networks*

Game theory has been successfully employed in non-cooperative decision-making [20, 59, 97, 115, 144, 157]. As the key solution concept, Nash Equilibrium (NE) plays a central role in game theoretic analysis defined, in a nutshell, as a strategy from which no player has an incentive to deviate from unilaterally. Upon convexity and continuous differentiability assumptions, the NE can be characterized as the solution of a Variational Inequality (VI) problem [61], where its existence can be guaranteed under assumptions on action sets and the Jacobian of the game [61, 136]. Distributed NE seeking and multiagent network games refers to the class of algorithms that aim to learn a global NE via local information exchange [12, 70, 72, 77, 103, 131, 138, 158], where each node of the network represents a player in the game.

In this section, we introduce a non-cooperative game between two teams, each consisting of players that interact over a network. While the goal of each team is to learn the global NE, players have no information from the other team, neither do they enjoy a global decision-making capability. Instead, only based on local observations, players choose a local strategy at each node and receive a local cost, resulting in learning the global NE in a distributed fashion. This setup resembles a scenario where each node in the network is subjected to actions that contribute to distinct objectives–in this sense, there is duality in the interactions between nodes in the network. For example, each node can represent a socioeconomic entity, with objectives that are not completely aligned with each other (altruistic vs. profit-seeking). Of prime interest in this section is how algebraic and combinatorial properties of the network, such as its connectivity, contribute to the performance of distributed learning in the context of games. The choice of a primal/dual approach that is built around dual averaging [123], and its online implementation [173], is primarily motivated by this overarching objective; this choice is also consistent with how dual averaging has been used in the context of distributed optimization to underscore the "network-effect" on the convergence and optimality properties of distributed optimization, where there is no "duality" in the nodes' operation. However, in order to extend the work of [58] to the game setting, one has to pay a particular attention to the information structure, as for example,

it would be unreasonable to assume information sharing with the opponents in the game setting.

In the sequel, we introduce the main framework of our analysis. Herein, we briefly mention some background material on dual averaging which is the workhorse of our methodology. We then continue by proposing the distributed setup followed by a two-player game-theoretic framework, which can be generalized to multi-player setting with minimum extra effort.

### *2.3.1 Standard Dual Averaging*

The dual averaging algorithm proposed by Nesterov [123] is a subgradient scheme for non-smooth convex problems. The primal-dual nature of this method generates two sequence of iterates $\{x(t), z(t)\}_{t=0}^{\infty}$ contained within $\mathcal{X} \times \mathbb{R}^d$ such that the updates of $z(t)$ is responsible for averaging the support functions in the dual space, while the updates of $x(t)$ establishes a dynamically updated scale between the primal and dual spaces. More precisely, after receiving the subgradient $g(t) \in \partial f(x(t))$ at iteration $t$, the algorithm is updated as follows,

$$\begin{aligned} z(t+1) &= z(t) + \gamma(t) g(t), \\ x(t+1) &= \Pi_{\mathcal{X}}^{\psi}(-z(t+1), \alpha(t)), \end{aligned} \tag{2.36}$$

where $\gamma(t) > 0$, $\{\alpha(t)\}_{t=0}^{\infty}$ is a positive non-increasing sequence, and

$$\Pi_{\mathcal{X}}^{\psi}(z, \alpha) \coloneqq \operatorname{argmin}_{x \in \mathcal{X}} \Big\{ \langle -z, x \rangle + \frac{1}{\alpha} \psi(x) \Big\} \tag{2.37}$$

is a generalized projection according to a strongly convex *prox-function* $\psi(.)$.

### *2.3.2 Our Model*

We consider the distributed learning problem for a game between two teams (players), both playing on a network consisting of $n$ nodes connected via a communication graph $\mathcal{G}$. [1] To this end, each team has a representative on each node, hence $2n$ members in total (Figure 2.8). The teams are grouped within the sets $\mathcal{I}_\ell = \{(\ell, 1), \ldots, (\ell, n)\}$ for $\ell \in \{A, B\}$

[1] It is worth noting that different networks could be associated with each team where the communication graph would be their overlaps, which is not considered here due to brevity.

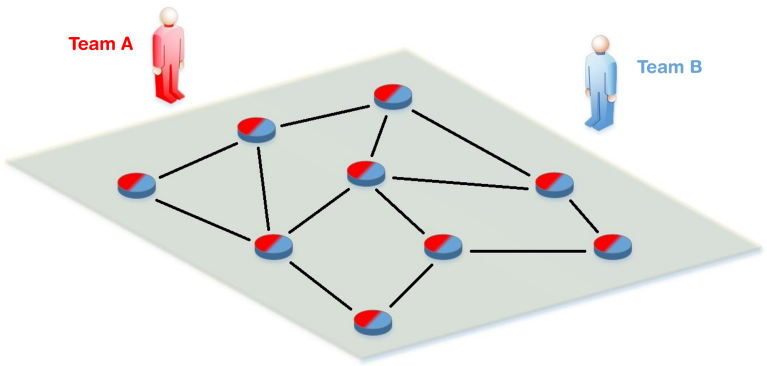



Figure 2.8: Schematic of our game setup including two teams (red A and blue B) with each team having a representative at each node.

and each team has a choice of action $x_\ell \in \mathcal{X}_\ell \subset \mathbb{R}^{d_\ell}$ that minimizes the following global cost,

$$f_\ell(x_A, x_B) = \frac{1}{n}\sum_{i=1}^{n} f_{\ell,i}(x_A, x_B), \tag{2.38}$$

which is the average of its members' costs at each node $i$, denoted by $f_{\ell,i}$. The goal is to learn a global NE while players have no global decision-making capability. Instead, $\mathcal{G}$ is assumed to be connected and players can communicate within their own team according to the network structure.

To learn a global NE, each team updates the state of its nodes using a distributed dual averaging type method. The network-based information flow of our algorithm is related to [58] in the sense of consensus error of the dual variable, however, in a non-cooperative game-theoretic setup, convergence of such iterative methods to NE is non-trivial due to nature of equilibria, limited information exchange, etc. Sufficient conditions of convergence is further discussed in the following sections.

In our proposed algorithm, at each node $i$ and iteration $t$, player $\ell \in \{A, B\}$ maintains an estimate of its team's action as $x_{\ell,i}(t)$. A communication protocol is designed for sharing dual variables among the nodes of each team, where node $i$ updates its dual variable $z_{\ell,i}(t)$ using a convex combination of those of its neighboring teammates. Then it maps $z_{\ell,i}(t)$ back to the set of admissible actions $\mathcal{X}_\ell$ followed by taking its local action $x_{\ell,i}(t)$. Subsequently,

players observe the actions of the opponent at each node and locally obtain an estimate of the subgradient of their distributed cost (or reward). We show that under some regularity assumptions, this process provides players with enough local information to decide about the next step and eventually learn the global NE.

### *2.3.3 Team-based Dual Averaging Algorithm*

We assume that the structure of $\mathcal{G}$ induces a doubly stochastic matrix $P_\ell$ available to each team where $P_{\ell,ij} > 0$ if and only if nodes $i$ and $j$ are connected. Then each player $\ell \in \{A, B\}$ at iteration $t$ and each node $i \in V$ computes updates by the following,

$$\begin{aligned} z_{\ell,i}(t+1) &= \sum_{j \in \mathcal{N}_{\ell,i}} P_{\ell,ij} z_{\ell,j}(t) + \gamma_\ell(t) g_{\ell,i}(t), \\ x_{\ell,i}(t+1) &= \Pi^{\psi}_{\mathcal{X}_\ell}\big( - z_{\ell,i}(t+1), \alpha_\ell(t)\big), \end{aligned} \tag{2.39}$$

where $z_{\ell,i}$ and $x_{\ell,i}$ are the dual variable and the local action of player $\ell$ at node $i$ respectively, $g_{\ell,i}$ is a subgradient of the local cost $f_{\ell,i}$ at the local actions $x_{\ell,i}(t)$ as,

$$g_{\ell,i}(t) \in \partial_\ell f_{\ell,i}(x_{A,i}(t), x_{B,i}(t)), \tag{2.40}$$

where $\partial_\ell$ is the differential w.r.t. the action of player $\ell$. Finally, $\alpha_\ell(t)$ and $\gamma_\ell(t)$ are sequences of positive step-size with $\alpha_\ell(t)$ being non-increasing. Note that $x_{\ell,i}$ can be viewed as the local copy of $x_\ell$ at node $i$, and its updates require access to only the $i$th row of the matrix $P_\ell$. We refer to the updates in (2.39) as *TDA*. The proposed methodology is summarized in Algorithm 2. We define the *running local average* at node $i$, for player $\ell \in \{A, B\}$ as,

$$\hat{x}_{\ell,i}(t) \coloneqq \frac{1}{t} \sum_{s=0}^{t} x_{\ell,i}(s). \tag{2.41}$$

### *2.3.4 Main Results*

In this section, we present the main results of this setup. We first make the following definition that is used in the subsequent analysis.

**Definition 2.** *A network game is called Lipschitz convex if the cost of player $\ell \in \{A, B\}$ at node $i \in [n]$ satisfies,*

**Algorithm 2** Team-based Dual Averaging

**Inputs**: For player $\ell$

  Local black-box oracle at node $i$ to compute a subgradient of the local cost at any test point

  Doubly stochastic matrix $P_\ell$ induced by the network structure

**Outputs**:

  Estimates of NE $\hat{x}_{\ell,i}(t)$ at node $i$ for player $\ell$

**Initialize**:

  $t = 0$

  **For** Player $\ell \in \{A, B\}$ at node $i \in [n]$

    $z_{\ell,i}(0) = 0$

    Take random action $x_{\ell,i}(0)$

**while convergence**

  **For** Player $\ell \in \{A, B\}$ at node $i \in [n]$

    Observe the opponents local action and get

    $g_{\ell,i}(t) \in \partial_\ell f_{\ell,i}(x_{A,i}(t), x_{B,i}(t))$

  **For** Player $\ell \in \{A, B\}$ at node $i \in [n]$

    Update the dual variable $z_{\ell,i}(t+1)$ by (2.39)

    Calculate and take action $x_{\ell,i}(t+1)$ by (2.39)

  $t = t + 1$

**Return:** the *running average* $\hat{x}_{\ell,i}$ by (2.41)

- $f_{\ell,i}(\,.\,, x_{-\ell}) : \mathcal{X}_\ell \mapsto \mathbb{R}$ *is convex* $\forall x_{-\ell} \in \mathcal{X}_{-\ell}$.
- $f_{\ell,i}(\,.\,, x_{-\ell}) : \mathcal{X}_\ell \mapsto \mathbb{R}$ *is* $L_\ell$*-Lipschitz continuous* $\forall x_{-\ell} \in \mathcal{X}_{-\ell}$ *such that* $|f_{\ell,i}(x_\ell, x_{-\ell}) - f_{\ell,i}(x'_\ell, x_{-\ell})| \leq L_\ell \|x_\ell - x'_\ell\|$, $\forall x_\ell, x'_\ell \in \mathcal{X}_\ell$.

Distinct from optimization problems where the notion of "optimality" plays a central role, in a game setup, the objective is seeking an equilibrium rather than an "optimal" solution. In general, in order to incorporate the interactions of players in convergence analysis of NE-learning algorithms, monotonicity has been known as a useful sufficient condition for problem "regularity", originally due to the seminal work of Rosen [136]. In contrast, with no further regularity assumptions, we will prove a sub-linear regret bound for TDA algo-

rithm. Convergence of the action iterates to the unique NE can be established using similar monotonicity assumptions, and will be provided in our subsequent work.

### *2.3.5 Regret Analysis*

First, we introduce two results that facilitate the basic convergence analysis. Herein, we borrow some tools from prior works, as such, we only mention the key steps that distinguish our contribution in the game setup. Furthermore, it is worth mentioning that even though we analyze TDA algorithm with $\gamma_\ell(t) = 1$, judicious choice of $\gamma_\ell$ can in fact improve the corresponding convergence rate.

Let us now define $\tilde{f}_A(\,.\,;t)$ and $\tilde{f}_B(\,.\,;t)$ as,

$$\tilde{f}_A(\,.\,;t) \coloneqq \tfrac{1}{n}\textstyle\sum_{j=1}^n f_{A,j}(\,.\,, x_{B,j}(t)),$$
$$\tilde{f}_B(\,.\,;t) \coloneqq \tfrac{1}{n}\textstyle\sum_{j=1}^n f_{B,j}(x_{A,j}(t),\,.\,),$$

where $x_{A,i}(t)$ and $x_{B,j}(t)$ are the sequences generated by the TDA algorithm. Now, for $\ell \in \{A, B\}$, we define $\mathcal{R}_{\ell,i}$ to be the regret of player $\ell$ for implementing the TDA algorithm at node $i$ versus any fixed action $x^*_\ell$, while the other player has committed to implement the TDA, *i.e.*,

$$\mathcal{R}_{\ell,i}(T) \coloneqq \textstyle\sum_{t=1}^T \tilde{f}_\ell(x_{\ell,i}(t);t) - \sum_{t=1}^T \tilde{f}_\ell(x^*_\ell;t). \tag{2.42}$$

**Lemma 13.** *Following Algorithm 2, suppose that player $\ell \in \{A, B\}$ has access to $g_{\ell,i}$ for $(\ell, i) \in \mathcal{I}_\ell$ at each node $i \in [n]$, and $\{\alpha_\ell(t)\}_{t=0}^\infty$ is a non-increasing sequence of positive stepsizes and $\gamma_\ell(t) = 1$. Then for any Lipschitz convex network game we have,*

$$\mathcal{R}_{\ell,i}(T) \leq \tfrac{\psi(x^*_\ell)}{\alpha_\ell(T)} + \tfrac{L_\ell^2}{2}\textstyle\sum_{t=1}^T \alpha_\ell(t-1) + L_\ell \sum_{t=1}^T \alpha_\ell(t)\left[\mathcal{Z}_{\ell,i}(t) + \tfrac{2}{n}\sum_{j=1}^n \mathcal{Z}_{\ell,i}(t)\right],$$

*for any fixed $x^* = (x^*_A, x^*_B) \in \mathcal{X}_A \times \mathcal{X}_B$, where $\mathcal{Z}_{\ell,i}(t) = \|\bar{z}_\ell(t) - z_{\ell,i}(t)\|_*$ is the consensus error at each time with $\bar{z}_\ell(t) = \frac{1}{n}\sum_{i=1}^n z_{\ell,i}(t)$ as the averaging term in the dual space.*

*Proof.* For each team define,

$$y_\ell(t) \coloneqq \Pi^\psi_{\mathcal{X}_\ell}(-\bar{z}_\ell(t), \alpha(t)). \tag{2.43}$$

The doubly stochastic nature of $P_\ell$ implies an iterative form of $\bar{z}_\ell(t)$ as,

$$\bar{z}_\ell(t+1) = \bar{z}_\ell(t) + \tfrac{1}{n}\textstyle\sum_{j=1}^{n} g_{\ell,j}(t).$$

Hence, by the zero choice of the dual initialization we get,

$$y_\ell(t) = \text{argmin}_{x\in\mathcal{X}_\ell}\left\{ \sum_{s=1}^{t-1}\langle \frac{1}{n}\sum_{i=1}^{n} g_{\ell,i}(s), x\rangle + \frac{1}{\alpha_\ell(t)}\psi(x)\right\}.$$

Now, using the $L_A$-Lipschitz property of $f_A$ we can show that,

$$\mathcal{R}_{A,i}(T) \leq \textstyle\sum_{t=1}^{T}\left[\tilde{f}_A(y_A(t)) - \tilde{f}_A(x_A^*)\right] + \sum_{t=1}^{T} L_A\|y_A(t) - x_{A,i}(t)\|.$$

Also, by adding and subtracting $\sum_j f_{A,j}(x_{A,j}(t), x_{B,j}(t))$ to the first term and using convexity we have,

$$\begin{aligned}\mathcal{R}_{A,i}(T) \leq \textstyle\sum_{t=1}^{T}\frac{1}{n}\sum_{j=1}^{n}\langle g_{A,j}(t), x_{A,j}(t) - x_A^*\rangle \\ + \textstyle\sum_{t=1}^{T}\frac{L_A}{n}\sum_{j=1}^{n}\left[\|y_A(t) - x_{A,j}(t)\| + \|y_A(t) - x_{A,i}(t)\|\right],\end{aligned}$$

where $g_{A,j}(t) \in \partial_A f_{A,j}(x_{A,j}(t), x_{B,j}(t))$ and the $L_A$-Lipschitz condition is leveraged again. Then by adding and subtracting $y_A(t)$ in the inner-product we observe that,

$$\begin{aligned}\mathcal{R}_{A,i}(T) &\leq \textstyle\sum_{t=1}^{T}\frac{1}{n}\sum_{j=1}^{n}\langle g_{A,j}(t), y_A(t) - x_A^*\rangle \\ &+ \textstyle\sum_{t=1}^{T}\frac{L_A}{n}\sum_{j=1}^{n}\left[2\|y_A(t) - x_{A,j}(t)\| + \|y_A(t) - x_{A,i}(t)\|\right] \\ &\leq \textstyle\sum_{t=1}^{T}\langle\frac{1}{n}\sum_{j=1}^{n} g_{A,j}(t), y_A(t) - x_A^*\rangle \\ &+ \textstyle\frac{L_A}{n}\sum_{t=1}^{T}\alpha_A(t)\sum_{j=1}^{n}\left[2\mathcal{Z}_{A,j}(t) + \mathcal{Z}_{A,i}(t)\right],\end{aligned}$$

resulting from the $L_A$-Lipschitz property of $f_A$ and $\alpha$-Lipschitz continuity of the generalized projection $\Pi_{\mathcal{X}}^{\psi}(.,\alpha)$ (which is a direct consequence of Lemma 1 in [123]). Thereby, the lemma follows by applying Theorem 2 and Equation (3.3) in [123] (also restated as Lemma 3 in [58]) to the first term of the inequality above and using the fact that $\|g_{A,j}\|_* \leq L_A$. Similar analysis results in the bound for $\mathcal{R}_{B,i}(T)$. □

### *2.3.6 Choice of the Learning Rate $\alpha_\ell(t)$*

In order to achieve convergence, an appropriate choice of step-size (learning rate) $\alpha_\ell(t)$ is required. The next result shows how specific choices of $\alpha_\ell(t)$ result in practical bounds on $\mathcal{R}_{\ell,i}$ by essentially bounding the $\mathcal{Z}_{\ell,k}$ terms.

**Theorem 13.** *Under the notation adopted in Lemma 13, suppose that* $\psi(x_\ell^*) \leq R_\ell^2$. *Then by choosing the step-size,*

$$\alpha_\ell(t) = \frac{R_\ell\sqrt{1-\sigma_2(P_\ell)}}{\sqrt{13}L_\ell\sqrt{t}},$$

*for any Lipschitz convex network game we obtain,*

$$\mathcal{R}_{\ell,i}(T) \leq \sqrt{T}\log(T\sqrt{n})\frac{2\sqrt{13}R_\ell L_\ell}{\sqrt{1-\sigma_2(P_\ell)}}.$$

*Proof.* Stacking the updates of dual variables (2.39) into a matrix $Z_\ell = [z_{\ell,1} \ \dots \ z_{\ell,n}]$ and similarly $G_\ell = [g_{\ell,1} \ \dots \ g_{\ell,n}]$, for an undirected graph ($P_\ell^\top = P_\ell$) leads to,

$$Z_\ell(t+1) = Z_\ell(t)P_\ell + G_\ell(t).$$

Define $\Phi_\ell(t+1,s) = P_\ell^{t+1-s}$, where $\Phi$ is the transition matrix for a discrete-time linear system with state $Z$. Thus,

$$Z_\ell(t+1) = Z(s)\Phi_\ell(t+1,s) + \textstyle\sum_{r=s+1}^{t+1} G_\ell(r-1)\Phi_\ell(t+1,r),$$

for $0 \leq s \leq t$. Note that $\Phi_\ell(t+1,s)\mathbb{1} = \mathbb{1}$ and by definition $\bar{z}_\ell(t) = Z_\ell(t)\mathbb{1}/n$ and $z_{\ell,i}(t) = Z_\ell(t)e_i$, we obtain,

$$\bar{z}_\ell(t) - z_{\ell,i}(t) = \bar{z}_\ell(s) - Z(s)\Phi_\ell(t,s)e_i + \sum_{r=s+1}^{t} G_\ell(r-1)[\mathbb{1}/n - \Phi_\ell(t,r)e_i].$$

For simplicity we assume that $z_{\ell,i}(0) = \bar{z}_\ell(0)$ (say by choosing $z_{\ell,i}(0) = 0$); then we have $\bar{z}_\ell(s) - Z(s)\Phi_\ell(t,s)e_i = 0$ at $s = 0$. This implies that,

$$\bar{z}_\ell(t) - z_{\ell,i}(t) = \textstyle\sum_{r=1}^{t} G_\ell(r-1)[\mathbb{1}/n - \Phi_\ell(t,r)e_i].$$

We can then proceed to bound this error as,

$$\|\bar{z}_\ell(t) - z_{\ell,i}(t)\|_* \leq L_\ell \textstyle\sum_{r=1}^{t} \|\mathbb{1}/n - \Phi_\ell(t,r)e_i\|_1,$$

where we have used $\|g_{\ell,i}\|_* \leq L_\ell$ and norm inequalities. Consider the following standard inequality ([54]),

$$\|\mathbb{1}/n - \Phi_\ell(t,r)e_i\|_1 \leq \sqrt{n}\sigma_2(P_\ell)^{t+1-r}.$$

We say that $r$ is "small" if $r \leq t + 1 + \log(T\sqrt{n})/\log \sigma_2(P_\ell)$ (otherwise it is "large"). Then by splitting the sum, one would note that for the small $r$,

$$\|\mathbb{1}/n - \Phi_\ell(t, r)e_i\|_1 \leq \frac{1}{T},$$

since $\sigma_2(P_\ell) < 1$. Otherwise, for the large $r$,

$$\|\mathbb{1}/n - \Phi_\ell(t, r)e_i\|_1 \leq 2.$$

Then it can be shown that,

$$\mathcal{Z}_{\ell,i}(t) = \|\bar{z}_\ell(t) - z_{\ell,i}(t)\|_* \leq 2L_\ell \frac{\log(T\sqrt{n})}{1 - \sigma_2(P_\ell)}, \tag{2.44}$$

using $\log \sigma_2(P_\ell)^{-1} \geq 1 - \sigma_2(P_\ell)$. Hence from Lemma 13,

$$\mathcal{R}_{\ell,i}(T) \leq \frac{\psi(x_\ell^*)}{\alpha_\ell(T)} + \frac{L_\ell^2}{2} \sum_{t=1}^{T} \alpha_\ell(t-1) + \frac{6L_\ell^2 \log(T\sqrt{n})}{1 - \sigma_2(P_\ell)} \sum_{t=1}^{T} \alpha_\ell(t).$$

Define the sequence $\{\alpha(t)\}_{t=0}^{\infty}$ as, $\alpha_\ell(t) = K_\ell/\sqrt{t}$, $\alpha_\ell(0) = 1$. Since $\psi(x_\ell^*) \leq R_\ell^2$ and $\sum_{t=1}^{T} t^{-1/2} \leq 2\sqrt{T} - 1$, choosing $K_\ell = R_\ell\sqrt{1 - \sigma_2(P_\ell)}/\sqrt{13}L_\ell$ completes the proof. □

### *2.3.7 Convergence of TDA Algorithm*

A natural question on the performance of the TDA algorithm pertains to the convergence of the corresponding action iterates. However, it is known that the convergence of action iterates for this general class of algorithms cannot be guaranteed without further regularity assumptions. Nevertheless, with minimal continuity assumptions, our next result ensures that the point of convergence of TDA algorithm is in fact a NE. First a relevant definition.

**Definition 3.** *A network game is called continuous if the cost of player $\ell \in \{A, B\}$ at node $i \in [n]$ satisfies,*

- *$f_{\ell,i}(x_\ell, x_{-\ell})$ is continuously differentiable in $x_\ell$,*
- *$f_{\ell,i}(x_\ell, x_{-\ell})$ and $\nabla f_{\ell,i}(x_\ell, x_{-\ell})$ are both continuous in the joint variable $(x_\ell, x_{-\ell})$.*

**Theorem 14.** *Under the notation adopted in Theorem 13, for a continuous network game, if Algorithm 2 converges, i.e., $x_{\ell,i}(t) \to x_{\ell,i}^*$ as $t \to \infty$, then $x_{\ell,i}^* = x_\ell^*$ for all $i \in [n]$ and $(x_\ell^*, x_{-\ell}^*)$ is a NE.*

*Proof.* By continuous differentiability, we have that $g_{\ell,i} = \nabla f_{\ell,i}$ for all $i$ and $g_{\ell,i} \to g^*_{\ell,i}$ due to the joint continuity. Now by $\alpha_\ell$-Lipschitz continuity of the generalized projection we have,

$$\|x_{\ell,i}(t) - x_{\ell,j}(t)\| \leq \alpha_\ell(t) \left[\mathcal{Z}_{\ell,i}(t) + \mathcal{Z}_{\ell,j}(t)\right].$$

From (2.44) and the choice of $\alpha_\ell(t) = K_\ell/\sqrt{t}$ we conclude that for all $i, j$, $\|x_{\ell,i}(t) - x_{\ell,j}(t)\| \to 0$ as $t \to \infty$, and thus $x^*_{\ell,i} = x^*_\ell$ for all $i$. Similarly, this implies $y_\ell(t) \to x^*_\ell$ since,

$$\|y_\ell(t) - x^*_\ell\| \leq \alpha_\ell(t)\mathcal{Z}_{\ell,i}(t) + \|x_{\ell,i}(t) - x^*_\ell\|.$$

The rest of the proof is by contradiction. Suppose $x^* = (x^*_\ell, x^*_{-\ell})$ is not a NE and define $G^*_\ell = G_\ell(x^*)$. Then by definition of NE in our setup and noting that $\nabla_\ell f_\ell(x^*) = G^*_\ell \mathbb{1}/n$, at least for one of the players (say player $\ell$) we have the following (Proposition 1.4.2 in [62]),

$$\exists q_\ell \in \mathcal{X}_\ell \quad \text{s.t.} \quad \langle G^*_\ell \mathbb{1}/n, q_\ell - x^*_\ell \rangle < 0.$$

By continuity there exist a constant $c > 0$ and neighborhoods $U, V$ of points $x^*_\ell$, $G^*_\ell \mathbb{1}/n$, respectively, such that,

$$\langle G'_\ell \mathbb{1}/n, q_\ell - x'_\ell \rangle \leq -c, \quad \forall x'_\ell \in U, \quad \forall G'_\ell \text{ s.t. } G'_\ell \mathbb{1}/n \in V.$$

On the other hand, by definition of $\Pi^\psi_{\mathcal{X}}$ and $y_\ell$ as in (2.43), and strong convexity of $\psi$ we can conclude that $-\alpha_\ell(t)\bar{z}_\ell(t) \in \partial\psi(y_\ell(t))$, and therefore,

$$\psi(q_\ell) - \psi(y_\ell(t)) \geq -\alpha_\ell(t)\langle \bar{z}_\ell(t), q_\ell - y_\ell(t) \rangle.$$

Note that by convergence of $y_\ell(t)$, there exists $N$ such that $y_\ell(t) \in U$ and $G_\ell(t)\mathbb{1}/n \in V$, $\forall t \geq N$. Furthermore, $\bar{z}_\ell(t) = \sum_{r=1}^{t-1} G_\ell(r)\mathbb{1}/n$ since $z_{\ell,i}(0) = \bar{z}_\ell(0)$; thus we can conclude that,

$$\psi(q_\ell) - \psi(y_\ell(t)) \geq \alpha_\ell(t) \sum_{r=N}^{t-1} c - \alpha_\ell(t)\langle \sum_{r=1}^{N-1} G_\ell(r)\mathbb{1}/n, q_\ell - y_\ell(t) \rangle.$$

Now as $t \to \infty$ the right hand side of the above inequality approaches positive infinity, which is a contradiction. □

### *2.3.8 Example*

In this section we illustrate the performance of our method for a two-team game on a network where each player has an objective at each node as follows,

$$f_{\ell,i}(x_\ell, x_{-\ell}) = \frac{1}{2}\left\|x_\ell - a_{\ell,i}\right\|^2 + \frac{1}{4}\langle x_\ell, x_{-\ell} - b_{\ell,i}\rangle,$$

where $a_{\ell,i}$ and $b_{\ell,i}$ are arbitrary prescribed parameters. Although the cost functions are only locally Lipschitz, they can be treated as a Lipschitz continuous function over any bounded (potentially large) domain. We have simulated the performance of TDA over complete, random 6-regular, and cycle graphs each consisting of 50 nodes. Figure 2.10(b) shows the random 6-regular network used for our simulations. To illustrate the advantage of dual averaging in our algorithm, we compare the results with another distributed algorithm—referred to as *TMD*—which is an extension of *Distributed Mirror Descent* algorithm [55] adopted for our game model with $\psi(.) = \frac{1}{2}\left\|.\right\|^2$. Also, as in standard form, the step-size of this algorithm $\beta_\ell(t) = K_\ell/t^{0.8}$ is chosen to be not summable but square summable where $K_\ell$ is as defined in Theorem 13. The TMD algorithm is detailed below with the same initialization as TDA. At iteration $t$, members of each team at each node observe the opponent's action $x_{-\ell,i}(t)$ locally and compute the local average estimate $v_{\ell,i}(t)$ as,

$$v_{\ell,i}(t) = \textstyle\sum_{j\in\mathcal{N}_{\ell,i}} P_{\ell,ij} x_{\ell,j}(t),$$

and get $g_{\ell,i}(t) \in \partial_\ell f_{\ell,i}(v_{\ell,i}(t), x_{-\ell,i}(t))$. Next, they project the average estimates back to the set of actions as,

$$x_{\ell,i}(t+1) = \mathrm{Proj}_{\mathcal{X}_\ell}[v_{\ell,i}(t) - \beta_\ell(t) g_{\ell,i}(t)].$$

Next, we define the NAE as the normalized mean of the running average error from NE over all nodes of the network as follows,

$$\mathrm{NAE}(t) = \frac{\sum_{i=1}^n \left\|\hat{x}_{A,i}(t) - x_A^*\right\| + \left\|\hat{x}_{B,i}(t) - x_B^*\right\|}{\sum_{i=1}^n \left\|\hat{x}_{A,i}(0) - x_A^*\right\| + \left\|\hat{x}_{B,i}(0) - x_B^*\right\|}. \tag{2.45}$$

Figure 2.9 shows the NAE at each iteration for both TDA and TMD algorithms. It can be noted that on networks with the same structure, the TDA method has a much better convergence rate as compared with TMD, primary due to the dual averaging nature of TDA.

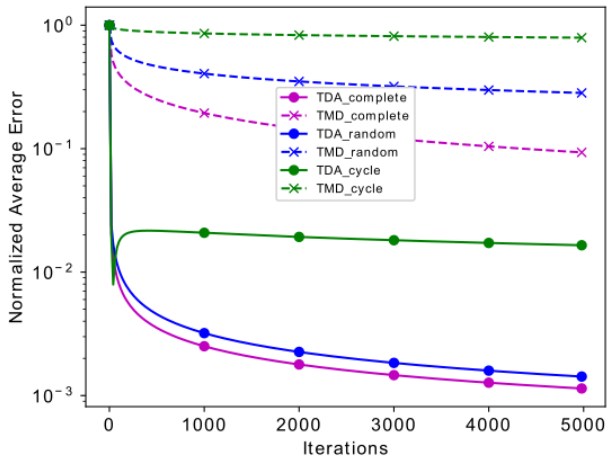
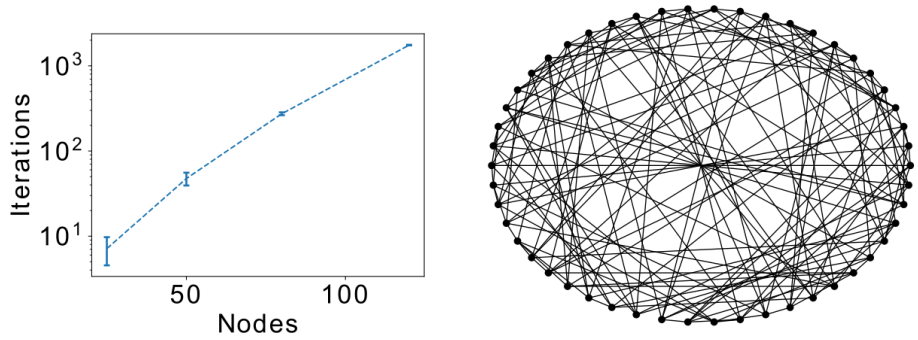



Figure 2.9: NAE at each iteration for both TDA and TMD algorithms in complete, random 6-regular, and cycle graphs with 50 nodes.

Clearly, as the connectivity of the network decreases from complete to cycle, convergence rate becomes slower. This is captured in Theorem 13 where lower connectivity in the network leads to an increase of $\sigma_2(P_\ell)$ closer to 1, resulting in a larger bound for each time horizon $T$. Next, we examine the performance of TDA in terms of the required iteration for a given error (from the NE). To do so, we consider four complete graphs with 25, 50, 80, and 120 nodes and require the algorithm to achieve a NAE of less than 0.1. Figure 2.10 illustrates the number of iterations $T$ needed for each network to achieve $\text{NAE}(T) < 0.1$. Since, the initialization of the algorithm is random, for each network, we have illustrated the mean and variance of the number of iterations required by 30 realizations. It is evident that the number of required iterations to achieve the same error-bound, increases exponentially in this simulation. This behavior is partially due to the fact that TDA is a first-order method in a game setting and it is converging towards an equilibrium point, unlike distributed optimization scenario, where convergence is towards an attractive optimal point.

## *2.4 Layered Networks*

In this part of the thesis, we consider a specific structure of networked systems that we refer to as *layered networks* and their applications to real systems and, in particular, so-




(a) (b)

Figure 2.10: (a) Number of iterations needed for each network so that NAE is less than 0.1, (b) The random 6-regular network with 50 nodes simulated in the example.

cial networks. Communities are formed by a large number of local and global interactions, linked by a wide range of social and economic interdependencies. Systematic understanding of the evolution of communities can be achieved by understanding the influence of its members on each other as well as the role of external factors. Such an understanding often requires an accurate model that captures the interactive behaviors [134]. With the introduction of *sociogram* in 1930's [118], graphical models of interconnections among a group of individuals was adopted to examine the evolution of communities. This line of work led to several branches in social and behavioral research such as the interdisciplinary science of Social Networks Analysis (SNA). The development of these dynamic models has provided an intellectual bridge between the communal social and behavioral interdependencies on one hand, and techniques in system sciences such as control and estimation, on the other [44]. In the meantime, the advent of large-scale modeling techniques due to complexities of the interdependencies, increasing population size, and the corresponding datasets, have led to the need to revisit algorithms and solution strategies for network-level control and estimation.

In spite of the complexity and unpredictability of large-scale social interactions, characterizing reliable models for these interactions are promising in cases where prior knowledge about the underlying structures of these systems is available. In particular, for certain types

of social networks, the layering structure allows a compositional approach for the mathematical representation of the system. The layering structure in a social network can be induced by a variety of motives such as the presence of distinct social types, geographical coordinates, and financial or political ties. The idea of compositional study of a layered system can be compared to distributed systems analysis in the sense that the problem is split into manageable subproblems that can be subsequently solved independently [18, 24]. For example, decomposition of consensus-type networks leads to examining the protocol for each layer [124]. Furthermore, [39] provides a controllability and observability analysis on large-scale composite networked systems based on their factors.

Despite the many advantages of a decompositional approach, the high dimension of the system poses new challenges primary due to the layers' uncertainty as well as perturbations to the layering structure as a whole [71]. In the context of social networks these uncertainties may be due to inaccurate modeling of the nature of the interactions as well as whether or not two social entities are directly interdependent. Such uncertainties pose difficulties for the control and estimation of such systems. As an example, the adoption of a linear quadratic (LQ) theory in social networks, is not only hindered by high dimensionality, but also by inherent model uncertainties. As a result, the strong robustness properties of say, the Linear Quadratic Regulator (LQR) approach, can vanish where small changes to the system parameters lead to instabilities. We present a compositional method to characterize performance guarantees on layered social networks with model uncertainties. The corresponding distributed analysis and control presented here is closely related to [37], where a composite LQR solution is derived from the parameters of the two layers. Here, we obtain sufficient conditions for the robust stability of the composite network based on a layered control mechanism.

### *2.4.1 Problem Setup*

Specific classes of large-scale social networks can be modeled, at least approximately, via a layered structure representing interdependent subsystems. One may then aim to characterize the properties of the system via those of its factors or layers. This decompositional

approach is effective for various classes of social and economic networks, where for example, inter-nodal influences among distinct groups lead to opinions on a sequence of issues [90] (Fig. 2.11). The evolutionary study of interconnections among political parties in elections or the investigation of financial ties between different branches of an international organization are two examples of the layered structures in behavioral sciences. This types of system representation make it possible to embed more structure into the system and use this embedding to simplify the subsequent computational and theoretic analysis.

### *2.4.2 Dynamics of each layer of the network*

We assume that the evolution of opinions is captured by Taylor's model of influenced attitude change [159]. The model considers the change in attitudes of a set of individuals as a result of influence processes within the set, as well as the exposure to external sources. Based on this model, the opinion dynamics of an individual $p$ in a networked system with $n$ agents and $m$ external inputs can be represented as,

$$\dot{y}_p(t) = \sum_{q=1}^{n} a_{pq}(y_q(t) - y_p(t)) + \sum_{k=1}^{m} b_{pk}(s_k - y_p(t)),$$

where $y_p$ is the state of $p$, $s_k$ is the $k$th external input, $a_{pq}$ captures the influence between agents $p$ and $q$ and $b_{pk}$ defines the interaction between agent $p$ and the $k$th external (static) input source. If the individual $r$ is not directly influenced by an external input, $b_{rk} = 0$ for all $k$. Particularly, the input to a social organization may be due to a stationary source of communication such as mass media or an influential administrative center. In the matrix

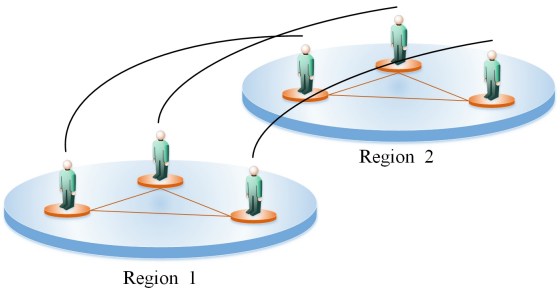



Figure 2.11: A example of a layered social networks due to geographical distributions

form, Taylor's dynamical model for layer $i$ of the network assumes the form,

$$\dot{x}_i(t) = A_i x_i(t) + B_i u_i(t) \quad \forall i \in \{1, 2, \ldots, m\},$$

where $A_i \in \mathbb{R}^{n_i \times n_i}$ is equivalent to graph *Laplacian*,[2] capturing the difference in attitudes and $B_i \in \mathbb{R}^{n_i \times p_i}$ defines the control "knob" for the external inputs. Following [39], for this type of dynamics,

$$A(\overset{m}{\underset{i=1}{\square}} \mathcal{G}_i) = \overset{m}{\underset{i=1}{\oplus}} A_i,$$

where for simplicity $A_i = A(\mathcal{G}_i)$. Then the overall network is formed as,

$$\dot{x}(t) = \big(\overset{m}{\underset{i=1}{\oplus}} A_i\big)x(t) + \big(\overset{m}{\underset{i=1}{\otimes}} B_i\big)u(t), \tag{2.46}$$

where $\overset{m}{\underset{i=1}{\oplus}} A_i \in \mathbb{R}^{\prod_{i=1}^{m} n_j \times \prod_{i=1}^{m} n_j}$ and $\overset{m}{\underset{i=1}{\otimes}} B_i \in \mathbb{R}^{\prod_{i=1}^{m} n_j \times \prod_{i=1}^{m} p_j}$. This dynamics can also be formulated in discrete-time as well [39].

### *2.4.3 Guaranteed LQ Performance*

Perturbations can be induced in social networks due to distortions in existence, nature, or intensity of interactions among the individuals. This model uncertainty can eventually lead to instabilities in a social influence model. Unstable behavior in social networks generally have unfavorable ramifications such as the advent of clustering or community cleavage [15, 45, 68].

Here, we employ the LQ theory as a potential methodology to design state-feedback controllers for systems with layered structures. LQ methods have been applied in the literature for the control of large-scale systems and social networks [160, 174]. In general, the applicability of the LQ framework is reasonable when the resources used for social influence are restricted.[3] In such a setting, an LQ regulator can be used to attenuate the effect of the undesirable external influences through minimal adjustments in the control variables. However, it is well-known that the stability margins of the LQ design do not guarantee

[2]The notation is due to the applicability of the methods in the current part for any general linear dynamics and shall not be confused with the adjacency matrix $\mathcal{A}$ of the network.

[3]For example, when there is cap on the advertisement budget.

robustness to variations in system parameters [35]. It is thus desirable to enhance the LQ design in order to obtain guarantees on the stability and performance of the system. One approach to achieve this is through extending the *Algebraic Riccati Equation* (ARE); the baseline form of the setup assumes the form,

$$A^\top P + PA^\top + Q - PBR^{-1}B^\top P = 0, \tag{2.47}$$

for given $Q \succeq 0, R \succ 0$, leading to the LQR optimal controller as $K = R^{-1}B^\top P$. It is known that robustness to variations in system parameters can be handled by an additional term in (2.47). We will utilize this methodology to ensure guarantees on the large-scale system performance in presence of uncertainties. Accordingly, appending the uncertainties to the dynamics in (2.46) results in,

$$\dot{x}(t) = \overset{m}{\underset{i=1}{\oplus}} \big(A_i + \Delta A_i\big)x(t) + \big(\overset{m}{\underset{i=1}{\otimes}} B_i\big)u(t),$$

where $\Delta A_i$ denotes the uncertainty of the model in layer $i$. There are many different structures suggested for $\Delta A_i$ in the literature [56]. One common choice of these structured perturbations is,

$$\Delta A = \sum_{j=1}^{d} w_j \tilde{A}_j, \tag{2.48}$$

for given $\tilde{A}_j$. Nevertheless, the results are derived for the general form of the uncertainty $\Delta A$ of each layer.

The layered structure is one example where compositional control is feasible by applying similar inputs to the network layers. In this case, the generalized input matrix can be written as,

$$B_\otimes = B_1 \otimes I_{n_2} \otimes \cdots \otimes I_{n_m}.$$

This assumption helps reduce the intra-layered couplings. Hence the main analysis here is building upon the following generalized dynamics,

$$\dot{x}(t) = \overset{m}{\underset{i=1}{\oplus}} \big(A_i + \Delta A_i\big)x + B_\otimes u(t). \tag{2.49}$$

Our goal is to find a generalized structured controller to achieve an upper bound on the LQR performance index for system (2.49).

### *2.4.4 Analysis*

In this section we propose a framework for guaranteed performance design for the $m$-layered dynamical system (2.49) using an LQ approach. We restate a theorem from [56] that our main result is built upon. We then generalize the sufficient conditions for the layered case.

**Theorem 15.** *[56] Consider the perturbed dynamical system,*

$$\dot{x} = (A + \Delta A)x + Bu,$$

*and define the quadratic performance measure,*

$$J = \int_0^\infty \big(x^\top Q x + u^\top R u\big) dt. \tag{2.50}$$

*Let $x_0$ be the initial state and $P \succ 0$ be the solution to the modified ARE,*

$$A^\top P + PA + Q - PBR^{-1}B^\top P + \mathcal{U}(P) = 0, \tag{2.51}$$

*where $\mathcal{U}(.)$ is a positive symmetric function for which,*

$$\Delta A^\top P + P\Delta A \preceq \mathcal{U}(P). \tag{2.52}$$

*Then the feedback control law defined as $u = -Kx$ leads to $J \leq x_0^\top P x_0$, where $K = R^{-1}B^\top P$.*

In Theorem 15, choosing $\mathcal{U}$ is dictated by a trade-off between the complexity of the design and analytical properties of the solution strategy. The choice, however, depends on the nature of the perturbations. For instance, the structure given in (2.48) implies that,

$$\Delta A^\top P + P\Delta A = \sum_{j=1}^{d} w_j \big(\tilde{A}_j^\top P + P\tilde{A}_j\big).$$

One suggested form of $\mathcal{U}$ induced by this type of perturbation is [35],

$$\mathcal{U}(P) = \sum_{j=1}^{d} Q_j |\Lambda_j| Q_j^\top, \tag{2.53}$$

where $Q_j$ and $\Lambda_j$ are obtained from the eigendecomposition of the symmetric matrix $\tilde{A}_j^\top P + P\tilde{A}_j$ as,

$$\tilde{A}_j^\top P + P\tilde{A}_j = Q_j \Lambda_j Q_j^\top.$$

It is straightforward to check that (2.52) holds under such a definition of $\mathcal{U}$. Nonetheless, the analysis is not limited to any specific types of $\mathcal{U}$.

**Definition 4.** *Given matrices $D$ and $C_i$ for $i \in \{1, 2, \ldots, \ell\}$, we define,*

$$C_D^{\otimes_{k,\ell}} = C_1 \otimes \cdots \otimes C_{k-1} \otimes D \otimes C_{k+1} \otimes \cdots \otimes C_\ell, \tag{2.54}$$

*i.e., $C_D^{\otimes_{k,\ell}}$ replaces $C_k$ with $D$ in $\overset{\ell}{\underset{i=1}{\otimes}} C_i$.*

Definition 4 is followed by some useful properties that is presented in the following.

**Proposition 1.** *Given Definition 4, the following hold:*

1. $X_Y^{\otimes_{k,\ell}} \pm X_Z^{\otimes_{k,\ell}} = X_{Y\pm Z}^{\otimes_{k,\ell}}$,
2. $(X_Y^{\otimes_{k,\ell}})(V_W^{\otimes_{k,\ell}}) = (XV)_{(YW)}^{\otimes_{k,\ell}}$,
3. $(X_Y^{\otimes_{k,\ell}})^\top = (X^\top)_{Y^\top}^{\otimes_{k,\ell}}$,
4. $(X_Y^{\otimes_{k,\ell}})^{-1} = (X^{-1})_{Y^{-1}}^{\otimes_{k,\ell}}$,

*where with a slight abuse of notation,*

$$(XV)_{(YW)}^{\otimes_{k,\ell}} = X_1V_1 \otimes \cdots \otimes X_{k-1}V_{k-1} \otimes YW \otimes X_{k+1}V_{k+1} \otimes \cdots \otimes X_\ell V_\ell.$$

**Lemma 14.** *The dynamics in (2.49) can be written as,*

$$\dot{x} = \big(A_\oplus + \Delta A_\oplus\big)x + B_\otimes u, \tag{2.55}$$

*where $A_\oplus = \overset{m}{\underset{i=1}{\oplus}} A_i$ and $\Delta A_\oplus = \overset{m}{\underset{i=1}{\oplus}} \Delta A_i$.*

The proof of Lemma 14 is straightforward using induction and the properties of Kronecker products. We assume that the generalized perturbation $\Delta A_\oplus$ represents a structured uncertainty composed of the perturbations from each layer of the system. Generalization of a layer-independent perturbation or leveraging other well-known uncertainty structures is out of the scope of this thesis and are addressed for future works.

**Theorem 16.** *Consider the generalized dynamics in (2.55). Assume that $Q_1 \succeq 0$, $R_1 \succ 0$, and symmetric positive function $\mathcal{U}_1$ is given such that,*

$$\Delta A_1^\top \bar{P} + \bar{P}\Delta A_1 \preceq \mathcal{U}_1(\bar{P}), \tag{2.56}$$

*holds for all $\bar{P} \succ 0$ and $\Delta A_1$. Furthermore, let $P_1 \succ 0$ be the solution to,*

$$A_1^\top P_1 + P_1 A_1 + Q_1 - P_1 B_1 R_1^{-1} B_1^\top P_1 + \mathcal{U}_1(P_1) = 0,$$

*and define,*

$$F_i = A_i^\top M_i + M_i A_i, \qquad G_i = \Delta A_i^\top M_i + M_i \Delta A_i,$$

*where $M_i \succ 0$ is such that $F_i \preceq 0$ and $G_i \prec 0$ for all $i = 2, \ldots, m$. Then the generalized state-feedback control law $u = -K_\otimes x$ with $K_\otimes = K_1 \otimes I_{n_2} \otimes \cdots \otimes I_{n_m}$ implies that,*

$$\bar{J} = \int_0^\infty \big(x^\top Q_\otimes x + u^\top R_\otimes u\big) dt \;\; \leq \;\; x_0^\top P_\otimes x_0, \tag{2.57}$$

*where $P_\otimes$, $Q_\otimes$, and $R_\otimes$ are defined as,*

$$P_\otimes = M_{P_1}^{\otimes_{1,m}}, \qquad R_\otimes = M_{R_1}^{\otimes_{1,m}}, \qquad Q_\otimes \qquad = M_{Q_1}^{\otimes_{1,m}} - P_1 \otimes \big(\sum_{i=2}^{m} M_{F_i}^{\otimes_{i,m}}\big).$$

*Proof.* We proceed by checking the conditions of Theorem 15 but for the layered system in (16). To this end, we need a new definition for a symmetric positive function that generalizes $\mathcal{U}$. Let,

$$\mathcal{V}(T_1, T_2, \ldots, T_\ell) = T_{\mathcal{U}_1(T_1)}^{\otimes_{1,\ell}}. \tag{2.58}$$

Then we note that for $m = 2$,

$$\begin{aligned}
\Delta A_\oplus^\top P_\otimes &= (\Delta A_1 \oplus \Delta A_2)^\top (P_1 \otimes M_2) \\
&= (\Delta A_1^\top \otimes I_{n_2} + I_{n_1} \otimes \Delta A_2^\top)(P_1 \otimes M_2) \\
&= \Delta A_1^\top P_1 \otimes M_2 + P_1 \otimes \Delta A_2^\top M_2 \; .
\end{aligned}$$

Similarly,

$$P_\otimes \Delta A_\oplus = P_1 \Delta A_1 \otimes M_2 + P_1 \otimes M_2 \Delta A_2.$$

Hence by induction, it can be shown that for any $m$,

$$\begin{aligned}
&\Delta A_{\oplus}^{\top} P_{\otimes} + P_{\otimes} \Delta A_{\oplus} \\
&\quad = \left(\Delta A_1^{\top} P_1 + P_1 \Delta A_1\right) \otimes M_2 \otimes \cdots \otimes M_m \\
&\qquad\qquad + P_1 \otimes \Big(\sum_{i=2}^{m} M_{F_i}^{\otimes_{i,m}}\Big) \qquad (2.59) \\
&\quad \preceq \ \mathcal{U}_1(P_1) \otimes M_2 \otimes \cdots \otimes M_m \\
&\quad = \ \mathcal{V}(P_1, M_2, \ldots, M_m),
\end{aligned}$$

where we have used (2.56) and the fact that Kronecker products preserve positive-definiteness. Also from Proposition 1.3, it is straightforward to show that $\mathcal{V}$ is a symmetric and positive. From Proposition 1.4, $R_{\otimes}^{-1} = (M^{-1})_{R_1^{-1}}^{\otimes_{1,m}}$; hence,

$$P_{\otimes} B_{\otimes} R_{\otimes}^{-1} B_{\otimes}^{\top} P_{\otimes} = M_{P_1 B_1 R_1^{-1} B_1^{\top} P_1}^{\otimes_{1,m}}, \qquad (2.60)$$

which gives,

$$\begin{aligned}
&A_{\oplus}^{\top} P_{\otimes} + P_{\otimes} A_{\oplus} + Q_{\otimes} \\
&\qquad - P_{\otimes} B_{\otimes} R_{\otimes}^{-1} B_{\otimes}^{\top} P_{\otimes} + \mathcal{V}(P_1, M_2, \ldots, M_m) \\
&\qquad = M_{A_1^{\top} P_1 + P_1 A_1 + Q_1 + P_1 B_1 R_1^{-1} B_1^{\top} P_1 + \mathcal{U}_1(P_1)}^{\otimes_{1,m}} = 0,
\end{aligned}$$

and from proposition 1.2,

$$K_{\otimes} = -R_{\otimes}^{-1} B_{\otimes}^{\top} P_{\otimes} = K_1 \otimes I_{n_2} \otimes \cdots \otimes I_{n_m}.$$

□

There are some remarks needed in relation to Theorem 16. First, from the definitions of $F_i$ and $G_i$, the perturbed dynamics $A_i + \Delta A_i$ is implicitly assumed to be stable which is not necessarily required. In this sense, the assumptions $M_i \succ 0$, $F_i \preceq 0$, and $G_i \prec 0$ might be restrictive. Indeed, we need $M_i$'s to be selected in a way that inequalities such as (2.59) and $Q_{\otimes} \succeq 0$ hold which may require further assumptions on the structure of $M_i$ such as being diagonal or sparse. This also limits the freedom of the designer to only select the matrices $M_i$ while forming the cost of the LQR problem.

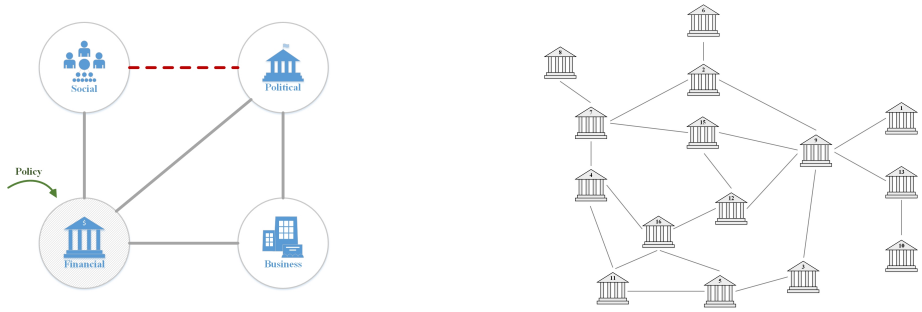
Moreover, we have assumed a layered structure for the controller where the input to the first layer is repeated in the subsequent layers reflected into the Kronecker structure. While this assumption reduces system couplings, the presence of the other layers' dynamics is implicit in parameters $F_i$ and $G_i$. Finally, the proposed $Q_\otimes$ essentially removes the couplings of the dynamics of different layers that shows up in $A_\oplus^\top P_\otimes + P_\otimes A_\oplus$ in the problem formulation. However, it needs to be verified whether this $Q_\otimes$ satisfies the existence and stabilizability criteria of the LQR solution. To that end, it is straightforward to check that $Q_\otimes \succeq 0$; in fact $Q_\otimes = L^\top L$ where,

$$L = \begin{bmatrix} D \otimes M_2^{1/2} \otimes M_3^{1/2} \otimes \cdots \otimes M_m^{1/2} \\ H \otimes N_2 \otimes M_3^{1/2} \otimes \cdots \otimes M_m^{1/2} \\ H \otimes M_2^{1/2} \otimes N_3 \otimes \cdots \otimes M_m^{1/2} \\ \vdots \\ H \otimes M_2^{1/2} \otimes M_3^{1/2} \otimes \cdots \otimes N_m \end{bmatrix},$$

and $Q_1 = D^\top D$, $P_1 = H^\top H$, and $A_i^\top M_i + M_i A_i = N_i^\top N_i$ by Cholesky decomposition. Hence to obtain the stability of the generalized LQR solution, we need the implicit assumption that $(A_\oplus, L)$ is observable (via proper choices of $M_i$'s) and the controllability of $(A_\oplus, B_\otimes)$ (discussed in [39]).

### *2.4.5 Example: Compositional Synthesis for Social Networks*

Layered networks can be used for modeling geographical distribution of various social types. In this section we implement the guaranteed-cost compositional design on a social influence network. This case study is inspired by Padgett's research on 15 elite families in 1282-1500, Florence [127] and the impact of Renaissance on Italian art and culture in the same time interval. The analysis provides a grouping of these families into social, political, business, and financial members and the interactions between families were limited to these corresponding members. Based on the geographical distribution and ties between these families, we leverage our methodology to model this multi-layered network. Inherently, modeling such an organization is challenging due to the complexity of societal interactions as well as the population size. We account for these types of uncertainties in parameterizing the net-




(a) (b)

Figure 2.12: The elite family layered structures (a) interconnections within each family (b) connections among all families

work dynamics. These uncertainties can potentially lead to misclassification in the nature of connections among individuals. Our model contains three different layers: The first layer, $\mathcal{G}_1$, represents the structure of each family containing the four groups (Fig. 2.12a). The dashed line denotes a negative edge denoting a disagreement between social and political entities. An input to the financial member of each family is considered in order to both react to a change in fiscal strategy in response to Renaissance fluctuations and avoid social cleavage due to the opposition between two main members of the family.[4] All connections are assumed to be equal (not weighted). The perturbation to the system comes from a mistakenly flipped sign of the connection between social and financial groups. This results in clustering leading to the instability of the system. The Florentine elite families graph, $\mathcal{G}_2$, designates the second layer of the network (Fig. 2.12b). The third layer is inspired by the spread of Renaissance throughout other provinces of Italy such as Rome and Venice (Fig. 2.13). This extra layer signifies the computational efficiency of the method. We use Taylor's model of opinion evolution. In particular, we leverage Equation (2.49) to model

[4]In LQR terminology, we only have access to the financial control knob to bound the system performance. This is just a simplified assumption and the control can take place on every node.

this 3-layer dynamics as,

$$\dot{x} = -\left[\bigoplus_{i=1}^{3}(A_i + \Delta A_i)\right] x + \left(\bigotimes_{i=1}^{3} B_i\right) u,$$

where $A_i$ and $\Delta A_i$ denote the Laplacian and the uncertainty matrices of layer $i$. We assume $\Delta\mathcal{L}_2 = 0$, $\Delta\mathcal{L}_3 = 0$, and $B_1 = [0 \quad 0 \quad 0 \quad 1]^\top$ reflecting the control over the financial node. We use (2.48) to model the perturbation with $d = 1$, $w_1 = 2$, and $\tilde{A}_1 = e_1 e_2^\top + e_2 e_1^\top$, *i.e.*, a change in the sign between social and political groups. Fig. 2.14a depicts the instability of the system when the baseline LQR algorithm is used without taking the uncertainties into account. Fig. 2.14b shows the guaranteed performance for a similar setup but with an updated LQ controller design methodology presented in the analysis. Table 2.1 shows the time it takes to run the LQR algorithm (updated ARE in particular) based on the size of $\mathcal{G}_3$ (number of provinces). Similar results can also be obtained for $\mathcal{G}_1$ and $\mathcal{G}_2$.

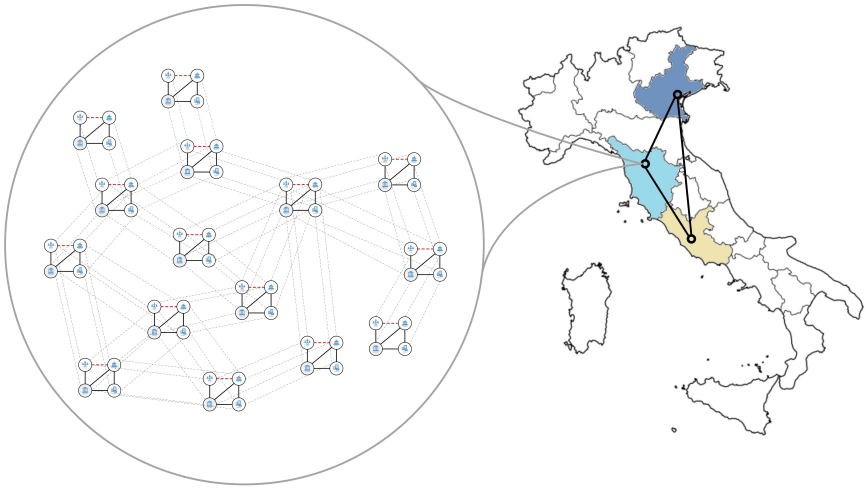
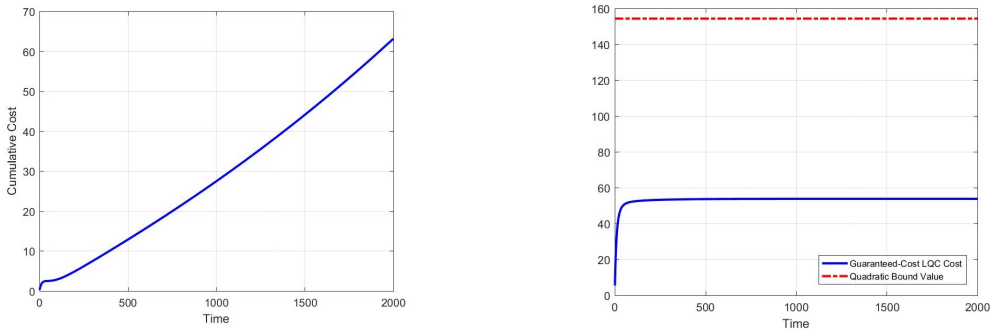


Figure 2.13: Composition of the elite families network layers.




(a) (b)

Figure 2.14: System performance (a) Baseline LQR (b) Our method

| Size of $\mathcal{G}_3$ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Time (sec) | 0.3 | 2 | 216 | 419 |

Table 2.1: Computational performance for solving modified ARE for different sizes of $\mathcal{G}_3$

# Chapter 3

# NETWORKS AS SYSTEM-OF-SYSTEMS

In this chapter, we continue our investigation on networked system with a generalization. Unlike the previous chapter, that each node in the network belongs to one agent, in this part, we assume that each node of the graph hosts a linear dynamical system itself. Hence, now the information exchange throughout the network is between the states and possibly the costs of the interconnected systems. We refer to this type of mechanism *distributed control of interconnected systems.*

Distributed control of large-scale systems is a well-established area of research. The roots of the field trace back to the socioeconomics literature of 1970's [113] and early works in the control literature began to emerge later in the same decade [167]. The inspiration of these types of works were that the presupposition of centrality fails to hold due to the lack of either a central intelligence or computational capability [89, 139]. Subsequently, stability conditions were derived for multi-channel linear systems [43], solidifying the research one further step. Fast forward a few decades, sufficient graph-theoretic conditions were provided for stability of formations comprised of identical vehicles [65] and, along the same lines, graph-based distributed controller synthesis was further examined independently in works such as [24, 52, 101, 111, 112, 168]. The topic was also studied from a spatially distributed control viewpoint [19, 120] and a compositional layered design approach [10, 37]. Moreover, from an agent-level perspective, the problem has been tackled for both homogeneous systems [24, 112, 168] and more recently heterogeneous ones [153].

The complete knowledge of the underlying system model is a common assumption in the literature on distributed control, where the goal is to find a distributed feedback mechanism that follows an underlying network topology. However, derivation of models from first principles could be restrictive when the system is highly complex and uncertain. Such restrictions also hold for parametric perturbations that occur due to inefficient modeling or

other unknown design factors.[1] Robust synthesis approaches could alleviate this issue if the perturbations follow specific models, in both centralized [94] and distributed [104] cases. However, if the original estimates of system parameters are faulty or the perturbations violate the presumed model, then both stability and optimality of the proposed feedback mechanisms will be shattered. Data-driven control, on the other hand, circumvents such drawback by performing the modeling, optimization, and synthesis only by using the data when the physical model is unavailable. From an asymptotic analysis point of view, such machinery are studied under adaptive control and system identification [17,105]. Later, more research was conducted in a discrete non-asymptotic fashion where control and analysis are performed on batches of collected data [6, 50, 164] or through an online iterative procedure [126, 156]. Besides, in regards to the adaptive nature of such algorithms, there is a close connection between online data-driven control and reinforcement learning [27,100,154]. The latter works extend policy iteration [83] to approximate LQR by avoiding the direct solution of ARE; yet most of them fail to scale towards high dimensions.

Control and estimation of large-scale systems in essence, is a more complicated problem than single-agent (centralized) control because of higher levels of uncertainty, dimensions, and modeling errors. Nevertheless, model-free large-scale analysis—as one tool to address such setbacks—is still an immature research area. Similar ideas have been investigated in Multiagent Reinforcement Learning (MARL) community, mainly presented on a more general Markov Decision Processes (MDP) framework [33, 177], but still application of most of the MARL-based algorithms remains challenging as the state-action space grows exponentially large with the number of agents [114]. From a control theoretic perspective, [109] addresses this using ideas from mean-field multiagent systems and with the key assumption of partial exchangeability. Also, an SDP projection-based analysis is studied in [36] where each agent is shown to obtain sublinear regret compared to the best fixed controller in hindsight. The problem is also considered from a game-theoretic standpoint [102,125] where the set of agents possess conflicting objectives.

In the sequel, we first introduce two distributed data-driven control frameworks. The

[1] For instance, even the LQR solution with its strong input robustness properties, may have small stability margins for general parameter perturbations [57].

first scheme proposes a consensus-based setup where each agent's objective is to find its own optimal LQR controller, no matter the overall cost of the distributed scheme. In the second part, we take the structure of the distributed controller into consideration and find a suboptimal policy.

### *3.1 Mathematical Preliminaries*

Consider a homogeneous network of identical agents with interdependecies briefed in a common network-level objective function. In particular, we assume that the system contains $N$ agents forming a graph $\mathcal{G}$, where each node of the graph represents a linear discrete-time decoupled dynamical system as,

$$\boldsymbol{x}_{i,t+1} = A\boldsymbol{x}_{i,t} + B\boldsymbol{u}_{i,t}, \qquad i = 1, 2, \dots, N \tag{3.1}$$

where $\boldsymbol{x}_{i,t} \in \mathbb{R}^n$ and $\boldsymbol{u}_{i,t} \in \mathbb{R}^m$ denote the state and input of agent $i$ at time-step $t$, respectively, and $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are unknown system parameters that form a controllable pair. The formulation can be compactly presented as,

$$\hat{\mathbf{x}}_{t+1} = \hat{\mathbf{A}}\hat{\mathbf{x}}_t + \hat{\mathbf{B}}\hat{\mathbf{u}}_t, \tag{3.2}$$

where $\hat{\mathbf{x}}_t \in \mathbb{R}^{Nn}$ and $\hat{\mathbf{u}}_t \in \mathbb{R}^{Nm}$ contain the entirety of the states and inputs,

$$\hat{\mathbf{x}}_t = \big[\boldsymbol{x}_{1,t}; \dots; \boldsymbol{x}_{N,t}\big], \qquad \hat{\mathbf{u}}_t = \big[\boldsymbol{u}_{1,t}; \dots; \boldsymbol{u}_{N,t}\big],$$

with $\hat{\mathbf{A}} \in \mathbb{R}^{Nn\times Nn}$ and $\hat{\mathbf{B}} \in \mathbb{R}^{Nn\times Nm}$ defined in block diagonal forms $\hat{\mathbf{A}} = \mathrm{I}_N \otimes A$ and $\hat{\mathbf{B}} = \mathrm{I}_N \otimes B$. The agents are assumed to collectively minimize the cost,

$$\min_{(\boldsymbol{u}_{1,t})_{t=0}^{\infty},\dots,(\boldsymbol{u}_{N,t})_{t=0}^{\infty}\in\ell_2} \hat{\mathbf{J}}(\hat{\mathbf{x}}_0, \hat{\mathbf{u}}_t) = \sum_{t=0}^{\infty}\Bigg(\sum_{i=1}^{N}\Big[\boldsymbol{x}_{i,t}^\top Q_1 \boldsymbol{x}_{i,t} + \boldsymbol{u}_{i,t}^\top R \boldsymbol{u}_{i,t}\Big] + \sum_{i=1}^{N}\sum_{j\neq i}^{N}\Big[(\Delta_{ij}\boldsymbol{x}_t)^\top Q_2 (\Delta_{ij}\boldsymbol{x}_t)\Big]\Bigg), \tag{3.3}$$

subject to the dynamics in equation (3.2).[2] Intuitively, the first term in the sum indicates the absolute cost pertinent to each agent, while the second term denotes the cost that arises

[2]One instance of such interactive cost among agents appears in the cooperative game setup where agent $i$ solves the minimization problem,

$$\min_{(\boldsymbol{u}_{i,t})_{t=0}^{\infty}\in\ell_2} \mathbf{J}_i(\hat{\mathbf{x}}_0, \hat{\mathbf{u}}_t) = N\sum_{t=0}^{\infty}\Bigg(\boldsymbol{x}_{i,t}^\top Q_1 \boldsymbol{x}_{i,t} + \boldsymbol{u}_{i,t}^\top R \boldsymbol{u}_{i,t} + \sum_{j\neq i}^{N}(\Delta_{ij}\boldsymbol{x}_t)^\top Q_2 (\Delta_{ij}\boldsymbol{x}_t)\Bigg),$$

where $\Delta_{ij}\boldsymbol{x}_t = \boldsymbol{x}_{i,t} - \boldsymbol{x}_{j,t}$ and $Q_1 \succeq 0$, $Q_2 \succeq 0$, $R \succ 0$ are the given parameters. Then, it is well-known that the set of Pareto front solution of this game can be obtained by minimizing the parametric cost

from agents' couplings. The cost in (3.3) can also be written in compact form,

$$\hat{\mathbf{J}}(\hat{\mathbf{x}}_0, \hat{\mathbf{u}}_t) = \sum_{t=0}^{\infty} \hat{\mathbf{x}}_t^\top \hat{\mathbf{Q}} \hat{\mathbf{x}}_t + \hat{\mathbf{u}}_t^\top \hat{\mathbf{R}} \hat{\mathbf{u}}_t \ , \tag{3.4}$$

with $\hat{\mathbf{R}} = \mathrm{I}_N \otimes R$ and $\hat{\mathbf{Q}} \in \mathcal{U}_{n,n}^N(\mathcal{G})$, where $\mathcal{U}_{a,b}^N(\mathcal{G})$ denotes a linear subspace defined based on the graph structure,[3]

$$\mathcal{U}_{a,b}^N(\mathcal{G}) = \Big\{ M \in \mathbb{R}^{aN \times bN} |\ [M]_{ij} = \mathbf{0} \text{ if } j \notin \mathcal{N}_i,\ [M]_{ij} \in \mathbb{R}^{a \times b},\ i, j = 1, \cdots, N \Big\}.$$

Without having access to the system parameters, we are interested in designing feedback gains with a desired sparsity pattern using measurements from the system in (3.2). Of particular interest in this analysis are the structured feedback gains reflecting the underlying interaction network. The canonical distributed (structured) optimal control problem is often posed in the form of,

$$\begin{aligned} \min_{\hat{\mathbf{K}}} \quad & \hat{\mathbf{J}}(\hat{\mathbf{x}}_0, \hat{\mathbf{u}}_t) \\ \text{s.t.} \quad & \hat{\mathbf{x}}_{t+1} = \hat{\mathbf{A}} \hat{\mathbf{x}}_t + \hat{\mathbf{B}} \hat{\mathbf{u}}_t, \\ & \hat{\mathbf{u}}_t = \hat{\mathbf{K}} \hat{\mathbf{x}}_t, \qquad \hat{\mathbf{K}} \in \mathcal{U}_{m,n}^N(\mathcal{G}). \end{aligned} \tag{3.5}$$

In general, the constrained optimization problem in (3.5) is an NP-hard problem [78] and has been investigated via assumptions such as quadratic invariance of the controller [137, 169], spatially invariant dynamics [19], and local parameter tuning [24] or tackled directly with the aid of projected gradient-based policy updates [29]—all based on the complete knowledge of the system. Indeed, not knowing the true underlying system parameters, adds one additional layer of difficulty to this framework. Then the control designer is left with no choice rather than leveraging the input-output observations to find a "reasonable" distributed policy for the system.

---

function,

$$\min_{(\boldsymbol{u}_{1,t})_{t=0}^{\infty}, \ldots, (\boldsymbol{u}_{N,t})_{t=0}^{\infty} \in \ell_2} \ \sum_{i=1}^{N} \alpha_i \mathbf{J}_i(\hat{\mathbf{x}}_0, \hat{\mathbf{u}}_t),$$

parameterized by $\alpha_1, \ldots, \alpha_N$ where $\alpha_i \in [0, 1]$ and $\sum_i \alpha_i = 1$ (see *e.g.* [59]). Therefore, a cost such as in (3.3) would be a specific case of the fair Pareto optimal solution with the choice of $\alpha_i = 1/N$ for all $i$.

[3] Such interdependent structure of the cost naturally arises in a graph-based distributed control framework (See for instance [24, 53, 112, 168]).

Inspired by a $\mathcal{Q}$-learning-based policy iteration algorithm, we propose a model-free distributed policy iteration scheme to obtain a suboptimal solution to the problem in (3.5). Inherent to our algorithm is a subproblem synthesis whose dimension is related to the maximum degree of the underlying graph. We will show our design is suboptimal in the sense that the convergence depends on finding the optimal solution to this subproblem along the way. Our method is also capable of finding stability margins from data without the need to solve a high-dimensional optimization and LMIs [141].

### *3.1.1 Model-Free Policy Iteration for LQR*

We first cover some basics of policy iteration and its connections to LQR. Policy iteration for discrete-time infinite-horizon LQR was initially introduced by Hewer [83] to approximate the solution to ARE in a two-step iterative algorithm. For a system with parameters $(A, B, Q, R)$ with symmetric $Q \succeq 0$ and $R \succ 0$, the algorithm is initialized from some stabilizing initial policy $K_0$ and includes *policy evaluation*,

$$(A - BK_j)^\top P_{j+1}(A - BK_j) - P_{j+1} + Q + K_j^\top R K_j = 0, \tag{3.6}$$

followed by *policy update*,

$$K_{j+1} = \left(R + B^\top P_{j+1} B\right)^{-1} B^\top P_{j+1} A. \tag{3.7}$$

Hewer showed that this iteration converges to the optimal solution $(K^*, P^*)$ under the stabilizability and detectability assumptions. However, this method assumes perfect knowledge of the system parameters $A$ and $B$. In order to obtain an approximate to $K^*$ without direct knowledge of $A$ and $B$, Bradtke et al. [27] proposed a mechanism inspired by the $\mathcal{Q}$-learning algorithm [170] to perform a data-driven version of policy iteration. $\mathcal{Q}$-learning has a trial-and-error nature where a control agent optimizes some value function from observing the results of its own actions. Characterized by Bellman [21], this value function can be formulated by the $\mathcal{Q}$-function of the state-action pair $(\boldsymbol{x}_t, \boldsymbol{u}_t)$ at time-step $t$ as,

$$\mathcal{Q}(\boldsymbol{x}_t, \boldsymbol{u}_t) = \mathcal{R}(\boldsymbol{x}_t, \boldsymbol{u}_t) + \gamma \mathcal{Q}(\boldsymbol{x}_{t+1}, \boldsymbol{u}_{t+1}), \tag{3.8}$$

where $\mathcal{Q}(\boldsymbol{x}_t, \boldsymbol{u}_t) = \boldsymbol{x}_t^\top P \boldsymbol{x}_t$ is the *state-action* $\mathcal{Q}$-function, $P$ is the solution to the Lyapunov equation (3.6), $\mathcal{R}(\boldsymbol{x}_t, \boldsymbol{u}_t) = \boldsymbol{x}_t^\top Q \boldsymbol{x}_t + \boldsymbol{u}_t^\top R \boldsymbol{u}_t$ is the one-step *reward function*, and $\gamma \in [0, 1]$

denotes the discount factor. The control actions are derived from a set of optimal policies that assume the form of a linear feedback law $\boldsymbol{u}_t = -\hat{K}\boldsymbol{x}_t$ in the LQR framework where $\hat{K}$ denotes the estimate of the controller. Equation (3.8) can also be reformulated as,

$$\mathcal{Q}(\boldsymbol{x}_t, \boldsymbol{u}_t) = \boldsymbol{z}_t^\top H \boldsymbol{z}_t, \tag{3.9}$$

where $\boldsymbol{z}_t = [\boldsymbol{x}_t;\ \boldsymbol{u}_t]$ and $H$ is a block matrix defined as,

$$H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} = \begin{bmatrix} Q + \gamma A^\top P A & \gamma A^\top P B \\ \gamma B^\top P A & R + \gamma B^\top P B \end{bmatrix}. \tag{3.10}$$

Having estimated the parameters in $H$ through observations $\boldsymbol{z}_t$ by utilizing least-squares, the controller is then updated as,

$$\hat{K} = -\hat{H}_{22}^{-1}\hat{H}_{21} = -\gamma\big(R + \gamma B^\top P B\big)^{-1} B^\top P A, \tag{3.11}$$

which is equivalent to the solution of $\partial\mathcal{Q}/\partial\boldsymbol{u}_t = 0$ in a model-based scenario. The adaptive nature of the algorithm is originated from a linear Recursive Least-Squares (RLS) step to learn the parameters of $H$ in real-time. Hence, a linear parameterization of (3.9) is formed as,

$$\mathcal{Q}(\boldsymbol{x}_t, \boldsymbol{u}_t) = \boldsymbol{z}_t^\top H \boldsymbol{z}_t = \mathbf{z}_t^\top \boldsymbol{\theta}_H,$$

where $\mathbf{z}_t, \boldsymbol{\theta}_H \in \mathbb{R}^{p(p+1)/2}$ with $p = n + m$, $\mathbf{z}_t$ forms a quadratic basis of the elements in $\boldsymbol{z}_t$ as,

$$\mathbf{z}_t = \begin{bmatrix} \boldsymbol{z}_1^2 & \boldsymbol{z}_1\boldsymbol{z}_2 & \cdots & \boldsymbol{z}_1\boldsymbol{z}_p & \boldsymbol{z}_2\boldsymbol{z}_1 & \cdots & \boldsymbol{z}_p^2 \end{bmatrix}^\top, \tag{3.12}$$

and $\boldsymbol{\theta}_H$ vectorizes the upper right triangle of the symmetric matrix $H$. We also show this half-vectorization by $\boldsymbol{\theta}_H = \text{vech}(H)$. Therefore from (3.8),

$$\mathcal{R}(\boldsymbol{x}_t, \boldsymbol{u}_t) = \mathcal{Q}(\boldsymbol{x}_t, \boldsymbol{u}_t) - \gamma\mathcal{Q}(\boldsymbol{x}_{t+1}, \boldsymbol{u}_{t+1}) = \boldsymbol{z}_t^\top H \boldsymbol{z}_t - \gamma \boldsymbol{z}_{t+1}^\top H \boldsymbol{z}_{t+1} = \boldsymbol{\phi}_t^\top \boldsymbol{\theta}_H, \tag{3.13}$$

where $\boldsymbol{\phi}_t = \mathbf{z}_t - \mathbf{z}_{t+1}$. Henceforth, RLS can be iteratively employed to find the estimate of $\boldsymbol{\theta}_H$ as,

$$\begin{aligned} \hat{\boldsymbol{\theta}}_j &= \hat{\boldsymbol{\theta}}_{j-1} + \frac{\mathcal{P}_{j-1}\boldsymbol{\phi}_t(r_t - \boldsymbol{\phi}_t^\top \hat{\boldsymbol{\theta}}_{j-1})}{1 + \boldsymbol{\phi}_t^\top \mathcal{P}_{j-1}\boldsymbol{\phi}_t} \\ \mathcal{P}_j &= \mathcal{P}_{j-1} - \frac{\mathcal{P}_{j-1}\boldsymbol{\phi}_t\boldsymbol{\phi}_t^\top \mathcal{P}_{j-1}}{1 + \boldsymbol{\phi}_t^\top \mathcal{P}_{j-1}\boldsymbol{\phi}_t}, \end{aligned} \tag{3.14}$$

where $r_t = \mathcal{R}(\boldsymbol{x}_t, \boldsymbol{u}_t)$ and $\mathcal{P}$ is the projection adjustment factor that is reset at the beginning of each iteration in RLS with $\mathcal{P}_0 = \beta \mathrm{I}$ for some large enough $\beta > 0$. This recursive algorithm converges asymptotically if the data vector $\boldsymbol{\phi}_t$ is Persistently Excited (PE) (see [17, 76]), *i.e.*,

$$\alpha' \mathrm{I} \leq \frac{1}{M} \sum_{i=1}^{M} \phi_{t-i} \phi_{t-i}^{\top} \leq \beta' \mathrm{I} \qquad \forall\, t, M \geq M_0, \tag{3.15}$$

for some positive parameters $M_0$, $\alpha'$, and $\beta'$. This machinery also enables getting an updated estimate of $\hat{H}$ and subsequently use (3.11) to update the controller. Additionally, $P$ can also be approximated by,

$$\hat{P} = \hat{H}_{11} - \hat{H}_{12}\hat{K} - \hat{K}^{\top}\hat{H}_{21} + \hat{K}^{\top}\hat{H}_{22}\hat{K}.$$

The data-driven nature of the machinery planned out above helps circumvent difficulty of modeling under (possibly unknown) uncertainties. However, such machinery has been focused for a centralized (single agent) system and not directly applicable to large-scale systems. In the next section, we will introduce one strategy to make this shift.

### *3.2 Distributed $Q$-Learning*

In this section, we extend the $Q$-learning setup based on the distributed control framework defined earlier. To this end, we assume that each agent enjoys its own $Q$-function whose reward is a function of the state of the agent as well as the state of its neighbors. For agent $i$ we define,

$$\begin{aligned} Q_i(x_{i,t}, u_{i,t}) &= R_i(x_{i,t}, u_{i,t}) + \gamma Q_i(x_{i,t+1}, u_{i,t+1}) \\ &= y_{i,t}^{\top} \mathcal{Q}_1 y_{i,t} + \gamma x_{i,t+1}^{\top} P_i x_{i,t+1}, \end{aligned} \tag{3.16}$$

here $y_{i,t} = \begin{bmatrix} x_{i,t};\ u_{i,t};\ x_{j_1,t};\ \ldots\ ;\ x_{j_{d_i},t} \end{bmatrix}$, $d_i$ is the degree of agent $i$, $j_{d_k} \in \mathcal{N}_i$ for $k = 1, \ldots, i$, and $\mathcal{Q}_i$ is defined as,

$$\mathcal{Q}_i = \begin{bmatrix} (d_i+1)Q_1 & \mathbf{0} & -Q_1 & \ldots & -Q_1 \\ \mathbf{0} & R & \mathbf{0} & \ldots & \mathbf{0} \\ -Q_1 & \mathbf{0} & Q_1 & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -Q_1 & \mathbf{0} & \mathbf{0} & \ldots & Q_1 \end{bmatrix} \in \mathbb{R}^{(d_i+2)n \times (d_i+2)n}. \tag{3.17}$$

The structure of $\mathcal{Q}_i$ is resulting from equation (2.50) and implies the new definition of reward function for multiple agents in the system. Note that equations (3.16) and (3.17) make two implicit assumptions: (1) there is no control coupling amongst agents and, (2) each agent has only access to the reward form the coupling between its own state and the states of neighbors. This motivates the existence of zero blocks in (3.17). Equation (3.16) can also be rearranged into,

$$Q_i(x_i, u_i) = y_{i,t}^\top H_i y_{i,t},$$

where,

$$H_i = \begin{bmatrix} (d_i+1)Q_1 + \gamma A^\top P_i A & \gamma A^\top P_i B & -Q_1 & \dots & -Q_1 \\ \gamma B^\top P_i A & R + \gamma B^\top P_i B & \mathbf{0} & \dots & \mathbf{0} \\ -Q_1 & \mathbf{0} & Q_1 & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -Q_1 & \mathbf{0} & \mathbf{0} & \dots & Q_1 \end{bmatrix}$$

Since there is no control coupling, in order to update the controller for each agent at each iteration we set again,

$$K_{i,\text{new}} = -H_{i,22}^{-1} H_{i,21}. \tag{3.18}$$

Finally, as in the centralized case, for each agent $i$ we define $\phi_{i,t} = \bar{z}_{i,t} - \gamma \bar{z}_{i,t+1}$.

### *3.2.1 Main Results*

We now introduce the distributed policy iteration algorithm. The analysis in this part is mainly inspired by [28], however, there are fundamental differences as we only assume couplings through a global cost function; as such, the state transition or feedback of each agent only depends on their own history of states and actions. Under these assumptions, we show that this way of coupling in the case of identical systems signifies the interdependency of the agents in the decision-making process. We briefly explain the steps of the algorithm: $\hat{\theta}_{i,k}$ is the estimate of $H_i$. In the sequel, $\theta^*_{i,k}$ denotes the parameters of $H_i$ obtained using the true system parameters. $K_{i,k}$ denotes the controller estimate. The counter $t$ keeps track of the number of collected data while $k$ designates the iteration count on the parameters estimate. Note that these counters are never reset to zero. $P_k(j)$ is the covariance matrix

**Algorithm 3** The distributed $Q$-learning Algorithm

**Initialize**:
  Random: $\hat{\theta}_{1,0}(0), \ldots, \hat{\theta}_{N,0}(0)$
  Stabilizable: $K_{1,0}, \ldots, K_{N,0}$
  $t = 0$, $k = 1$
**while convergence:**
  Reset Covariance: $P_k(0) = P_0$
  **For** $j = 1$ to $M$:
    **For system** $i = 1, \ldots, N$:
      Choose $e_t$ and find $u_{i,t} = K_{i,k}x_{i,t} + e_t$
      Collect $x_{i,t+1}$ by applying $u_{i,t}$ to the system
      Update $\hat{\theta}_{i,k}(j)$ using RLS
      $t = t + 1$
  **For system** $i = 1, \ldots, N$:
    Find symmetric $\hat{H}_{i,k}$ corresponding to $\hat{\theta}_{i,k}$
    Policy update: $K_{i,k+1} = -\hat{H}^{-1}_{i,k(22)}\hat{H}_{i,k(21)}$
    Initialize parameters $\hat{\theta}_{i,k+1}(0) = \hat{\theta}_{i,k}(M)$
  $k = k + 1$

reset to some constant $P_0$ at each iteration to revitalize the gain. Each RLS estimation interval includes $M$ time-steps. The value of $M$ is dependent on the number of unknown parameters in $\hat{\theta}_{i,k}$ and also the desired accuracy. The control signal is PE at each iteration of the RLS and $e_t$ is the excitation component which is assumed to be the same for all agents. After convergence of RLS, the controller for each agent is updated based on (3.18). The estimation parameters are reinitialized from the final value of the previous iteration such that $\hat{\theta}_{k+1}(0) = \hat{\theta}_k(M)$. The reader is referred to Chapter 3 of [76] for exact steps of RLS.

**Theorem 17.** *Assume that for all $i = 1, \ldots, N$, the pair $(A, B)$ is a controllable and $K_{i,0}$ is stabilizing with a PE signal $\phi_{i,t}$. Then there exists $M < \infty$ such that Algorithm 3 generates*

*a sequence* $\{K_{i,k}\}$ *with* $\lim_{k\to\infty} \|K_{i,k} - K^*\| = 0$*, where* $K^* = LQR(A, B, Q_1, R)$.

*Proof.* From the definition of the reward for agent $i$,

$$r_{i,t} = y_{i,t}^\top H_i y_{i,t} - \gamma y_{i,t+1}^\top H_i y_{i,t+1}.$$

Furthermore,

$$r_{i,t} = x_{i,t}^\top Q_1 x_{i,t} + u_{i,t}^\top R u_{i,t} + \sum_{k=1}^{d_i} \big(x_{i,t} - x_{j_k,t}\big)^\top Q_1 \big(x_{i,t} - x_{j_k,t}\big),$$

and,

$$y_{i,t}^\top H_i y_{i,t} = \begin{bmatrix} x_{i,t}^\top & u_{i,t}^\top \end{bmatrix} \begin{bmatrix} Q_1 + \gamma A^\top P_i A & \gamma A^\top P_i B \\ \gamma B^\top P_i A & R + \gamma B^\top P_i B \end{bmatrix} \begin{bmatrix} x_{i,t} \\ u_{i,t} \end{bmatrix}.$$

Consequently, we obtain,

$$\xi_{i,t} = \bar{z}_{i,t}^\top \hat{\theta}_{i,k}, \tag{3.19}$$

where,

$$\xi_{i,t} = x_{i,t}^\top Q_1 x_{i,t} + u_{i,t}^\top R u_{i,t} + \sum_{k=1}^{d_i} \big(x_{i,t+1} - x_{j,t+1}\big)^\top Q_1 \big(x_{i,t+1} - x_{j,t+1}\big). \tag{3.20}$$

Hence the distributed nature of the problem narrows down to a particular distributed form of RLS. We stack the $N$ equations of the form (3.19) for all agents into vector form as,

$$\underbrace{\begin{bmatrix} \xi_{1,t} \\ \vdots \\ \xi_{N,t} \end{bmatrix}}_{\Xi_t} = \underbrace{\begin{bmatrix} \bar{z}_{1,t}^\top & & \\ & \ddots & \\ & & \bar{z}_{N,t}^\top \end{bmatrix}}_{Z_t} \underbrace{\begin{bmatrix} \hat{\theta}_{1,k} \\ \vdots \\ \hat{\theta}_{N,k} \end{bmatrix}}_{\hat{\Theta}_k}. \tag{3.21}$$

Based on the definition of PE in (3.15), it is straightforward to show that the matrix $Z_t$ is PE if $\bar{z}_{i,t}$ is PE for all $i$. This results in the convergence of equation (3.21) to some $\Theta^*$ for large enough $M$.[4] From Theorem 5.1 in [28],

$$\lim_{k\to\infty} \left\|\hat{\theta}_{i,k} - \theta_{i,k}^*\right\| = 0, \quad \lim_{k\to\infty} \left\|\theta_{i,k}^* - \theta_{i,k-1}^*\right\| = 0. \tag{3.22}$$

[4]Parameter estimation for the multi-output system is an straightforward extension of the scalar case and is discussed in Chapter 3.8 of [76].

However, the convergence of $\hat{\theta}_{i,k}$ for all $i$ to one single value is non-trivial due to the inter-dependency in RLS. We will show that for a connected network of agents,

$$\lim_{t\to\infty} \|x_{i,t} - x_{j,t}\| = 0,$$

for any $i$ and $j$. Note that according to (3.20), if a node is disconnected from the graph it can be individually examined as in the centralized case. Recall that for $\ell = i, j$,

$$x_{\ell,t+1} = Ax_{\ell,t} + Bu_{\ell,t} = (A - BK_{\ell,k})x_{\ell,t} + Be_t.$$

As such,

$$x_{i,t+1} - x_{j,t+1} = (A - BK_{i,k})(x_{i,t} - x_{j,t}) + B\Delta K_k x_{j,t} \tag{3.23}$$

where $\Delta K_k = K_{i,k} - K_{j,k}$. Then if we show that $\|\Delta K_k\| \to 0$ as $k \to \infty$ we obtain,

$$\|x_{i,t+1} - x_{j,t+1}\| = \left\|(A - BK_{i,k})^{t+1}(x_0^{(i)} - x_0^{(j)})\right\| \to 0, \tag{3.24}$$

given that the policy iteration algorithm leads to a more stabilizing controller $K_{i,k}$ as $k$ increases [28]. Then,

$$\begin{aligned}\|K_{i,k} - K_{j,k}\| &= \left\|\hat{H}^{-1}_{j,k-1,22}\hat{H}_{j,k-1,21} - \hat{H}^{-1}_{i,k-1,22}\hat{H}_{i,k-1,21}\right\| \\ &= \Big\|\hat{H}^{-1}_{j,k-1,22}\Big((\hat{H}_{j,k-1,21} - \hat{H}_{i,k-1,21}) \\ &\qquad + (\hat{H}_{i,k-1,22} - \hat{H}_{j,k-1,22})\hat{H}^{-1}_{j,k-1,22}\hat{H}^{-1}_{i,k-1,22}\hat{H}_{i,k-1,21}\Big)\Big\|.\end{aligned} \tag{3.25}$$

Since $\hat{H}_{22}$ and $\hat{H}_{21}$ contain only a subset of elements in $\hat{\theta}$,

$$\begin{aligned}\left\|\hat{H}_{j,k-1,21} - \hat{H}_{i,k-1,21}\right\| &\le \left\|\hat{\theta}_{j,k-1} - \hat{\theta}_{i,k-1}\right\|, \\ \left\|\hat{H}_{j,k-1,22} - \hat{H}_{i,k-1,22}\right\| &\le \left\|\hat{\theta}_{j,k-1} - \hat{\theta}_{i,k-1}\right\|.\end{aligned} \tag{3.26}$$

Hence equations (3.25) and (3.26) lead to,

$$\begin{aligned}\|K_{i,k} - K_{j,k}\| &\le \left\|\hat{H}^{-1}_{j,k-1,22}\right\| \left\|\hat{\theta}_{j,k-1} - \hat{\theta}_{i,k-1}\right\| \left\|1 + \hat{H}^{-1}_{j,k-1,22}\hat{H}^{-1}_{i,k-1,22}\hat{H}_{i,k-1,21}\right\| \\ &\le \kappa_0 \left\|\hat{\theta}_{j,k-1} - \hat{\theta}_{i,k-1}\right\|,\end{aligned} \tag{3.27}$$

where we have used the fact that the estimated parameters are bounded and $\kappa_0 > 0$ is a constant such that,

$$\left\|\hat{H}^{-1}_{j,k-1,22}\right\| \cdot \left\|1 + \hat{H}^{-1}_{j,k-1,22}\hat{H}^{-1}_{i,k-1,22}\hat{H}_{i,k-1,21}\right\| \le \kappa_0.$$

From Lemma 5.2 in [28],

$$\left\|\theta^*_{\ell,k} - \hat{\theta}_{\ell,k}\right\| \leq \epsilon_M \left( \left\|\theta^*_{\ell,k} - \theta^*_{\ell,k-1}\right\| + \left\|\theta^*_{\ell,k-1} - \hat{\theta}_{\ell,k-1}\right\| \right),$$

which for large enough $M$ results in,

$$\begin{aligned}\left\|\theta^*_{i,k} - \hat{\theta}_{i,k}\right\| + \left\|\theta^*_{j,k} - \hat{\theta}_{j,k}\right\| \leq \epsilon_M \Big( & \left\|\theta^*_{i,k} - \theta^*_{i,k-1}\right\| + \left\|\theta^*_{i,k-1} - \hat{\theta}_{i,k-1}\right\| \\ & + \left\|\theta^*_{j,k} - \theta^*_{j,k-1}\right\| + \left\|\theta^*_{j,k-1} - \hat{\theta}_{j,k-1}\right\| \Big).\end{aligned} \tag{3.28}$$

Using triangle inequality on the left side of this inequality,

$$\begin{aligned}\left| \left\|\hat{\theta}_{i,k} - \hat{\theta}_{j,k}\right\| - \left\|\theta^*_{i,k} - \theta^*_{j,k}\right\| \right| &\leq \left\| \left(\hat{\theta}_{i,k} - \hat{\theta}_{j,k}\right) - \left(\theta^*_{i,k} - \theta^*_{j,k}\right) \right\| \\ &\leq \left\|\theta^*_{i,k} - \hat{\theta}_{i,k}\right\| + \left\|\theta^*_{j,k} - \hat{\theta}_{j,k}\right\|.\end{aligned} \tag{3.29}$$

Then, from (3.28) and (3.29) and for large $k$,

$$\begin{aligned}\left\|\hat{\theta}_{i,k} - \hat{\theta}_{j,k}\right\| \leq \epsilon_M \Big( & \left\|\theta^*_{i,k} - \theta^*_{i,k-1}\right\| + \left\|\theta^*_{i,k-1} - \hat{\theta}_{i,k-1}\right\| + \left\|\theta^*_{j,k} - \theta^*_{j,k-1}\right\| + \left\|\theta^*_{j,k-1} - \hat{\theta}_{j,k-1}\right\| \Big) \\ & + \left\|\theta^*_{i,k} - \theta^*_{j,k}\right\|.\end{aligned}$$

Hence using the result in (3.22),

$$\left\|\hat{\theta}_{i,k} - \hat{\theta}_{j,k}\right\| \to 0,$$

and plugging this into (3.27),

$$\|\Delta K_k\| = \|K_{i,k} - K_{j,k}\| \to 0, \tag{3.30}$$

Hence,

$$\|x_{i,t+1} - x_{j,t+1}\| \to 0.$$

This implies that based on (3.20), for identical systems the algorithm moves towards $N$ decoupled $Q$-learning algorithms for each agent. Thus, although the provided data is from an interconnected system, each controller converges to its optimal value, i.e.,

$$\lim_{k\to\infty} \|K_{i,k} - K^*\| = 0, \qquad \text{for} \quad i = 1, 2, \ldots, N.$$

□

**Remark 6.** *In Algorithm 3, we have assumed that the exploration signal, $e_t$, is equal for every agent at each time step. This is a valid assumption as long as $Z_t$ in (3.21) is PE so that RLS is assured to converge. Another option would be to choose the excitation signals $e_{i,t}$ and $e_{j,t}$ in a way that,*

$$e_{i,t} - e_{j,t} = -\Delta K_k x_{j,t}.$$

*Hence, not only the input to the RLS is PE, the difference cancels out $\Delta K_k x_{j,t}$ in (3.23). However, this setup is more challenging to implement, particularly for large-scale systems.*

### *3.2.2 Computational Saving*

The computational saving resulting from using the distributed $Q$-learning algorithm is significant, since for a large system, the design of the LQR controller with the computational complexity of solving ARE of order $\mathcal{O}(n^3)$, can be prohibitively expensive. The main computational burden of Algorithm 3 comes from RLS where the complexity of its implementation is $\mathcal{O}(\gamma^2)$ with $\gamma$ parameters to learn. Assuming that the system contains $N$ agents each having $n$ states and $m$ inputs, the computational complexity of the centralized $Q$-learning is obtained by,

$$\mathcal{O}\left(\left(\frac{(Nn+Nm)(Nn+Nm+1)}{2}\right)^2\right) \approx \mathcal{O}\left(N^4(n+m)^4\right),$$

while for the distributed case the code performs $N$ repetitions of the same RLS leading to the complexity bound,

$$\mathcal{O}\left(N\left(\frac{(n+m)(n+m+1)}{2}\right)^2\right) \approx \mathcal{O}\left(N(n+m)^4\right).$$

Hence the complexity reduction is,

$$\frac{N^4(n+m)^4 - N(n+m)^4}{N^4(n+m)^4} \times 100 = \frac{N^3-1}{N^3} \times 100\%,$$

which is substantial for large $N$. Table 3.1 compares the computational saving for some values of $N$.

| N | 2 | 3 | 5 | 8 | 100 |
|---|---|---|---|---|---|
| Saving (%) | 87.5 | 96.29 | 99.2 | 99.8 | 99.99 |

Table 3.1: Approximate computational saving of the distributed $Q$-learning compared to the centralized case in [28].

### *3.2.3 Example*

In this section, we provide an example to show the efficiency of the distributed $Q$-learning algorithm for a set of identical communicating Unmanned Aerial Vehicle (UAV)s. We consider the autonomous flight of a network of six interconnected UAVs which are set to perform a common task such as geographical data collection or putting out a wildfire. To cover the whole targeted area, these UAVs are programmed to move in parallel and in order for minimal signal transmissions, each UAV only communicates with its closest neighbor in the network as depicted in Figure 3.1. The discrete-time dynamics of UAVs is considered by mini-aircraft linear parameters that can be found in [87]. the dynamics of each UAV is defined by,

$$A = \begin{bmatrix} 0.7000 & 0.0014 & 0.1132 & 0.0005 & -0.0967 \\ 0 & 0.6945 & -0.0171 & -0.0005 & 0.0068 \\ 0 & 0.0003 & 0.7000 & 0.0957 & -0.0048 \\ 0 & 0.0060 & -0.0000 & 0.6131 & -0.0936 \\ 0 & -0.0277 & 0.0002 & 0.0973 & 0.6287 \end{bmatrix}, \quad B = \begin{bmatrix} -0.0076 & 0.0000 & 0.0003 \\ -0.0115 & 0.0997 & 0.0000 \\ 0.0212 & 0.0000 & -0.0081 \\ 0.4152 & 0.0003 & -0.1589 \\ 0.1742 & -0.0014 & -0.0154 \end{bmatrix}$$

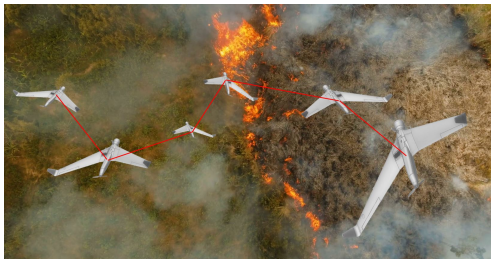
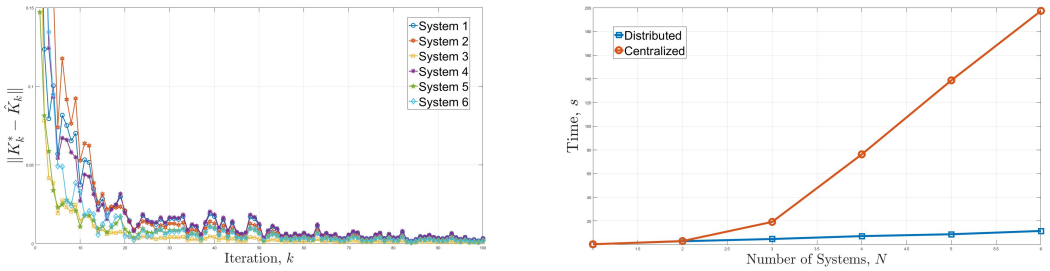


Figure 3.1: A group of identical firefighting UAVs maneuvering in parallel aiming to extinguish a blaze (Aerial view of the forest fire - *Photo Credit: Alex Punker, Bigstock*).




(a) (b)

Figure 3.2: Performance of the distributed Q-learning algorithm. (a) The error between the LQR optimal controller of each subsystem and the estimate of the algorithm at each iteration $k$. (b) Time complexity of distributed and centralized algorithms for different number of agents.

for a discrete-time linear time-invariant system. We assume $Q_1 = \mathrm{I}_5$ and $R = \mathrm{I}_3$. We will show the results of the distributed policy iteration for $N = 6$, $n = 5$, and $m = 3$ and compare the computational performance with the centralized case. For the distributed algorithm we consider $M = 50$ and the exploration signal $e_t$ is generated from a normal distribution. Figure 3.2a shows the results of simulations regarding the controller error norm. A comparison between the computational performance of the centralized and distributed methods is also provided in Figure 3.2b. For scaling purposes, $M$ and $e_t$ are re-adjusted for each $N$.[5]

### *3.3 Distributed Control: Structured $K$ Approach*

In this part, unlike the previous part, we propose the LQR-based Data-Driven Distributed Policy Iteration (D3PI) algorithm to iteratively learn a set of stabilizing controllers for unknown but identical linear dynamical systems that are connected with a network topology induced by couplings in their performance indices that can also be cast as a particular form of a cooperative game-theoretic setup. Furthermore, data-driven control of (unknown)

[5]The main reason for this is that $M$ needs to be modified since $N$ is proportionally related to the centralized system dimensions $Nn$ and $Nm$.

identical systems is motivated by applications such as formation flight [151] or monitoring networked cameras [25] where the agents possess corresponding dynamics whose exact models are inaccessible due to model perturbations and uncertainties. Notably, to avoid annihilation of the consensus-based nature of the distributed control setup, the option of single-agent learning is ruled out in this framework. We assume that the feedback signal for stabilization is available to each system locally through the same network topology. Then, we find a compound data-driven feedback mechanism for the entire networked system which is trained based on data collected from only a small portion of the network. In particular, considering a subgraph including the agent corresponding to the node with maximum degree, we require temporary feedback links within this subgraph in order to iteratively learn a stabilizing structured controller for the entire network that is optimal for the subgraph—thus, generally suboptimal for the entire network (Figure 3.3). Finally, the compound feedback for the entire network is constructed based on this locally optimal policy. We provide extensive analysis on the convergence and stability of our proposed distributed policy followed by comments on its suboptimality with respect to the optimal (unstructured) LQR controller with the same design parameters. We also give a simulation on distributed control of turbocharged diesel engines that showcase the usefulness of D3PI in a practical setting. In order to avoid notation overload, we use a slightly different notation in this part which would be mentioned in case it conflicts with the prior section.

### *3.3.1 The Objective of D3PI Algorithm*

We base our method on a policy iteration scheme where a linear feedback gain is updated at each iteration followed by a policy evaluation step that finds the solution to the corresponding Lyapunov function. With no knowledge of the system parameters $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$, we may follow a $\mathcal{Q}$-learning-based methodology with a trial-and-error nature in which a control agent optimizes some value function from observing the results of its own actions (see [27, 100] for instance). To implement such machinery, an online iterative scheme—say, RLS—is often utilized where at each iteration, data is collected using the current estimate of the policy and the same data is recursively employed to perform an inherent policy evalu-

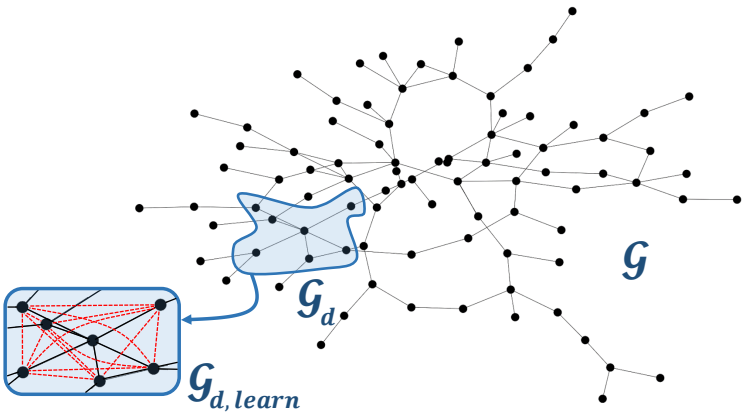



Figure 3.3: Addition of auxiliary links (dashed red) to the subgraph $\mathcal{G}_d$ in D3PI during the policy learning phase. The size of the subgraph depends on the maximum degree of the original graph $\mathcal{G}$.

ation step. Then the policy is updated in a similar manner to the discrete-time LQR update as,

$$\hat{\mathbf{K}} = -\mathbf{F}^{-1}\mathbf{G},$$

where $\mathbf{F}$ and $\mathbf{G}$ are matrix blocks estimated from data in the policy evaluation step. Nonetheless, the main issue with this approach is the fact that the policy obtained in this way, will not respect the hard constraint of $\hat{\mathbf{K}} \in \mathcal{U}_{m,n}^{N}(\mathcal{G})$ as was posed in the optimization (3.5). On the other hand, an arbitrary "projection" of the obtained policy on the set $\mathcal{U}_{m,n}^{N}(\mathcal{G})$ may fail to be stabilizing. Furthermore, such projection onto the intersection of this constraint and stabilizing controllers is not straightforward due to the complicated geometry of the set of stabilizing controllers [30].

As the constraint $\hat{\mathbf{K}} \in \mathcal{U}_{m,n}^{N}(\mathcal{G})$ is strict in distributed control synthesis, we shift our attention from the optimal solution of equation (3.5) towards a suboptimal stabilizing distributed controller with reduced computational burden. Hence, we need to exploit the structure that is incurred to this formulation from the underlying graph. Finally, we note that it is often prohibitive in real-world applications to stop the entire network from functioning for data collection or decision-making purposes. Therefore, we allow temporary

feedback links on a specific smaller portion of the underlying network $\mathcal{G}_d \subseteq \mathcal{G}$ with size $d$ during the learning stage of our algorithm that would be removed subsequently.[6] We will reason that for the learning phase, our method requires observations only from this portion of the graph where few temporary links are appended to make $\mathcal{G}_d$ complete. We show that such interdependency leads to finding control components separately that later on, prescribe a distinction between "self" versus "cooperative" controls in a network. In the next section, we provide the D3PI algorithm and sketch the analysis that connects the ideas presented above along with the convergence study.

### *3.3.2 Main Algorithm*

In this section, we propose and discuss the main algorithm of this part. Given the underlying communication graph $\mathcal{G}$, the system is considered as a black-box, whereas the designer is capable of inserting input signals to the system and observe states. The goal of D3PI is then to find a data-guided suboptimal solution to problem in (3.5) without the knowledge of system parameters $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$. To this end, we take a roundabout and consider the synthesis problem of only a subgraph $\mathcal{G}_d \subseteq \mathcal{G}$ and collect data only from that portion of the system. Before we proceed to the main algorithm, we formalize two notions in order to facilitate the presentation.

**Definition 5.** *Given any subgraph $\mathcal{G}' \subseteq \mathcal{G}$, we define the mapping "*Policy $(\mathcal{V}_{\mathcal{G}'})$*" as the concatenation of all the policies of the agents in $\mathcal{V}_{\mathcal{G}'}$, i.e.,*

$$\text{Policy}\,(\mathcal{V}_{\mathcal{G}'}) \coloneqq \Big[\boldsymbol{u}_1\big(\{\boldsymbol{x}_j | j \in \mathcal{N}_1\}\big)\ ;\ \boldsymbol{u}_2\big(\{\boldsymbol{x}_j | j \in \mathcal{N}_2\}\big)\ ;\ \cdots\ ;\ \boldsymbol{u}_{|\mathcal{V}_{\mathcal{G}'}|}\big(\{\boldsymbol{x}_j | j \in \mathcal{N}_{|\mathcal{V}_{\mathcal{G}'}|}\}\big)\Big],$$

*where each $\boldsymbol{u}_i(\{\boldsymbol{x}_j | j \in \mathcal{N}_i\})$ denotes the control policy of agent $i$ in the subgraph $\mathcal{G}'$ as a mapping from $\{\boldsymbol{x}_j | j \in \mathcal{N}_i\}$ to $\mathbb{R}^m$. Furthermore, we use* $\text{Policy}(\mathcal{V}_{\mathcal{G}'})\big|_t$ *to denote the realization of these policies (i.e. the control signals) at time $t$. Similarly, we define ,*

$$\text{State}\,(\mathcal{V}_{\mathcal{G}'}) \coloneqq \Big[\boldsymbol{x}_1\ ;\ \boldsymbol{x}_2\ ;\ \cdots\ ;\ \boldsymbol{x}_{|\mathcal{V}_{\mathcal{G}'}|}\Big].$$

[6] Of our particular interest are those systems that can be modeled with graphs in which $d \ll N$. Examples of such occurrences are prevalent in grid-based applications such as social media, power grids, etc.

The D3PI algorithm is introduced in Algorithm 4. During the learning phase, we include temporary "auxiliary" links to $\mathcal{G}_d$ and make the communication graph complete. We show such distinction by $\mathcal{G}_{d,\text{learn}}$ where $|\mathcal{V}_{\mathcal{G}_d}| = \left|\mathcal{V}_{\mathcal{G}_{d,\text{learn}}}\right|$ and $\mathcal{G}_{d,\text{learn}} = \mathcal{K}(\mathcal{G}_d)$. Inherent to D3PI is a policy iteration on $\mathcal{G}_{d,\text{learn}}$ that finds components $K_k$ and $L_k$ which intuitively represent "self" and "cooperative" controls at iteration $k$, respectively. Even during the learning phase, we utilize these components in order to design and update an effective stabilizing controller for the remainder of the agents in $\mathcal{G} \setminus \mathcal{G}_{d,\text{learn}}$. We do so by ensuring that during the learning phase, information is exchanged unidirectional from $\mathcal{G}_{d,\text{learn}}$ to the rest of the network, therefore, the policy of the agents in $\mathcal{G} \setminus \mathcal{G}_{d,\text{learn}}$ is dependent on those in $\mathcal{G}_{d,\text{learn}}$, but not vice versa. Finally, with the convergence of the algorithm, we remove the temporary links so as to get back to the original communication topology, and propose a final suboptimal stabilizing scheme for the entire network that follows the original network topology. In the learning stage of D3PI, we use an RLS-based recursion to estimate the unknown parameters in the cost matrix at iteration $k$, referred to as $\widetilde{\mathbf{H}}_k$. This entire process is performed in the Subgraph Policy Evaluation (SPE) (Algorithm 5) given $\mathcal{G}$, $\mathcal{G}_{d,\text{learn}}$, the mapping policy($\mathcal{V}_{\mathcal{G}_d}$), and the previous estimate of $\widetilde{\mathbf{H}}_{k-1}$.[7] As will be discussed later, $\widetilde{\mathbf{H}}_k$ contains the required information to find $K_k$ and $L_k$ from data. We extract this square matrix through a recursive update on the vector $\boldsymbol{\theta}_{k-1}$, derived from half-vectorization of $\widetilde{\mathbf{H}}_{k-1}$, solving the RLS for the linear equation $\mathcal{R}(\tilde{\mathbf{x}}_t, \tilde{\mathbf{u}}_t) = \boldsymbol{\zeta}_t^\top \boldsymbol{\theta}_{k-1}$, where $\mathcal{R}(\tilde{\mathbf{x}}_t, \tilde{\mathbf{u}}_t)$ denotes the local cost and $\boldsymbol{\zeta}_t \in \mathbb{R}^p$ contains the data measurements.[8] The adaptive nature of the algorithm entails the exploration signal $\boldsymbol{e}_t$ to be augmented to the policy vector in order to satisfy persistence of excitation. In our setup, $\boldsymbol{e}_t$ is sampled from a normal distribution $\boldsymbol{e}_t \sim \mathcal{N}(\mathbf{0}, \Sigma)$ where the choice of the variance $\Sigma$ is problem-specific.[9] We denote by $\mathcal{P}_k$ the projection factor that is reset to $\mathcal{P}^\circ \succ 0$ at the beginning of each iteration. Convergence of the SPE—guaranteed based on the persistence of excitation condition—is followed by the update of $\widetilde{\mathbf{H}}_k$ that encodes the necessary information to get $K_k$ and $L_k$.

[7] Tilded notations are secured for the parameters related to $\mathcal{G}_{d,\text{learn}}$.

[8] We use the increments $k$ for the policy update and $t$ for the data collection.

[9] In practice, excitation of the input is a subtle task and has been implemented in a variety of forms such as random noise [27], sinusoidal signals [91], and exponentially decaying noise [100].

**Algorithm 4** Data-Driven Distributed Policy Iteration (D3PI)

**Initialization** ($t \leftarrow 1,\ k \leftarrow 1$)

Obtain graph $\mathcal{G}$ and choose $\mathcal{G}_d \subseteq \mathcal{G}$ with $d = d_{\max}(\mathcal{G}) + 1$

Obtain $Q_1$, $Q_2$, $R$ and set $\widetilde{Q} \leftarrow Q_1 + dQ_2$

Randomize $\tilde{\mathbf{x}}_1 \in \mathbb{R}^{dn}$ and $\widetilde{\mathbf{H}}_0 \in \mathbb{R}^{p\times p}$ with $p = d(n+m)$ and set $\tau_1 \leftarrow 0$

Set $\mathcal{P}^\circ \leftarrow \beta \mathrm{I}_{p(p+1)/2}$ for large enough $\beta > 0$ and fix variance $\Sigma$

Choose $K_1$ that stabilizes (3.1) and set $L_1 \leftarrow \mathbf{0}_{m\times n}$ and $\Delta K_1 \leftarrow K_1 - L_1$

Switch ON temporary links in $\mathcal{G}_d$ to set $\mathcal{G}_{d,\text{learn}} \leftarrow \mathcal{K}(\mathcal{G}_d)$

**While** $K_k$ **and** $L_k$ **not converged, do**

Set $\mathcal{P}_k \leftarrow \mathcal{P}^\circ$

Set $\text{Policy}_k(\mathcal{V}_\mathcal{G})$ such that for each $i \in \mathcal{V}_\mathcal{G}$,

$$\boldsymbol{u}_i \leftarrow \Delta K_k \boldsymbol{x}_i + L_k \left[\mathbb{I}_{\{i\in\mathcal{V}_{\mathcal{G}\setminus\mathcal{G}_d}\}} \sum_{j\in\mathcal{N}_i} \tfrac{\tau_k}{d-1}\boldsymbol{x}_j + \mathbb{I}_{\{i\in\mathcal{V}_{\mathcal{G}_d}\}} \sum_{j\in\mathcal{V}_{\mathcal{G}_d}\setminus\{i\}} \boldsymbol{x}_j\right]$$

Evaluate $\text{Policy}_k(\mathcal{V}_{\mathcal{G}_d})$ by obtaining $\widetilde{\mathbf{H}}_k \leftarrow \text{SPE}\left(\mathcal{G},\ \mathcal{G}_{d,\text{learn}},\ \text{Policy}_k(\mathcal{V}_\mathcal{G}),\ \widetilde{\mathbf{H}}_{k-1}, \mathcal{P}_k\right)$

Recover matrix blocks

$$\widetilde{\mathbf{H}}_{11,k} \leftarrow \widetilde{\mathbf{H}}_k[1:dn,\ 1:dn],$$
$$\widetilde{\mathbf{H}}_{21,k} \leftarrow \widetilde{\mathbf{H}}_k[dn+1:dn+dm,\ 1:dn],$$
$$\widetilde{\mathbf{H}}_{22,k} \leftarrow \widetilde{\mathbf{H}}_k[dn+1:dn+dm,\ dm+1:dn+dm]$$

Set

$$X_1 \leftarrow \widetilde{\mathbf{H}}_{11,k}[1:n,\ 1:n], \quad Y_1 \leftarrow \widetilde{\mathbf{H}}_{22,k}[1:m,\ 1:m], \quad Z_1 \leftarrow \widetilde{\mathbf{H}}_{21,k}[1:m,\ 1:n],$$
$$X_2 \leftarrow \widetilde{\mathbf{H}}_{11,k}[1:n,\ n+1:2n], \quad Y_2 \leftarrow \widetilde{\mathbf{H}}_{22,k}[1:m,\ m+1:2m], \quad Z_2 \leftarrow \widetilde{\mathbf{H}}_{21,k}[1:m,\ n+1:2n],$$
$$\Delta X \leftarrow X_1 - X_2, \quad \Delta Y \leftarrow Y_1 - Y_2, \quad \Delta Z \leftarrow Z_1 - Z_2$$

Set $F \leftarrow \left(Y_1 - (d-1)Y_2\big(Y_1 + (d-2)Y_2\big)^{-1}Y_2\right)^{-1}$ and $G \leftarrow (Y_1 + (d-1)Y_2)^{-1}\, Y_2\, (Y_1 - Y_2)^{-1}$

Update,

$$K_{k+1} \leftarrow FZ_1 + (d-1)GZ_2, \quad L_{k+1} \leftarrow FZ_2 + GZ_1 + (d-2)GZ_2, \quad \Delta K_{k+1} \leftarrow K_{k+1} - L_{k+1}$$

Set $\Xi_{k+1} \leftarrow \Delta X - \widetilde{Q} + Q_2 + \Delta K_{k+1}^\top \Delta Z + \Delta Z^\top \Delta K_{k+1} + \Delta K_{k+1}^\top \big(\Delta Y - R\big)\Delta K_{k+1}$

Set $\gamma_{k+1} \leftarrow \sigma_{\min}\Big(\Delta K_{k+1}^\top R \Delta K_{k+1} + \widetilde{Q}\Big) \Big/ \sigma_{\max}\Big(\Xi_{k+1} + L_{k+1}^\top(\Delta Y - R)L_{k+1}\Big)$

Set $\tau_{k+1} \leftarrow \sqrt{\gamma_{k+1}^2/(1+\gamma_{k+1})}$

$k \leftarrow k+1$

Switch OFF the temporary links and retrieve $\mathcal{G}_d$

Set $\text{Policy}_k(\mathcal{V}_\mathcal{G})$ such that for each $i \in \mathcal{V}_\mathcal{G}$, $\boldsymbol{u}_i \leftarrow \Delta K_k \boldsymbol{x}_i + \frac{\tau_k}{d-1} L_k \sum_{j\in\mathcal{N}_i} \boldsymbol{x}_j$

This is done by recovering the block matrices $X_1$, $X_2$, $Y_1$, $Y_2$, $Z_1$, and $Z_2$ from $\widetilde{\mathbf{H}}_k$ that are further used to form the intermediate variables $F$ and $G$.[10] Such recovery of meaningful blocks from $\widetilde{\mathbf{H}}_k$ is due to the specific matrix structure that is resulted from adding extra links to $\mathcal{G}_d$ and will be discussed further in the following section. Each iteration loop is then ended by updating the parameters $\gamma_k$ and $\tau_k$ which are effective in the stability analysis of the proposed controller throughout the process. With the convergence of D3PI, $\mathcal{G}_d$ is retrieved by removing the temporary links and the distributed policy is extrapolated to the entire graph $\mathcal{G}$.

**Algorithm 5** Subgraph Policy Evaluation (SPE)

1: **Input:** Graph $\mathcal{G}$, subgraph $\mathcal{G}' \subseteq \mathcal{G}$, Policy($\mathcal{V}_\mathcal{G}$), $\mathbf{H}$, $\mathcal{P}$

2: **Output:** The updated cost matrix $\mathbf{H}^+$ associated with $\mathcal{G}'$

3: **While H not converged, do**

4: Set $\tilde{\mathbf{x}}_t \leftarrow \text{State}(\mathcal{V}_{\mathcal{G}'})\big|_t$ and $\tilde{\mathbf{u}}_t \leftarrow \text{Policy}(\mathcal{V}_{\mathcal{G}'})\big|_t$

5: Choose $\boldsymbol{e}_t \sim \mathcal{N}(\mathbf{0}, \Sigma)$ and update $\text{Policy}(\mathcal{V}_{\mathcal{G}'})\big|_t$ for all $i \in \mathcal{V}_{\mathcal{G}'}$ as $\tilde{\mathbf{u}}_t \leftarrow \tilde{\mathbf{u}}_t + \boldsymbol{e}_t$

6: Let entire $\mathcal{G}$ run under $\text{Policy}(\mathcal{V}_\mathcal{G})$ and collect $\text{State}(\mathcal{V}_{\mathcal{G}'})\big|_{t+1}$ only from $\mathcal{G}'$

7: Set $\tilde{\mathbf{x}}_{t+1} \leftarrow \text{State}(\mathcal{V}_{\mathcal{G}'})\big|_{t+1}$ and $\tilde{\mathbf{u}}_{t+1} \leftarrow \text{Policy}(\mathcal{V}_{\mathcal{G}'})\big|_{t+1}$

8: Set $\tilde{\mathbf{z}}_t \leftarrow [\tilde{\mathbf{x}}_t;\ \tilde{\mathbf{u}}_t]$ and $\tilde{\mathbf{z}}_{t+1} \leftarrow [\tilde{\mathbf{x}}_{t+1};\ \tilde{\mathbf{u}}_{t+1}]$ and form $\boldsymbol{\phi}_t \leftarrow \tilde{\mathbf{z}}_t - \tilde{\mathbf{z}}_{t+1}$

9: Compute $\boldsymbol{\zeta}_t \leftarrow \big[\phi_{1,t}^2 \ \ \phi_{1,t}\phi_{2,t} \ \ \cdots \ \ \phi_{1,t}\phi_{p,t} \mid \phi_{2,t}^2 \ \ \phi_{2,t}\phi_{3,t} \ \ \cdots \ \ \phi_{2,t}\phi_{p,t} \mid \cdots \mid \phi_{p,t}^2\big]^\top$

10: Compute $\mathcal{R}(\tilde{\mathbf{x}}_t, \tilde{\mathbf{u}}_t) \leftarrow \tilde{\mathbf{x}}_t^\top \big(\mathrm{I} \otimes \widetilde{Q} - \mathbb{1}\mathbb{1}^\top \otimes Q_2\big)\tilde{\mathbf{x}}_t + \tilde{\mathbf{u}}_t^\top \big(\mathrm{I} \otimes R\big)\tilde{\mathbf{u}}_t$

11: Form $\boldsymbol{\theta} = \text{vech}(\mathbf{H})$ and use RLS update,

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \frac{\mathcal{P}\boldsymbol{\zeta}_t\big(\mathcal{R}(\tilde{\mathbf{x}}_t, \tilde{\mathbf{u}}_t) - \boldsymbol{\zeta}_t^\top \boldsymbol{\theta}\big)}{1 + \boldsymbol{\zeta}_t^\top \mathcal{P}\boldsymbol{\zeta}_t}, \qquad \mathcal{P} \leftarrow \mathcal{P} - \frac{\mathcal{P}\boldsymbol{\zeta}_t\boldsymbol{\zeta}_t^\top \mathcal{P}}{1 + \boldsymbol{\zeta}_t^\top \mathcal{P}\boldsymbol{\zeta}_t} \tag{3.31}$$

12: Find $\mathbf{H}^+ = \text{vech}^{-1}(\boldsymbol{\theta})$ and update $\mathbf{H} \leftarrow \mathbf{H}^+$

13: $t \leftarrow t + 1$

**Remark 7.** *Adding temporary links within the subgraph $\mathcal{G}_d$ is a way to learn optimal $K_k$ and $L_k$ for the subgraph $\mathcal{G}_{d,\text{learn}}$ by fully examining the dynamical interference among the agents. Moreover, initializing $K_k$ such that (3.1) is Schur stable—as assumed in our algorithm—has become a standard in the data-driven control literature. However, we acknowledge that this*

[10] Matrix inversions on line 14 of Algorithm 4 will be justified in the analysis.

*is not a trivial assumption and often impractical, in particular, when the system is unknown. While we stick to this assumption for the brevity of analysis, the interested reader is referred to the recent works [40, 156] to learn how the issue can be circumvented for specific classes of systems.*

### *3.3.3 Computational Complexity*

Note that the inverse operations on line 14 of the algorithm occur on matrices of size $m \times m$ and are computationally negligible. Furthermore, the complexity of finding extreme singular values—as on line 17 in Algorithm 4—is shown to be as low as $\mathcal{O}(n^2)$ [42]. Hence, the computational complexity of D3PI is mainly originated from the SPE recursion that is equivalent to the complexity of RLS for the number of unknown system parameters in $\mathcal{G}_d$ and is equal to $\mathcal{O}\left(d^2(n+m)^2\right)$ [80]. This implies that the complexity of the algorithm will be fixed for any number of agents $N$, as long as the maximum degree of the graph remains unchanged.

### *3.3.4 Analysis Prerequisites*

Before we proceed to the main analysis, we initiate the building blocks and tools that we need for the proofs and the analysis of our algorithm. We first provide some insight on how our setup is connected to the classic model-based LQR machinery and some previously established results that we leverage from the literature. Next, we present some lemmas that will be used later on in the proof of the main results in the next section.

*Underlying Model of the Subsystem $\mathcal{G}_d$*

The configuration of the synthesis problem in D3PI intertwines an online recursion on the subsystem corresponding to $\mathcal{G}_d$ and the original system $\mathcal{G}$. To distinguish between these two frameworks, we use the tilded notation to present system parameters of $\mathcal{G}_d$ in contrast to the hatted notion for $\mathcal{G}$. In particular, from Equation (3.1) the dynamics of the subgraph $\mathcal{G}_d$ can be cast as,

$$\tilde{\mathbf{x}}_{t+1} = \widetilde{\mathbf{A}}\tilde{\mathbf{x}}_t + \widetilde{\mathbf{B}}\tilde{\mathbf{u}}_t, \tag{3.32}$$

where $\tilde{\mathbf{x}}_t$ and $\tilde{\mathbf{u}}_t$ are formed from concatenation of state and control signals in $\mathcal{G}_d$—recall that $\tilde{\mathbf{u}}_t$ is also denoted by Policy$(\mathcal{V}_{\mathcal{G}_d})$ in the algorithm to emphasize the functional independence of Algorithms 4 and 5—and $\widetilde{\mathbf{A}} = \mathrm{I}_d \otimes A$ and $\widetilde{\mathbf{B}} = \mathrm{I}_d \otimes B$ form the dynamics parameterization of $\mathcal{G}_d$. Also define $\widetilde{\mathbf{Q}} = \mathrm{I}_d \otimes \widetilde{Q} - \mathbb{1}\mathbb{1}^\top \otimes Q_2$ and $\widetilde{\mathbf{R}} = \mathrm{I}_d \otimes R$. From the Bellman equation [21] for the LQR problem with parameters $(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \widetilde{\mathbf{Q}}, \widetilde{\mathbf{R}})$, the cost-to-go matrix of $\mathcal{G}_d$ is correlated with the LQR cost,

$$\tilde{\mathbf{x}}_t^\top \widetilde{\mathbf{P}}_k \tilde{\mathbf{x}}_t = \mathcal{R}(\tilde{\mathbf{x}}_t, \tilde{\mathbf{u}}_t) + \tilde{\mathbf{x}}_{t+1}^\top \widetilde{\mathbf{P}}_k \tilde{\mathbf{x}}_{t+1}, \tag{3.33}$$

where $\mathcal{R}(\tilde{\mathbf{x}}_t, \tilde{\mathbf{u}}_t) = \tilde{\mathbf{x}}_t^\top \widetilde{\mathbf{Q}} \tilde{\mathbf{x}}_t + \tilde{\mathbf{u}}_t^\top \widetilde{\mathbf{R}} \tilde{\mathbf{u}}_t$ is the one-step cost, $\widetilde{\mathbf{P}}_k$ is the solution to the Lyapunov equation,

$$\widetilde{\mathbf{P}}_k = \big(\widetilde{\mathbf{A}} + \widetilde{\mathbf{B}}\widetilde{\mathbf{K}}_k\big)^\top \widetilde{\mathbf{P}}_k \big(\widetilde{\mathbf{A}} + \widetilde{\mathbf{B}}\widetilde{\mathbf{K}}_k\big) + \widetilde{\mathbf{Q}} + \widetilde{\mathbf{K}}_k^\top \widetilde{\mathbf{R}} \widetilde{\mathbf{K}}_k, \tag{3.34}$$

and $\widetilde{\mathbf{K}}_k$ is the controller estimate at iteration $k$. The dynamic programming solution to the LQR problem suggests a linear feedback form $\tilde{\mathbf{u}}_t = \widetilde{\mathbf{K}}_k \tilde{\mathbf{x}}_t$ for the subsystem $\mathcal{G}_d$ at each iteration. As mentioned earlier in the preliminaries, combining (3.32) and (3.33) with some rearrangements result in

$$\tilde{\mathbf{x}}_t^\top \widetilde{\mathbf{P}}_k \tilde{\mathbf{x}}_t = \tilde{\mathbf{z}}_t^\top \begin{bmatrix} \widetilde{\mathbf{Q}} + \widetilde{\mathbf{A}}^\top \widetilde{\mathbf{P}}_k \widetilde{\mathbf{A}} & \widetilde{\mathbf{A}}^\top \widetilde{\mathbf{P}}_k \widetilde{\mathbf{B}} \\ \widetilde{\mathbf{B}}^\top \widetilde{\mathbf{P}}_k \widetilde{\mathbf{A}} & \widetilde{\mathbf{R}} + \widetilde{\mathbf{B}}^\top \widetilde{\mathbf{P}}_k \widetilde{\mathbf{B}} \end{bmatrix} \tilde{\mathbf{z}}_t = \tilde{\mathbf{z}}_t^\top \begin{bmatrix} \widetilde{\mathbf{H}}_{11,k} & \widetilde{\mathbf{H}}_{12,k} \\ \widetilde{\mathbf{H}}_{21,k} & \widetilde{\mathbf{H}}_{22,k} \end{bmatrix} \tilde{\mathbf{z}}_t = \tilde{\mathbf{z}}_t^\top \widetilde{\mathbf{H}}_k \tilde{\mathbf{z}}_t, \tag{3.35}$$

where $\tilde{\mathbf{z}}_t = [\tilde{\mathbf{x}}_t;\ \tilde{\mathbf{u}}_t]$. Note that the discrete-time LQR policy update can be obtained from,

$$\widetilde{\mathbf{K}}_{k+1} = -\left(\widetilde{\mathbf{R}} + \widetilde{\mathbf{B}}^\top \widetilde{\mathbf{P}}_k \widetilde{\mathbf{B}}\right)^{-1} \widetilde{\mathbf{B}}^\top \widetilde{\mathbf{P}}_k \widetilde{\mathbf{A}} = -\widetilde{\mathbf{H}}_{22,k}^{-1} \widetilde{\mathbf{H}}_{21,k}, \tag{3.36}$$

and the Lyapunov equation in (3.34) can be rewritten as,

$$\widetilde{\mathbf{P}}_k = \widetilde{\mathbf{H}}_{11,k} + \widetilde{\mathbf{H}}_{12,k}\widetilde{\mathbf{K}}_k + \widetilde{\mathbf{K}}_k^\top \widetilde{\mathbf{H}}_{21,k} + \widetilde{\mathbf{K}}_k^\top \widetilde{\mathbf{H}}_{22,k} \widetilde{\mathbf{K}}_k. \tag{3.37}$$

Hence, $\widetilde{\mathbf{H}}_k$ provides the required information to perform both policy update and policy evaluation steps in a policy iteration algorithm. We will show that because of the particular structure of our setup, $\widetilde{\mathbf{H}}_k$ enjoys a special block pattern, justifying the recovery of the block matrices $X_1$, $X_2$, $Y_1$, $Y_2$, $Z_1$, and $Z_2$ from $\widetilde{\mathbf{H}}_k$ in D3PI. Such constraint on the subspace of

this particular class of matrices $\widetilde{\mathbf{H}}_k$ can also be employed in order to potentially reduce the computational complexity of the algorithm. D3PI leverages this idea to implicitly learn $\widetilde{\mathbf{H}}_k$ from data and exploit these matrix blocks in order to find a suboptimal solution to problem (3.5).

### *3.3.5 Linear Algebraic Lemmas*

In the remainder of this section, we first state a well-known result to ensure completeness, and then propose two new linear algebraic facts as vital complements to our analysis.

**Lemma 15.** *The following relations hold,*

1. *([85]) When $X \succ 0$,*

$$M^\top X N + N^\top X M \succeq -(aM^\top X M + \frac{1}{a} N^\top X N),$$
$$M^\top X N + N^\top X M \preceq aM^\top X M + \frac{1}{a} N^\top X N,$$

   *where $M, N \in \mathbb{R}^{n\times m}$ with $n \geq m$ and $a > 0$.*

2. *([98]) Suppose that $A \in \mathbb{R}^{n\times n}$ has spectral radius bounded by 1, i.e., $\rho(A) < 1$. Then*

$$A^\top X A + Q - X = 0$$

   *has a unique solution,*

$$X = \sum_{j=0}^{\infty} (A^\top)^j Q A^j.$$

   *Furthermore, if $Q \succ 0$, then $X \succ 0$.*

3. *([85]) The following equation holds for matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ with compatible dimensions,*

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{H}^{-1} & -\boldsymbol{H}^{-1}\boldsymbol{B}\boldsymbol{D}^{-1} \\ -\boldsymbol{D}^{-1}\boldsymbol{C}\boldsymbol{H}^{-1} & \boldsymbol{D}^{-1} + \boldsymbol{D}^{-1}\boldsymbol{C}\boldsymbol{H}^{-1}\boldsymbol{B}\boldsymbol{D}^{-1} \end{bmatrix},$$

   *where $\boldsymbol{D}$ and $\boldsymbol{H} = \boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}$ are invertible.*

4. *(Matrix Inversion Lemma [172]) The following holds,*

$$(A+UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1}+VA^{-1}U)^{-1}VA^{-1},$$

*for matrices $A$, $U$, $C$, and $V$ with compatible dimensions where $A$, $C$, and $A+UCV$ are invertible.*

**Lemma 16.** *Suppose $A$ and $B$ are symmetric matrices such that $A$, $A-B$, and $A+(n-1)B$ are all invertible for some $n \in \mathbb{N}_0$. Then the following algebraic relations hold:*

1. $\big(A+nB\big)\big(A+(n-1)B\big)^{-1}\big(A-B\big) = A - nB\big(A+(n-1)B\big)^{-1}B$

2. $\big(A+nB\big)\big(A+(n-1)B\big)^{-1}\big(A-B\big) = \big(A-B\big)\big(A+(n-1)B\big)^{-1}\big(A+nB\big)$

3. $\big(A+nB\big)\big(A-B\big)^{-1}B = B\big(A-B\big)^{-1}\big(A+nB\big)$

*Proof.* 1) $\big(A+nB\big)\big(A+(n-1)B\big)^{-1}\big(A-B\big)$

$$\begin{aligned}
&= \Big(\big(A+(n-1)B\big)+B\Big)\big(A+(n-1)B\big)^{-1}\big(A-B\big)\\
&= \Big(\mathrm{I}+B\big(A+(n-1)B\big)^{-1}\Big)\big(A-B\big)\\
&= A-B+B\big(A+(n-1)B\big)^{-1}A - B\big(A+(n-1)B\big)^{-1}B\\
&= A+B\Big(\big(A+(n-2)B\big)^{-1}A-\mathrm{I}\Big) - B\big(A+(n-1)B\big)^{-1}B\\
&= A+B\big(A+(n-2)B\big)^{-1}\Big(A-\big(A+(n-2)B\big)\Big) - B\big(A+(n-1)B\big)^{-1}B\\
&= A-(n-1)B\big(A+(n-1)B\big)^{-1}B - B\big(A+(n-1)B\big)^{-1}B\\
&= A-nB\big(A+(n-1)B\big)^{-1}B
\end{aligned}$$

2) $\big(A+nB\big)\big(A+(n-1)B\big)^{-1}\big(A-B\big)$

$$\begin{aligned}
&= -n\big(A-B\big)\big(A+(n-1)B\big)^{-1}\big(A-B\big) + (n+1)A\big(A+(n-1)B\big)^{-1}\big(A-B\big)\\
&= -n\big(A-B\big)\big(A+(n-1)B\big)^{-1}\big(A-B\big) + (n+1)\big(\mathrm{I}+(n-1)BA^{-1}\big)^{-1}\big(A-B\big)\\
&= -n\big(A-B\big)\big(A+(n-1)B\big)^{-1}\big(A-B\big) + (n+1)\big(A-B\big)\Big(\big(\mathrm{I}+(n-1)BA^{-1}\big)\big(A-B\big)\Big)^{-1}\big(A-B\big)\\
&= -n\big(A-B\big)\big(A+(n-1)B\big)^{-1}\big(A-B\big) + (n+1)\big(A-B\big)\Big(\big(A-B\big)\big(\mathrm{I}+(n-1)A^{-1}B\big)\Big)^{-1}\big(A-B\big)\\
&= -n\big(A-B\big)\big(A+(n-1)B\big)^{-1}\big(A-B\big) + (n+1)\big(A-B\big)\big(A+(n-1)B\big)^{-1}A\\
&= \big(A-B\big)\big(A+(n-1)B\big)^{-1}\big(A+nB\big)
\end{aligned}$$

3) $\big(A+nB\big)\big(A-B\big)^{-1}B = \big(A-B+(n+1)B\big)\big(A-B\big)^{-1}B$

$= \big(\mathrm{I}+(n+1)B(A-B)^{-1}\big)B = B\big(\mathrm{I}+(n+1)(A-B)^{-1}B\big) = B\big(A-B\big)^{-1}\big(A+nB\big)$

□

**Lemma 17.** *Suppose* $\boldsymbol{N}_r \in \mathrm{PL}(r\times n,\mathbb{R})$ *for some* $n$ *and* $r\geq 2$ *such that* $\boldsymbol{N}_r = \mathrm{I}_r\otimes\big(A-B\big)+\mathbb{1}_r\mathbb{1}_r^\top\otimes B$, *for some* $A\in\mathrm{GL}(n,\mathbb{R})\cap\mathbb{S}^{n\times n}$ *and* $B\in\mathbb{S}^{n\times n}$. *Then the followings hold:*

1. $\det(\boldsymbol{N}_r) = \det(A-B)^{r-1}\det(A+(r-1)B)$

2. *If* $\boldsymbol{N}_r\succ 0$, *then* $A-B$, $A+\ell B\succ 0$, *and* $A-\ell B\,(A+(\ell-1)B)^{-1}\,B\succ 0$ *for* $\ell = 1,2,\cdots,r-1$.

3. $\boldsymbol{N}_r^{-1}\in\mathrm{PL}(r\times n,\mathbb{R})$, *i.e.,*

$$\boldsymbol{N}_r^{-1} = \mathrm{I}_r\otimes\big(F+G\big)-\mathbb{1}_r\mathbb{1}_r^\top\otimes G,$$

*with* $F$ *and* $G$ *defined as,*

$$F = \Big(A-(r-1)B\big(A+(r-2)B\big)^{-1}B\Big)^{-1},$$
$$G = \big(A+(r-1)B\big)^{-1}B\big(A-B\big)^{-1}$$

4. *If* $\boldsymbol{M}_r = \mathrm{I}_r\otimes\big(C-D\big)+\mathbb{1}_r\mathbb{1}_r^\top\otimes D$ *then,*

$$\boldsymbol{N}_r\boldsymbol{M}_r = \mathrm{I}_r\otimes\big(A-B\big)\big(C-D\big)+\mathbb{1}_r\mathbb{1}_r^\top\otimes\Big(B\big(C-D\big)+\big(A-B\big)D+nBD\Big)$$

*Proof.* 1) We prove the claim by induction. First, note that since $\mathbf{N}_r$ is invertible, $A$ as a leading principal minor, is also invertible. For $r=1$, the result follows trivially from definition. For $r=2$, from the Schur complement of $\mathbf{N}_2$,

$$\begin{aligned}\det(\mathbf{N}_2) &= \det(A)\det(A-BA^{-1}B)\\ &= \det(A)\det(\mathrm{I}_2-BA^{-1})\det(\mathrm{I}_2+BA^{-1})\det(A)\\ &= \det(A-B)\det(A+B).\end{aligned}$$

Let the claim hold for $r = p-1$. Then for $r = p$ we again use the Schur complement to get,

$$\begin{aligned}\det(\mathbf{N}_p) &= \det(A)\det\left(\mathbf{N}_{p-1} - \mathbb{1}\mathbb{1}^\top \otimes BA^{-1}B\right)\\ &= \det(A)\det(A-B)^{r-2}\det(A - BA^{-1} + B - BA^{-1}B).\end{aligned}$$

Note that $B - BA^{-1}B = (A-B)A^{-1}B$ and $A - BA^{-1}B = (A-B)A^{-1}(A+B)$. Then, plugging these into the latter equality results in the final form of $\det(\mathbf{N}_p)$.

2) From item 1 of the current lemma,

$$\det(\mathbf{N}_r - \lambda \mathrm{I}_r \otimes \mathrm{I}) = \det(A - \lambda\mathrm{I} - B)^{r-1}\det(A - \lambda\mathrm{I} + (r-1)B),$$

implying that the spectrum of $A-B$ and $A+(r-1)B$ are subsets of the eigenvalues of $\mathbf{N}_r$. Hence, $\mathbf{N}_r \succ 0$ results in $A - B \succ 0$ and $A + (r-1)B \succ 0$. Furthermore, $\mathbf{N}_r \succ 0$ if and only if all the leading principal minors of $\mathbf{N}_r$ are positive. This implies that $A + \ell' B \succ 0$ for $\ell' = 1, \cdots, r-2$. Lastly, from item 1 of Lemma 16,

$$A - \ell B\big(A + (\ell-1)B\big)^{-1}B = \big(A+\ell B\big)\big(A+(\ell-1)B\big)^{-1}\big(A-B\big) \succ 0, \qquad \forall \ell = 1, \cdots, r-1.$$

3) Since $\mathbf{N}_r$ is invertible, $\mathbf{N}_{r-1} - L_{r-1}A^{-1}L_{r-1}^\top$ is invertible by Schur complement where $L_{r-1} = \mathbb{1}_{r-1} \otimes B$. We follow the proof by induction. For $r = 2$, from item item 3 in Lemma 15,

$$\mathbf{N}_2^{-1} = \begin{bmatrix} H^{-1} & -H^{-1}BA^{-1} \\ -A^{-1}BH^{-1} & A^{-1} + A^{-1}BH^{-1}BA^{-1}\end{bmatrix},$$

where $H = A - BA^{-1}B$ is the Schur complement of $A$. From item 4 in Lemma 15, $H^{-1} = A^{-1} + A^{-1}BH^{-1}BA^{-1}$, establishing the recurrence of diagonal blocks. Furthermore,

$$A^{-1}BH^{-1} = A^{-1}B\big(A^{-1} + A^{-1}BH^{-1}BA^{-1}\big) = \big(A^{-1}H + A^{-1}BA^{-1}B\big)H^{-1}BA^{-1} = H^{-1}BA^{-1}.$$

Also, from item 1 in Lemma 16 with $n = 1$, we get that

$$\begin{aligned}A^{-1}BH^{-1} &= A^{-1}B\big(A-B\big)^{-1}A\big(A+B\big)^{-1} = A^{-1}B\big(A+B\big)^{-1}A\big(A-B\big)^{-1}\\ &= A^{-1}B\big(\mathrm{I} + A^{-1}B\big)^{-1}\big(A-B\big)^{-1} = \big(\mathrm{I} - (\mathrm{I}+A^{-1}B)^{-1}\big)\big(A-B\big)^{-1} = \big(A+B\big)^{-1}B\big(A-B\big)^{-1},\end{aligned}$$

where we also used $\big(A-B\big)^{-1}A\big(A+B\big)^{-1} = \big(A+B\big)^{-1}A\big(A-B\big)^{-1}$ derived from Lemma 16 item 2. Hence,

$$\mathbf{N}_2^{-1} = \mathrm{I}_2 \otimes \big(H^{-1} + H^{-1}BA^{-1}\big) - \mathbb{1}_2\mathbb{1}_2^\top \otimes \big(H^{-1}BA^{-1}\big).$$

Assume that the claim holds for $r = p$. To extend the result to $r = p+1$, again from Lemma 15 items 3 and 4,

$$\mathbf{N}_{p+1}^{-1} = \begin{bmatrix} A & L_p^\top \\ L_p & \mathbf{N}_p \end{bmatrix}^{-1} = \begin{bmatrix} P^{-1} & -P^{-1}L_p^\top\mathbf{N}_p^{-1} \\ -\mathbf{N}_p^{-1}L_pP^{-1} & \big(\mathbf{N}_p - L_pA^{-1}L_p\big)^{-1} \end{bmatrix},$$

where $P = A - L_p^\top\mathbf{N}_p^{-1}L_p$ and $L_p = \mathbb{1}_p \otimes B$. Let $\mathbf{N}_p^{-1} = \mathrm{I}_p \otimes \big(F_p + G_p\big) - \mathbb{1}_p\mathbb{1}_p^\top \otimes G_p$ where,

$$\begin{aligned} F_p &= \Big(A - (p-1)B\big(A + (p-2)B\big)^{-1}B\Big)^{-1}, \\ G_p &= \big(A + (p-1)B\big)^{-1}B\big(A - B\big)^{-1}, \end{aligned}$$

where the inversions are valid from item 2 of the current Lemma. By simplification we get $P = A - nB\big(F_p + (n-1)G_p\big)B$ and from Lemma 16 items 1 and 2,

$$\begin{aligned} F_p + (n-1)G_p &= \Big(A - (p-1)B\big(A + (p-2)B\big)^{-1}B\Big)^{-1} + (n-1)\big(A + (p-1)B\big)^{-1}B\big(A - B\big)^{-1} \\ &= \big(A + (p-1)B\big)^{-1}\big(A + (p-2)B\big)\big(A - B\big)^{-1} - (n-1)\big(A + (p-1)B\big)^{-1}B\big(A - B\big)^{-1} \\ &= \big(A + (p-1)B\big)^{-1}, \end{aligned}$$

which, in turn, gives $P = A - nB\big(A + (p-1)B\big)^{-1}B$. Also with similar reasoning, each block of $P^{-1}L_p^\top\mathbf{N}_p^{-1}$ has the following form,

$$\begin{aligned} P^{-1}B\big(A + (p-1)B\big)^{-1} &= \big(A + pB\big)^{-1}\big(A + (p-1)B\big)\big(A - B\big)^{-1}B\big(A + (p-1)B\big)^{-1} \\ &= \big(A + pB\big)^{-1}B\big(A - B\big)^{-1}, \end{aligned}$$

where we used the fact $\big(A - B\big)^{-1}B\big(A + (p-1)B\big)^{-1} = \big(A + (p-1)B\big)^{-1}B\big(A - B\big)^{-1}$ derived from item 3 in Lemma 16. In a similar fashion, each block of $\mathbf{N}_p^{-1}L_pP^{-1}$ is also equal to $\big(A + (p-1)B\big)^{-1}B\big(A - B\big)^{-1}$. Therefore, we only need to show that the blocks of $\big(\mathbf{N}_p - L_pA^{-1}L_p\big)^{-1}$ are consistent with the remaining of the blocks in $\mathbf{N}_{p+1}^{-1}$. Note that $\mathbf{N}_p - L_pA^{-1}L_p = \mathrm{I} \otimes \big(A - B\big) + \mathbb{1}\mathbb{1}^\top \otimes \big(B - BA^{-1}B\big)$. Hence, for each diagonal term of the

inverse,

$$\begin{aligned}
&\Big(\big(A-BA^{-1}B\big)+(p-1)\big(B-BA^{-1}B\big)\Big)^{-1}\Big(\big(A-BA^{-1}B\big)+(p-2)\big(B-BA^{-1}B\big)\Big)\big(A-B\big)^{-1}\\
&\quad=\Big(\big(I-BA^{-1}\big)\big(I+BA^{-1}\big)A+(p-1)\big(I-BA^{-1}\big)B\Big)^{-1}\times\\
&\qquad\qquad\Big(\big(I-BA^{-1}\big)\big(I+BA^{-1}\big)A+(p-2)\big(I-BA^{-1}\big)B\Big)\big(A-B\big)^{-1}\\
&\quad=\Big(\big(I+BA^{-1}\big)A+(p-1)B\Big)^{-1}\big(I-BA^{-1}\big)^{-1}\times\\
&\qquad\qquad\big(I-BA^{-1}\big)\Big(\big(I+BA^{-1}\big)A+(p-2)B\Big)\big(A-B\big)^{-1}\\
&\quad=\big(A+pB\big)^{-1}\big(A+(p-1)B\big)\big(A-B\big)^{-1},
\end{aligned}$$

and for the off-diagonal terms,

$$\begin{aligned}
&-\Big(\big(A-BA^{-1}B\big)+(p-1)\big(B-BA^{-1}B\big)\Big)^{-1}\big(\mathrm{I}-BA^{-1}\big)B\big(A-B\big)^{-1}\\
&\qquad=-\Big(\big(I-BA^{-1}\big)\big(I+BA^{-1}\big)A+(p-1)\big(I-BA^{-1}\big)B\Big)^{-1}\big(\mathrm{I}-BA^{-1}\big)B\big(A-B\big)^{-1}\\
&\qquad=-\big(A+pB\big)B\big(A-B\big)^{-1}.
\end{aligned}$$

Hence, the inverse can be cast as,

$$\mathbf{N}_{p+1}^{-1}=\mathrm{I}\otimes\big(F_{p+1}+G_{p+1}\big)-\mathbb{1}\mathbb{1}^{\top}G_{p+1},$$

with $F_{p+1}$ and $G_{p+1}$ defined as,

$$\begin{aligned}
F_{p+1}&=\big(A+pB\big)^{-1}\big(A+(p-1)B\big)\big(A-B\big)^{-1}\\
G_{p+1}&=\big(A+pB\big)B\big(A-B\big)^{-1}.
\end{aligned}$$

4) With direct multiplication and using the mixed-product property of Kronecker products,

$$\begin{aligned}
\mathbf{N}_r\mathbf{M}_r&=\Big(\mathrm{I}_r\otimes\big(A-B\big)+\mathbb{1}_r\mathbb{1}_r^{\top}\otimes B\Big)\Big(\mathrm{I}_r\otimes\big(C-D\big)+\mathbb{1}_r\mathbb{1}_r^{\top}\otimes D\Big)\\
&=\mathrm{I}_r\otimes\big(A-B\big)\big(C-D\big)+\mathbb{1}_r\mathbb{1}_r^{\top}\otimes B\big(C-D\big)+\mathbb{1}_r\mathbb{1}_r^{\top}\otimes\big(A-B\big)D+n\mathbb{1}_r\mathbb{1}_r^{\top}\otimes BD\\
&=\mathrm{I}_r\otimes\big(A-B\big)\big(C-D\big)+\mathbb{1}_r\mathbb{1}_r^{\top}\otimes\Big(B\big(C-D\big)+\big(A-B\big)D+nBD\Big).
\end{aligned}$$

□

### *3.4 Convergence and Stability*

In this section, we provide the convergence and stability analyses of the D3PI algorithm. First, we study the structure and gain margins of each local controller and then establish the stability property of the proposed controller for the entire network throughout the learning process of D3PI algorithm. Lastly, we show the convergence of D3PI to a stabilizing suboptimal distributed controller followed by the derivation of a suboptimality bound characterized by the problem parameters.

We begin by introducing a linear group and show its useful properties in our setup. For any integer $r \geq 2$, let us define the *patterned linear group* as follows

$$\mathrm{PL}(r \times n, \mathbb{R}) \coloneqq \Big\{ \mathbf{N}_r \in \mathrm{GL}(rn, \mathbb{R}) \;\Big|\; \mathbf{N}_r = \mathrm{I}_r \otimes \big(A - B\big) + \mathbb{1}_r \mathbb{1}_r^\top \otimes B, \\ \text{for some } A \in \mathrm{GL}(n, \mathbb{R}) \cap \mathbb{S}^{n \times n} \text{ and } B \in \mathbb{S}^{n \times n} \Big\}.$$

**Proposition 2.** *The* $\mathrm{PL}(r \times n, \mathbb{R})$ *is indeed a linear group. Furthermore, for any stable matrix* $A \in \mathrm{PL}(r \times n, \mathbb{R})$*, let* $P$ *denote the unique solution to the discrete-time Lyapunov equation for any* $0 \prec Q \in \mathbb{S}^{rn \times rn}$*, i.e., it satisfies* $P = A^\top P A + Q$*. Then,* $P \in \mathrm{PL}(r \times n, \mathbb{R})$ *if and only if* $Q \in \mathrm{PL}(r \times n, \mathbb{R})$*.*

*Proof.* The fact that $\mathrm{PL}(r \times n, \mathbb{R})$ is a linear group follows directly by item 3 and item 4 in Lemma 17. Now for any Schur stable matrix $A$, and any symmetric positive definite matrix $Q$ there exists a unique positive definite solution $P$ to the discrete Lyapunov equation described by $P = \sum_{j=0}^{\infty} (A^\top)^j Q A^j$ (item 2 in Lemma 16). Note, that $\mathrm{PL}(r \times n, \mathbb{R}) \subset \mathbb{S}^{rn \times rn}$ by construction. Therefore, since $\mathrm{PL}(r \times n, \mathbb{R})$ is a linear group, each summand also falls in $\mathrm{PL}(r \times n, \mathbb{R})$ whenever $Q \in \mathrm{PL}(r \times n, \mathbb{R})$. Thus, as the infinite sum preserves the structure, $P \in \mathrm{PL}(r \times n, \mathbb{R})$ whenever $Q \in \mathrm{PL}(r \times n, \mathbb{R})$. Conversely, if $P \in \mathrm{PL}(r \times n, \mathbb{R})$, then $Q = P - A^\top P A^\top$ also must lie in $\mathrm{PL}(r \times n, \mathbb{R})$. This completes the proof. □

The fact that the patterned linear group is preserved under Lyapunov equation is a key to our analysis that facilitates efficient propagation of information in our algorithm. Next, we demonstrate how a specific structure and stability of the controller for the subgraph $\mathcal{G}_{d,\mathrm{learn}}$, if initialized properly, can be preserved throughout D3PI algorithm.

**Proposition 3.** *Let* $\widetilde{\boldsymbol{K}}_k \coloneqq \mathrm{I}_d \otimes \big(K_k - L_k\big) + \mathbb{1}\mathbb{1}^\top \otimes L_k$ *for all* $k \geq 0$ *where* $K_k$ *and* $L_k$ *are obtained as in Algorithm 4. If* $\widetilde{\boldsymbol{K}}_1$ *is a stabilizing controller for the system in* $\mathcal{G}_{d,\text{learn}} = \mathcal{K}(\mathcal{V}_{\mathcal{G}_d})$*, then the following statements are true for all* $k \geq 0$*,*

- $\text{Policy}_k(\mathcal{V}_{\mathcal{G}_{d,learn}})\big|_t = \widetilde{\boldsymbol{K}}_k \tilde{\mathbf{x}}_t,\ \forall t.$
- $\widetilde{\boldsymbol{K}}_k$ *is stabilizing for all* $k \geq 0$ *for the system in* $\mathcal{G}_{d,\text{learn}}$*.*
- $\Delta K_k \coloneqq K_k - L_k$ *stabilizes the dynamics of one single agent, i.e.,* $A + B\Delta K_k$ *is Schur stable.*

*Proof.* The first claim is a direct consequence of the structure of $\mathcal{G}_{d,\text{learn}}$ during the learning phase, where $\mathcal{G}_{d,\text{learn}} = \mathcal{K}\left(\mathcal{G}_d\right)$ and hence $\boldsymbol{u}_i = K_k \boldsymbol{x}_i + L_k \sum_{j \in \mathcal{V}_{\mathcal{G}_{d,\text{learn}}} \setminus \{i\}} \boldsymbol{x}_j$ for all $i \in \mathcal{V}_{\mathcal{G}_{d,\text{learn}}}$ which, in turn, results in $\tilde{\mathbf{u}}_t = \widetilde{\mathbf{K}}_k \tilde{\mathbf{x}}_t$. Since the underlying system $\big(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}\big)$ is assumed to be controllable and $\tilde{\mathbf{u}}_t$ is persistently excited, the stabilizability of the policy $\widetilde{\mathbf{K}}_k$ follows directly form Theorem 1 in [27]. Now, assume $\widetilde{\mathbf{K}}_\ell$ is stabilizing for $\ell = 1, \cdots, k$. From Lemma 15, the solution to (3.34) is given by,

$$\widetilde{\mathbf{P}}_k = \sum_{i=0}^{\infty} \left( \left( \widetilde{\mathbf{A}} + \widetilde{\mathbf{B}} \widetilde{\mathbf{K}}_k \right)^\top \right)^i \left( \widetilde{\mathbf{Q}} + \widetilde{\mathbf{K}}_k^\top \widetilde{\mathbf{R}} \widetilde{\mathbf{K}}_k \right) \left( \widetilde{\mathbf{A}} + \widetilde{\mathbf{B}} \widetilde{\mathbf{K}}_k \right)^i.$$

From item 4 in Lemma 17, the term inside the latter sum preserves the structure of $\widetilde{\mathbf{K}}_k$. Therefore, $\widetilde{\mathbf{P}}_k$ can be reformulated as,

$$\widetilde{\mathbf{P}}_k = \mathrm{I}_d \otimes \big(P_{1,k} - P_{2,k}\big) + \mathbb{1}\mathbb{1}^\top \otimes P_{2,k}, \tag{3.38}$$

where $P_{1,k}$ and $P_{2,k}$ are obtained from (3.34) as,

$$\begin{aligned} P_{1,k} &= \mathcal{C} A_{K_k} + (d-1)\mathcal{D} B L_k + Q_1 + (d-1)Q_2 + K_k^\top R K_k + (d-1) L_k^\top R L_k, \\ P_{2,k} &= \mathcal{C} B L_k + \mathcal{D}\big(A_{K_k} + (d-2) B L_k\big) - Q_2 + K_k^\top R L_k + L_k^\top R K_k + (d-2) L_k^\top R L_k, \end{aligned} \tag{3.39}$$

with $A_{K_k} = A + BK_k$ and,

$$\mathcal{C} = A_{K_k}^\top P_{1,k} + (d-1) L_k^\top B^\top P_{2,k}, \qquad \mathcal{D} = A_{K_k}^\top P_{2,k} + L_k^\top B^\top \big(P_{1,k} + (d-2) P_{2,k}\big).$$

Setting $\mathcal{Z}_1 = B^\top P_{1,k} A$, $\mathcal{Z}_2 = B^\top P_{2,k} A$, $\mathcal{Y}_1 = R + B^\top P_{1,k} B$, and $\mathcal{Y}_2 = B^\top P_{2,k} B$, the matrix blocks $\widetilde{\mathbf{H}}_{21,k}$ and $\widetilde{\mathbf{H}}_{22,k}$ (as defined in (3.35)) become,

$$\widetilde{\mathbf{H}}_{21,k} = \widetilde{\mathbf{B}}^\top \widetilde{\mathbf{P}}_k \widetilde{\mathbf{A}} = \mathrm{I}_d \otimes \big(\mathcal{Z}_1 - \mathcal{Z}_2\big) + \mathbb{1}\mathbb{1}^\top \otimes \mathcal{Z}_2,$$
$$\widetilde{\mathbf{H}}_{22,k} = \widetilde{\mathbf{R}} + \widetilde{\mathbf{B}}^\top \widetilde{\mathbf{P}}_k \widetilde{\mathbf{B}} = \mathrm{I}_d \otimes \big(\mathcal{Y}_1 - \mathcal{Y}_2\big) + \mathbb{1}\mathbb{1}^\top \otimes \mathcal{Y}_2.$$

Recall from (3.36) the policy update law $\widetilde{\mathbf{K}}_{k+1} = -\widetilde{\mathbf{H}}_{22,k}^{-1} \widetilde{\mathbf{H}}_{21,k}$. Then from Lemma 17 item 3 we know $\widetilde{\mathbf{H}}_{22,k}^{-1} \in \mathrm{PL}(d \times m, \mathbb{R})$ and,

$$\widetilde{\mathbf{K}}_k = \mathrm{I}_d \otimes \big(K_k - L_k\big) + \mathbb{1}\mathbb{1}^\top \otimes L_k, \tag{3.40}$$

where $K_k = \mathcal{F}' \mathcal{Z}_1 + (d-1)\mathcal{G}' \mathcal{Z}_2$ and $L_k = \mathcal{F}' \mathcal{Z}_2 + \mathcal{G}' \mathcal{Z}_1 + (d-2)\mathcal{G}' \mathcal{Z}_2$ with,

$$\mathcal{F}' = \Big(\mathcal{Y}_1 - (d-1)\mathcal{Y}_2 \left(\mathcal{Y}_1 + (d-2)\mathcal{Y}_2\right)^{-1} \mathcal{Y}_2\Big)^{-1}, \quad \mathcal{G}' = \left(\mathcal{Y}_1 + (d-1)\mathcal{Y}_2\right)^{-1} \mathcal{Y}_2 \left(\mathcal{Y}_1 - \mathcal{Y}_2\right)^{-1}.$$

Subtracting the two equations in (3.39) and some simplification results in,

$$\Delta P_k = A_{\Delta K_k}^\top \Delta P_k A_{\Delta K_k} + \widetilde{Q} + \Delta \widetilde{K}_k^\top R \Delta \widetilde{K}_k, \tag{3.41}$$

where $\Delta P_k = P_{1,k} - P_{2,k}$, $A_{\Delta K_k} = A + B\Delta K_k$, $\Delta K_k = K_k - L_k$, and $\widetilde{Q} = Q_1 + dQ_2$. Since $\widetilde{Q} + \Delta \widetilde{K}_k^\top R \Delta \widetilde{K}_k \succ 0$, by Lyapunov Stability Criterion it suffices to show that $\Delta P_k \succ 0$. Note that $\widetilde{\mathbf{K}}_k$ being stabilizing implies $\widetilde{\mathbf{P}}_k \succ 0$ from (3.34). Therefore, the final claim follows by item 3 in Lemma 17 and the structure of $\widetilde{\mathbf{P}}_k$ from (3.38). □

The results in Proposition 3 form the cornerstone of the remainder of the analysis in this section. It also gives insights on the existence of a stabilizing controller $\Delta K_k$ and its corresponding cost-to-go matrix $\Delta P_k$. In the sequel, our goal is to design a distributed suboptimal controller based on the components that shape $\Delta K_k$. To this end, we find stabilizability gain margins that induce more flexibility in designing the distributed controller. In the following proposition, we provide the backbone of the distributed controller design for the networked system of $\mathcal{G}$.

**Proposition 4.** *At each iteration $k \geq 0$, let $K_k$, $L_k$ and $\tau_k$ be as obtained in Algorithm 4 and choose $\alpha$ such that $|\alpha - 1| < \tau_k$. Then $A + B(K_k - \alpha L_k)$ is Schur stable.*

*Proof.* Define the Lyapunov candidate function $V_k(\boldsymbol{x}_t) = \boldsymbol{x}_t^\top \Delta P_k \boldsymbol{x}_t$ where $\Delta P_k$ is defined as in Proposition 3 and $\boldsymbol{x}_t$ contains the states of the closed-loop system $\boldsymbol{x}_{t+1} = \big(A + B(K_k - \alpha L_k)\big)\boldsymbol{x}_t$. We show that for the given choice of $\alpha$, $V_k$ is nonincreasing. From definition,

$$\Delta V_k(\boldsymbol{x}_t) \coloneqq V_k(\boldsymbol{x}_{t+1}) - V_k(\boldsymbol{x}_t) = \boldsymbol{x}_t^\top \Gamma_k \boldsymbol{x}_t,$$

where $\Gamma_k = (A + B(K_k - \alpha L_k))^\top \Delta P_k (A + B(K_k - \alpha L_k)) - \Delta P_k$. Suppose $\alpha < 1$. from (3.41),

$$\begin{aligned}
\Gamma_k &= A_{\Delta K_k}^\top \Delta P_k A_{\Delta K_k} + (1-\alpha)\left(A_{\Delta K_k}^\top \Delta P_k B L_k + L_k^\top B^\top \Delta P_k A_{\Delta K_k}\right) + (1-\alpha^2) L_k^\top B^\top \Delta P_k B L_k - \Delta P_k \\
&= (1-\alpha)\left(A_{\Delta K_k}^\top \Delta P_k B L_k + L_k^\top B^\top \Delta P_k A_{\Delta K_k}\right) + (1-\alpha^2) L_k^\top B^\top \Delta P_k B L_k - \widetilde{Q} - \Delta K_k^\top R \Delta K_k \\
&\leq (1-\alpha)\left(a A_{\Delta K_k}^\top \Delta P_k A_{\Delta K_k} + (1/a) L_k^\top B^\top \Delta P_k B L_k\right) + (1-\alpha^2) L_k^\top B^\top \Delta P_k B L_k - \widetilde{Q} - \Delta K_k^\top R \Delta K_k \\
&= (1-\alpha)\left(a A_{\Delta K_k}^\top \Delta P_k A_{\Delta K_k} + (1/a + 1 - \alpha) L_k^\top B^\top \Delta P_k B L_k\right) - \big(\widetilde{Q} + \Delta K_k^\top R \Delta K_k\big),
\end{aligned}$$

where the inequality holds for some $a > 0$ due to Lemma 15. Let $\beta = (1-\alpha)/2$ and choose $a = \beta + \sqrt{\beta^2 + 1}$. Then,

$$\begin{aligned}
\Gamma_k &= (1-\alpha) a \Big(A_{\Delta K_k}^\top \Delta P_k A_{\Delta K_k} + L_k^\top B^\top \Delta P_k B L_k\Big) - \Big(\widetilde{Q} + \Delta K_k^\top R \Delta K_k\Big) \\
&\leq (1-\alpha) a \sigma_{\max}\big(A_{\Delta K_k}^\top \Delta P_k A_{\Delta K_k} + L_k^\top B^\top \Delta P_k B L_k\big) - \sigma_{\min}\big(\widetilde{Q} + \Delta K_k^\top R \Delta K_k\big)
\end{aligned}$$

Note that using the parameterization of control blocks in Algorithm 4 and also from (3.35), the latter bound can be obtained completely from data as,

$$\begin{aligned}
\Gamma_k &\leq (1-\alpha) a \sigma_{\max}\big(\Xi_k + L_k^\top (\Delta Y - R) L_k\big) - \sigma_{\min}\Big(\widetilde{Q} + \Delta K_k^\top R \Delta K_k\Big) \\
&= \big((1-\alpha)a - \gamma_k\big)\, \sigma_{\max}\Big(\Xi_k + L_k^\top (\Delta Y - R) L_k\Big)
\end{aligned}$$

with $\Xi_k$ defined as,

$$\Xi_k = \Delta X - \widetilde{Q} + Q_2 + \Delta K_k^\top \Delta Z + \Delta Z^\top \Delta K_k + \Delta K_k^\top \big(\Delta Y - R\big) \Delta K_k.$$

where we used the structure in (3.35) and the way the matrix blocks were defined on line 13 of Algorithm 4. From the hypothesis $1 - \tau_k < \alpha$, we obtain $\gamma_k^2 - 2\beta^2 \gamma_k - 2\beta^2 > 0$. Since $\gamma_k > 0$, this second-order term in $\gamma_k$ only gets positive when,

$$\gamma_k > 2\beta^2 + 2\beta\sqrt{\beta^2 + 1} = 2\beta(\beta + \sqrt{\beta^2 + 1}) = (1-\alpha)a.$$

Therefore, $\Delta V_k(\boldsymbol{x}_t) < 0$ for $1 - \tau_k < \alpha < 1$. Similar reasoning for $1 < \alpha$ gives $\Delta V_k(\boldsymbol{x}_t) < 0$ for $1 < \alpha < 1 + \tau_k$ which completes the proof. □

The result in Proposition 4 provides model-free stability gain margins, $\tau_k$, at each iteration of the algorithm for the dynamics of one single agent in $\mathcal{G}$. We take advantage of these margins as a tool to find stability guarantee for the controller proposed in our algorithm. For the learning phase of D3PI, this is formally established in the following theorem.

**Theorem 18.** *Suppose $K_k$, $L_k$ and $\tau_k$ are defined as in Algorithm 4. Then the control policy* $\text{Policy}_k(\mathcal{V}_\mathcal{G})$ *designed as,*

$$\boldsymbol{u}_i = \Big(K_k - L_k\Big)\boldsymbol{x}_i + L_k \left[\mathbb{I}_{\left\{i\in\mathcal{V}_{\mathcal{G}\setminus\mathcal{G}_{d,\text{learn}}}\right\}} \sum_{j\in\mathcal{N}_i} \frac{\tau_k}{d-1}\boldsymbol{x}_j + \mathbb{I}_{\left\{i\in\mathcal{V}_{\mathcal{G}_{d,\text{learn}}}\right\}} \sum_{j\in\mathcal{V}_{\mathcal{G}_{d,\text{learn}}}\setminus\{i\}} \boldsymbol{x}_j\right], \tag{3.42}$$

*stabilizes the entire network $\mathcal{G}$ at each iteration of the learning phase in Algorithm 4 for any choice of $\mathcal{V}_{\mathcal{G}_d}$.*

*Proof.* From Definition 5 and Proposition 3, the feedback in (3.42) can be cast in compact form as,

$$\text{Policy}_k(\mathcal{G}) = \begin{bmatrix}\text{Policy}_k(\mathcal{G}\setminus\mathcal{G}_{d,\text{learn}})\\ \text{Policy}_k(\mathcal{G}_{d,\text{learn}})\end{bmatrix} = \left[\begin{array}{c|c}\hat{\mathbf{K}}_{\mathcal{G}\setminus\mathcal{G}_{d,\text{learn}}} & \\ \hline \mathbf{0} & \widetilde{\mathbf{K}}_k\end{array}\right] \begin{bmatrix}\text{State}_k(\mathcal{G}\setminus\mathcal{G}_{d,\text{learn}})\\ \text{State}_k(\mathcal{G}_{d,\text{learn}})\end{bmatrix} = \hat{\mathbf{K}}_k\text{State}_k(\mathcal{G}),$$

where,

$$\begin{aligned}\hat{\mathbf{K}}_{\mathcal{G}\setminus\mathcal{G}_{d,\text{learn}}} &= \mathrm{I}_{N-d}\otimes(K_k - L_k) - \frac{\tau_k}{d-1}\mathcal{A}\left(\mathcal{G}\setminus\mathcal{G}_{d,\text{learn}}\right)\otimes L_k,\\ \widetilde{\mathbf{K}}_k &= \mathrm{I}_d\otimes(K_k - L_k) + \mathbb{1}_d\mathbb{1}_d^\top\otimes L_k,\end{aligned}$$

where $\mathcal{A}$ denotes the adjacency matrix and the structure of $\hat{\mathbf{K}}_k$ emanates from the fact that the information exchange is unidirectional during the learning phase. Now consider the closed-loop system of $\mathcal{G}$,

$$\hat{\mathbf{A}}_\mathcal{G}\Big|_{cl} = \hat{\mathbf{A}} + \hat{\mathbf{B}}\hat{\mathbf{K}}_k = \left[\begin{array}{c|c}\big(\mathrm{I}_{N-d}\otimes A\big) + \big(\mathrm{I}_{N-d}\otimes B\big)\hat{\mathbf{K}}_{\mathcal{G}\setminus\mathcal{G}_d} & \\ \hline \mathbf{0} & \widetilde{\mathbf{A}}_{\widetilde{\mathbf{K}}_k}\end{array}\right],$$

where $\widetilde{\mathbf{A}}_{\widetilde{\mathbf{K}}_k} = \widetilde{\mathbf{A}} + \widetilde{\mathbf{B}}\widetilde{\mathbf{K}}_k$ is the closed-loop system of $\mathcal{G}_d$. Note that,

$$\hat{\mathbf{K}}_{\mathcal{G}\setminus\mathcal{G}_d} = \mathrm{I}_{N-d} \otimes K_k - \Big(\mathrm{I}_{N-d} + \frac{\tau_k}{d-1}\mathcal{A}(\mathcal{G})\Big) \otimes L_k \ .$$

Define $S = \mathrm{I}_{N-d} + \frac{\tau_k}{d-1}\mathcal{A}(\mathcal{G})$ and let $J$ be the Jordan block of $S$ with the similarity transformation $\mathcal{T}S\mathcal{T}^{-1} = J$ where $\mathcal{T} \in \mathbb{R}^{(N-d)\times(N-d)}$ is invertible. Therefore,

$$\begin{bmatrix} \mathcal{T}\otimes \mathrm{I}_{N-d} & \mathbf{0} \\ \mathbf{0} & \mathrm{I}_d \end{bmatrix} \Big( \hat{\mathbf{A}}_{\mathcal{G}}\Big|_{cl} \Big) \begin{bmatrix} (\mathcal{T}\otimes \mathrm{I}_{N-d})^{-1} & \mathbf{0} \\ \mathbf{0} & \mathrm{I}_d \end{bmatrix} = \left[\begin{array}{c|c} \mathrm{I}_{N-d}\otimes\big(A+BK_k\big) - J\otimes BL_k & \\ \hline \mathbf{0} & \widetilde{\mathbf{A}}_{\widetilde{\mathbf{K}}_k} \end{array}\right],$$

where the upper block forms an upper triangular matrix with block diagonals $A + B(K_k - \lambda_i(S)L_k)$. We already know from Proposition 3 that $\widetilde{\mathbf{A}}_{\widetilde{\mathbf{K}}_k}$ is Schur stable. Hence, we only need to show $\rho\left(A + B(K - \lambda_i(S)L)\right) < 1$ for $i = 1, \cdots, N-d$. We know $|\lambda_i(\mathcal{A}(\mathcal{G}))| \leq d_{\max}$ [23]. Then from this inequality, definition of $S$, and the fact that $d_{\max} = d - 1$,

$$|\lambda_i(S) - 1| \leq \tau_k.$$

The rest of the proof follows directly from Proposition 4. □

The latter theorem establishes the fact that the proposed feedback mechanism stabilizes the entire network throughout the learning phase. This facilitates the control of the agents outside of $\mathcal{G}_d$, while the learning process is activated in the subgraph. Nonetheless, the practicality and suboptimality characteristics of the algorithm heavily depend on its convergence. This will be addressed in the following theorem.

**Theorem 19.** *Assume that the initial controller $K_1$ is stabilizing for (3.1) and $\mathrm{Policy}_k(\mathcal{V}_{\mathcal{G}_d})\big|_t$ in Algorithm 5 is persistently excited. Then $K_k \to K^*$, $L_k \to L^*$, $\gamma_k \to \gamma^*$, and $\tau_k \to \tau^*$ as $k \to \infty$ where $\widetilde{\boldsymbol{K}}^* = \mathrm{I}_d \otimes \big(K^* - L^*\big) + \mathbb{1}\mathbb{1}^\top \otimes L^*$ is the optimal solution to the infinite-horizon state-feedback LQR problem with system parameters $\big(\widetilde{\boldsymbol{A}}, \widetilde{\boldsymbol{B}}, \widetilde{\boldsymbol{Q}}, \widetilde{\boldsymbol{R}}\big)$ defined as $\widetilde{\boldsymbol{A}} = \mathrm{I}_d \otimes A$, $\widetilde{\boldsymbol{B}} = \mathrm{I}_d \otimes B$, $\widetilde{\boldsymbol{Q}} = \mathrm{I}_d \otimes \widetilde{Q} - \mathbb{1}\mathbb{1}^\top \otimes Q_2$, and $\widetilde{\boldsymbol{R}} = \mathrm{I}_d \otimes R$.*

*Proof.* We first show $\boldsymbol{\theta}_k \to \boldsymbol{\theta}_k^*$ as $k \to \infty$ where $\boldsymbol{\theta}_k^*$ includes similar parameterization to that of $\boldsymbol{\theta}_k$ at iteration $k$ but for the true system parameters. Correspondingly, let $\widetilde{\mathbf{H}}^*$ be the counterpart to $\widetilde{\mathbf{H}}_k$. First, by induction we show that the parameter estimation error is

uniformly bounded. Then, as the established bound will be inversely proportional to the number of iterations of SPE, we conclude the convergence of the parameter estimator. For that, assume the parameter estimation error in Algorithm 4 is bounded,

$$\left\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_i^*\right\| \leq c_i \leq c_0,$$

for $i = 1, \cdots, p-1$ and $c_0 > 0$. From Lemma 2 in [27],

$$\left\|\boldsymbol{\theta}_p - \boldsymbol{\theta}_p^*\right\| \leq \epsilon\Big(\left\|\boldsymbol{\theta}_{p-1} - \boldsymbol{\theta}_{p-1}^*\right\| + \left\|\boldsymbol{\theta}_{p-1}^* - \boldsymbol{\theta}_p^*\right\|\Big),$$

where $\epsilon = 1/\big(C\epsilon_0\sigma_{\min}(\mathcal{P}_0)\big)$ for some $\epsilon_0 > 0$ and $C$ is the number of iterations of SPE for each $k$. From Theorem 1 in [83] we know $\left\|\boldsymbol{\theta}_{p-1}^* - \boldsymbol{\theta}_p^*\right\| \leq s_0$ at every consecutive step of policy iteration under controllability assumptions. Therefore,

$$\left\|\boldsymbol{\theta}_p - \boldsymbol{\theta}_p^*\right\| \leq \frac{c_0 + s_0}{C\epsilon_0\sigma_{\min}(\mathcal{P}_0)} = \frac{\kappa}{C},$$

where $\kappa = (c_0 + s_0) \,/\, (\epsilon_0\sigma_{\min}(\mathcal{P}_0)) > 0$. Hence by choosing $C > \kappa/\delta$ for any $\delta > 0$, we get $\left\|\boldsymbol{\theta}_p - \boldsymbol{\theta}_p^*\right\| < \delta$. We know from [83] that at convergence, the approximated policy for true system parameters can be derived from,

$$\widetilde{\mathbf{K}}^* = -\widetilde{\mathbf{H}}_{22}^{*-1}\widetilde{\mathbf{H}}_{21}^* = -\Big(\widetilde{\mathbf{R}} + \widetilde{\mathbf{B}}^\top\widetilde{\mathbf{P}}^*\widetilde{\mathbf{B}}\Big)^{-1}\widetilde{\mathbf{B}}^\top\widetilde{\mathbf{P}}^*\widetilde{\mathbf{A}}, \tag{3.43}$$

where $\widetilde{\mathbf{A}}$, $\widetilde{\mathbf{B}}$, $\widetilde{\mathbf{R}}$, and $\widetilde{\mathbf{Q}}$ are defined as in (3.34) and $\widetilde{\mathbf{P}}^*$ satisfies,

$$\widetilde{\mathbf{P}}^* = \Big(\widetilde{\mathbf{A}} + \widetilde{\mathbf{B}}\widetilde{\mathbf{K}}^*\Big)^\top\widetilde{\mathbf{P}}^*\Big(\widetilde{\mathbf{A}} + \widetilde{\mathbf{B}}\widetilde{\mathbf{K}}^*\Big) + \widetilde{\mathbf{Q}} + \widetilde{\mathbf{K}}^{*\top}\widetilde{\mathbf{R}}\widetilde{\mathbf{K}}^*.$$

Recall from (3.38) that the solution to this Lyapunov equation satisfies $\widetilde{\mathbf{P}}^* = \mathrm{I}_d \otimes (P_1^* - P_2^*) + \mathbb{1}\mathbb{1}^\top \otimes P_2^*$ with corresponding blocks $P_1^*$ and $P_2^*$. From Lemmas 16 and 17,

$$\widetilde{\mathbf{H}}_{21} = \mathrm{I}_d \otimes (F_1 - G_1) + \mathbb{1}\mathbb{1}^\top \otimes G_1, \qquad \widetilde{\mathbf{H}}_{22}^{-1} = \mathrm{I}_d \otimes (F_2 - G_2) + \mathbb{1}\mathbb{1}^\top \otimes G_2,$$

where,

$$F_1 = B^\top P_1^* A, \qquad G_1 = B^\top P_2^* A,$$
$$F_2 = \Big((R + B^\top P_1^* B) - (d-1)(B^\top P_2^* B)\big(R + B^\top P_1^* B + (d-2)(B^\top P_2^* B)\big)^{-1}(B^\top P_2^* B)\Big)^{-1},$$
$$G_2 = \Big((R + B^\top P_1^* B) + (d-1)(B^\top P_2^* B)\Big)^{-1}(B^\top P_2^* B)\Big(R + B^\top P_1^* B - B^\top P_2^* B\Big)^{-1}.$$

Then direct multiplication results in $\widetilde{\mathbf{K}}^* = \mathrm{I}_d \otimes (\mathcal{K}_1^* - \mathcal{K}_2^*) + \mathbb{1}\mathbb{1}^\top \otimes \mathcal{K}_2^*$ where,

$$\mathcal{K}_1^* = F_2F_1 + (d-1)G_2G_1, \qquad \mathcal{K}_2^* = F_2G_1 + G_2F_1 + (d-2)G_2G_1,$$

which coincides with the updates of $K_k$ and $L_k$ in the algorithm. Therefore, our policy update at optimality corresponds to (3.43) which completes the proof. □

As we remove the temporary links that were provided for the learning phase, the structure of the interconnections in $\mathcal{G}$ are returned to it original topology. Therefore, it is vital to provide stability guarantees after the recursion in Algorithm 4 is completed. This statement is a direct consequence of Theorem 18 when the control components are converged and is outlined in the following corollary.

**Corollary 1.** *Suppose $K^*$, $L^*$, $\gamma^*$, and $\tau^*$ are given as in Theorem 19 where Algorithm 4 is converged. Then* Policy($\mathcal{V}_\mathcal{G}$) *defined as,*

$$\boldsymbol{u}_i = \big(K^* - L^*\big)\boldsymbol{x}_i + \frac{\tau_k}{d-1} L^* \sum_{j \in \mathcal{N}_i} \boldsymbol{x}_j, \qquad \forall i \in \mathcal{V}_\mathcal{G}$$

*stabilizes the system in* (3.2).

Finally, we conclude this section by exploring the suboptimality of our proposed policy. Given the problem parameters, let $\hat{\mathbf{K}}^*_{\text{struc}}$ denote the globally optimal distributed solution for the structured LQR problem in (3.5) with the associated cost matrix $\hat{\mathbf{P}}^*_{\text{struc}}$. Given any other stabilizing distributed policy $\hat{\mathbf{K}}$ associated with cost matrix $\hat{\mathbf{P}}$, we define the *optimality gap* as,

$$\text{gap}(\hat{\mathbf{K}}) \coloneqq \text{Tr}[\hat{\mathbf{P}} - \hat{\mathbf{P}}^*_{\text{struc}}].$$

The following theorem provides an upperbound on the optimality gap of distributed policy learned by D3PI based on the problem parameters. Especially, when the system is "contractible", the upperbound depends on the difference of the distributed controller with that of unstructured optimal LQR controller.

**Theorem 20.** *Let $\hat{\boldsymbol{K}}^*$ be the distributed policy learned by Algorithm 4 at convergence corresponding to the cost matrix $\hat{\boldsymbol{P}}^*$. Also, let $\hat{\boldsymbol{K}}^*_{\text{lqr}}$ denote the optimal (unstructured) solution*

*to the infinite-horizon state-feedback LQR problem with parameters* $(\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}, \hat{\boldsymbol{Q}}, \hat{\boldsymbol{R}})$ *associated with the cost matrix* $\hat{\boldsymbol{P}}^*_{\text{lqr}}$*. Then the optimality gap of* $\hat{\boldsymbol{K}}^*$ *is bounded as follows:*

$$0 \leq \text{gap}(\hat{\boldsymbol{K}}^*) \leq \text{Tr}(\hat{\boldsymbol{P}}^* - \hat{\boldsymbol{P}}^*_{\text{lqr}}),$$

*Furthermore, if* $\hat{\boldsymbol{A}}_{\hat{\boldsymbol{K}}_{\text{lqr}}} = \hat{\boldsymbol{A}} + \hat{\boldsymbol{B}}\hat{\boldsymbol{K}}_{\text{lqr}}$ *is contractible (i.e.,* $\sigma_{\max}(\hat{\boldsymbol{A}}_{\hat{\boldsymbol{K}}_{\text{lqr}}}) < 1$*), then*

$$0 \leq \text{gap}(\hat{\boldsymbol{K}}^*) \leq \frac{\text{Tr}(\boldsymbol{M})}{1 - \sigma^2_{\max}(\hat{\boldsymbol{A}}_{\hat{\boldsymbol{K}}_{\text{lqr}}})},$$

*where,* $\boldsymbol{M} \coloneqq (\hat{\boldsymbol{R}} + \hat{\boldsymbol{B}}^\top \hat{\boldsymbol{P}}^* \hat{\boldsymbol{B}})(\hat{\boldsymbol{K}}^* \hat{\boldsymbol{K}}^{*\top} - \hat{\boldsymbol{K}}_{\text{lqr}} \hat{\boldsymbol{K}}_{\text{lqr}}^\top) + 2\hat{\boldsymbol{A}}^\top \hat{\boldsymbol{P}}^* \hat{\boldsymbol{B}}(\hat{\boldsymbol{K}}^* - \hat{\boldsymbol{K}}_{\text{lqr}})$.

*Proof.* Recall that the LQR problem can be cast as the minimization of $f_\Sigma(K) = \text{Tr}\left[P_K \Sigma\right]$ for some $\Sigma \succ 0$ [99, 111] where, w.l.o.g., we set $\Sigma = \text{I}$. From [75] we know that the optimal cost matrix satisfies,

$$\hat{\mathbf{P}}^*_{\text{lqr}} = \mathbf{P}_{\hat{\mathbf{K}}^*_{\text{lqr}}} = \mathbf{A}^\top_{\hat{\mathbf{K}}^*_{\text{lqr}}} \hat{\mathbf{P}}^*_{\text{lqr}} \mathbf{A}_{\hat{\mathbf{K}}^*_{\text{lqr}}} + \hat{\mathbf{K}}^{*\top}_{\text{lqr}} \hat{\mathbf{R}} \hat{\mathbf{K}}^*_{\text{lqr}} + \hat{\mathbf{Q}}, \tag{3.44}$$

where $\mathbf{A}_{\hat{\mathbf{K}}^*_{\text{lqr}}} = \hat{\mathbf{A}} + \hat{\mathbf{B}}\hat{\mathbf{K}}^*_{\text{lqr}}$ and $\hat{\mathbf{K}}^*_{\text{lqr}} = \arg\min_K f_{\text{I}}(K)$. Moreover, $\hat{\mathbf{K}}^*_{\text{struc}} = \arg\min_{K \in \mathcal{U}^N_{m,n}(\mathcal{G})} f_{\text{I}}(K)$ which is associated with the cost matrix $\hat{\mathbf{P}}^*_{\text{struc}} = \mathbf{P}_{\hat{\mathbf{K}}^*_{\text{struc}}}$ that satisfy,

$$\hat{\mathbf{P}}^* = \mathbf{P}_{\hat{\mathbf{K}}^*} = \mathbf{A}^\top_{\hat{\mathbf{K}}^*} \hat{\mathbf{P}}^* \mathbf{A}_{\hat{\mathbf{K}}^*} + \hat{\mathbf{K}}^{*\top} \hat{\mathbf{R}} \hat{\mathbf{K}}^* + \hat{\mathbf{Q}}, \tag{3.45}$$

where $\mathbf{A}_{\hat{\mathbf{K}}^*} = \hat{\mathbf{A}} + \hat{\mathbf{B}}\hat{\mathbf{K}}^*$ and $\hat{\mathbf{K}}^* \in \mathcal{U}^N_{m,n}(\mathcal{G})$. Therefore,

$$\text{Tr}\left[\hat{\mathbf{P}}^*_{\text{lqr}}\right] = f\left(\hat{\mathbf{K}}^*_{\text{lqr}}\right) \leq f\left(\hat{\mathbf{K}}^*_{\text{struc}}\right) = \text{Tr}\left[\hat{\mathbf{P}}^*_{\text{struc}}\right] \leq f\left(\hat{\mathbf{K}}^*\right) = \text{Tr}\left[\hat{\mathbf{P}}^*\right],$$

where the last inequality above follows by the fact that $\hat{\mathbf{K}}^*$ is a feasible solution to the structured problem by construction, *i.e.*, $\hat{\mathbf{K}}^* \in \mathcal{U}^N_{m,n}(\mathcal{G})$. Therefore, $0 \leq \text{gap}(\hat{\mathbf{K}}^*) \leq \text{Tr}\left[\hat{\mathbf{P}}^* - \hat{\mathbf{P}}^*_{\text{lqr}}\right]$. For the second part of the proof, one can obtain from (3.44), (3.45), and some algebraic manipulation,

$$\hat{\mathbf{P}}^* - \hat{\mathbf{P}}^*_{\text{lqr}} = \mathbf{A}^\top_{\hat{\mathbf{K}}^*_{\text{lqr}}} \left(\hat{\mathbf{P}}^* - \hat{\mathbf{P}}^*_{\text{lqr}}\right) \mathbf{A}_{\hat{\mathbf{K}}^*_{\text{lqr}}} + \mathbf{M}'.$$

where,

$$\mathbf{M}' = \mathbf{A}^\top_{\hat{\mathbf{K}}^*} \hat{\mathbf{P}}^* \mathbf{A}_{\hat{\mathbf{K}}^*} - \mathbf{A}^\top_{\hat{\mathbf{K}}^*_{\text{lqr}}} \hat{\mathbf{P}}^* \mathbf{A}_{\hat{\mathbf{K}}^*_{\text{lqr}}} + \hat{\mathbf{K}}^{*\top} \hat{\mathbf{R}} \hat{\mathbf{K}}^* - \hat{\mathbf{K}}^\top_{\text{lqr}} \hat{\mathbf{R}} \hat{\mathbf{K}}_{\text{lqr}}.$$

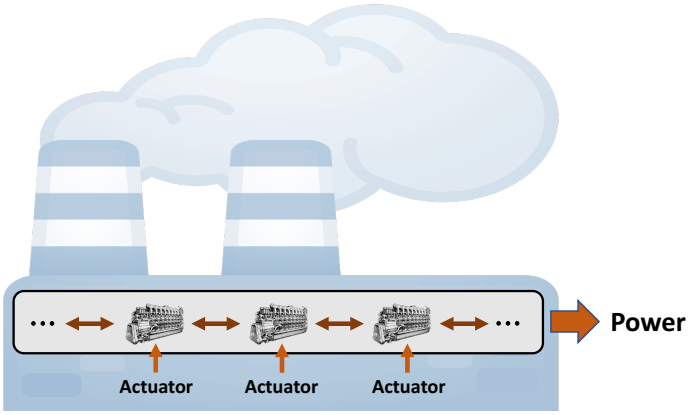



Figure 3.4: Distributed control of multi-engine power generation in an industrial setting.

Since $\hat{\mathbf{P}}^* - \hat{\mathbf{P}}^*_{\text{lqr}} \succ 0$ and $\hat{\mathbf{A}}_{\hat{\mathbf{K}}_{\text{lqr}}}$ is contractible by the hypothesis, from the first part and Theorem 1 in [119] we get that,

$$\text{gap}(\hat{\mathbf{K}}^*) \leq \frac{\text{Tr}(\mathbf{M}')}{1 - \sigma^2_{\max}(\hat{\mathbf{A}}_{\hat{\mathbf{K}}_{\text{lqr}}})} = \frac{\text{Tr}(\mathbf{M})}{1 - \sigma^2_{\max}(\hat{\mathbf{A}}_{\hat{\mathbf{K}}_{\text{lqr}}})},$$

where the last equality follows by cyclic permutation property of trace and definition of $\mathbf{A}_{\hat{\mathbf{K}}^*}$. □

**Remark 8.** *Contractibility of the pair $(A, B)$ entails more restriction than stabilizability or regularizability of the system [156] and is more recently employed in iterative data-guided control methods [2, 96] to streamline finite-time analysis for dynamical systems.*

### *3.4.1 Simulation*

In this section, we apply our method to a series of identical turbocharged diesel engines with exhaust gas recirculation in a chain formation that work collectively for industrial power generation (Figure 3.4). The dynamics of the engines are assumed to be unknown and we apply D3PI to find a model-free distributed policy based on the observed data from the

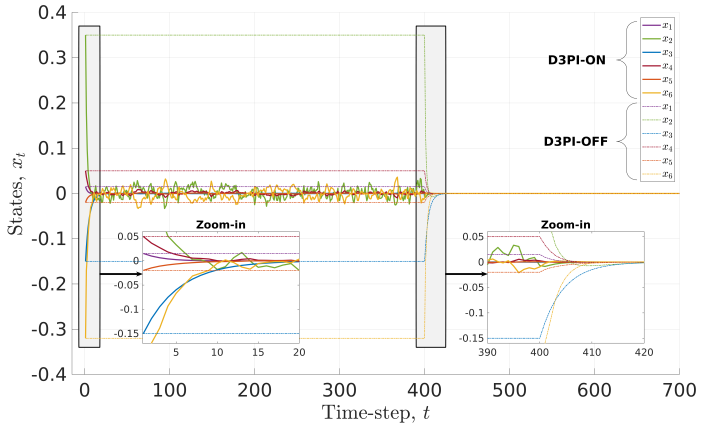



Figure 3.5: State trajectories of engine $i = 4$ during and after the learning process. Solid lines denote results from D3PI, whereas dashed lines refer to stationary $\mathcal{G} \setminus \mathcal{G}_d$ during the learning phase.

system. The values of the continuous-time system parameters are given as [92],

$$A = \begin{bmatrix} -0.4125 & -0.0248 & 0.0741 & 0.0089 & 0.0000 & 0.0000 \\ 101.5873 & -7.2651 & 2.7608 & 2.8068 & 0.0000 & 0.0000 \\ 0.0704 & 0.0085 & -0.0741 & -0.0089 & 0.0000 & 0.0200 \\ 0.0878 & 0.2672 & 0.0000 & -0.3674 & 0.0044 & 0.3962 \\ -1.8414 & 0.0990 & 0.0000 & 0.0000 & -0.0343 & -0.0330 \\ 0.0000 & 0.0000 & 0.0000 & -359.0000 & 187.5364 & -87.0316 \end{bmatrix}, \quad B = \begin{bmatrix} -0.0003 & 0.0005 \\ -0.0764 & 0.1149 \\ 0.0004 & 0.0000 \\ -0.0127 & 0.0016 \\ -0.0005 & -0.0011 \\ 0.0456 & -0.0075 \end{bmatrix}.$$

We normalize and discretized the dynamics with a sampling rate of $\Delta T = 0.1s$. We set $Q_1 = Q_2 = \mathrm{I}_n$ and $R = \mathrm{I}_m$ with $n = 6$ and $m = 2$ and sample a random exploration signal from a zero-mean normal distribution. Given the number of engines $N \geq 5$, the maximum degree of a path graph is $d_{\max} = 2$, and hence, $d = d_{\max} + 1 = 3$. W.l.o.g., we take the first three engines ($i = 1, 2, 3$) as the elements of $\mathcal{G}_d$.

Figure 3.5 demonstrates the trajectories of the states of node $i = 4$ for the case when D3PI synthesis is ON (solid curves) and contrasts it to the naive application of model-free policy iteration to $\mathcal{G}_d$ without considering the control of the rest of the network. The latter is referred to as D3PI-OFF in the plots (dashed curves). As the plot suggests, when D3PI is

activated, the states of each engine remain near the origin throughout learning. Note that when D3PI is ON, the noisy behavior of the trajectories is due to the (unidirectional) interconnections of $\mathcal{G}_d$—being persistently excited—with the rest of the network. Furthermore, the first few iterates of the learning and post-learning are magnified for better comparison of the trajectories. Figures 3.6 and 3.7 outline the performance of our algorithm on the nodes inside $\mathcal{G} \setminus \mathcal{G}_d$ over 5000 runs of the simulation. Figure 3.6 depicts the cumulative logarithmic cost $\hat{\mathbf{J}}_{\mathcal{G}\setminus\mathcal{G}_{d,learn}}$ for all nodes $i \in \mathcal{V}_{\mathcal{G}_d}$ over the entire horizon of the algorithm's implementation for $N = 10$. As the figure shows, the algorithm is converged by the end of the first iteration at time-step $t = 400$ where the cost indices get fixed, implying the convergence of the states to the origin (because the state cost matrix is positive definite). The result of our algorithm (blue) is compared with the case when D3PI is OFF (red) and also the optimal unstructured solution of LQR (for the same system parameters) as the baseline. The plot also verifies the stability of the proposed policy as the cost remains bounded bounded (and $\hat{\mathbf{Q}} \succeq 0$). Similar implementation is performed for $N = 5, 6, \cdots, 30$ and the final costs are reported in Figure 3.7. The increase in costs is due to the added value by the newly included agents in $\mathcal{G} \setminus \mathcal{G}_{d,learn}$. As the plot suggests, the gap between our method and the case with inactive D3PI shows the superiority of our algorithm for any problem setting with different number of agents in the system.

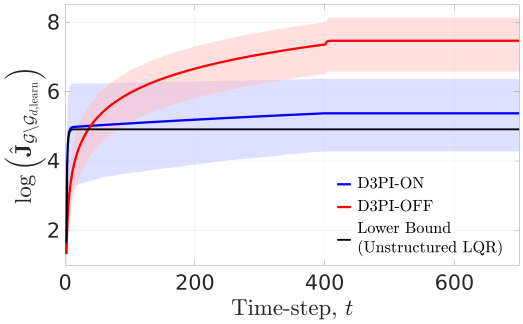



Figure 3.6: Cumulative cost vs time plot.

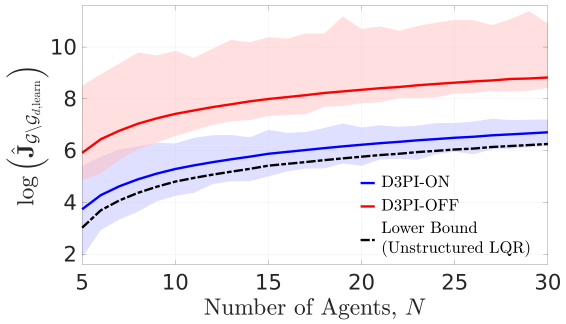



Figure 3.7: Cost vs number of agents plot.

# Chapter 4

# ITERATIVE DATA-DRIVEN CONTROL

In the previous chapter, we introduced distributed control strategies that were completely obtained from data. However, in those cases we use RLS to estimate the policy and this could be costly in that RLS-based method often incur high sample complexity and are difficult to tune. In this chapter, we shift the attention towards non-asymptotic control and analysis of linear systems when collecting data points are not cheap. Hence, a (stabilizing) policy is required to be obtained in a time-sensitive manner. In that regards, we ignore the LQR in this section and instead consider other optimization problems that provide such controller under the new problem constraints. While the methods in this part are derived for a more general setup, in each subsection we consider special cases where our method answers relevant questions to networked systems. Furthermore, as we will discuss in the second part, our method can serve as a remedy to the impractical assumption of initializing from stabilizing controllers that is often used in almost every work in the literature on iterative data-driven control.

## *4.1 Preliminaries*

Feedback control is ubiquitous in modern technology including applications where it provides means of stabilization in addition to performance. Control of open-loop unstable plants arising for instance, in industrial and flight control applications, underscores the importance of stabilization with robustness guarantees. As such, control of unstable systems is an important ongoing research topic, particularly in the context of safety-critical systems. It is well-known that unstable systems are fundamentally more difficult to control [150]; in fact, practical closed-loop systems with unstable subsystems are only locally stable [148]. Yet, most of the existing synthesis literature has focused on model-based control where the designer has to discern fundamental limitations stemming from process instabilities [147].

Recent interest in model-free stabilization in the meantime, has been motivated by modern sensing technologies, robust machine learning approaches and efficient computational methods to reason about control and estimation of uncertain systems- all from measured (online) data [86, 142]. Safety-critical concerns have in fact necessitated non-asymptotic analysis on data-driven methods; this is currently an active multi-disciplinary area of research [51, 63]. In particular, there has been a growing interest in finite-time control and analysis of unknown linear dynamical systems from time-series or a single trajectory [6, 22, 64, 67, 126, 140, 145, 161, 166]. Parallel to asymptotic analysis in traditional adaptive control and system identification (sysID) [17, 106, 122], model-based finite-time control has benefited from a least-squares approach to identification followed by robust synthesis–see for example [50]. In this direction, probabilistic bounds on the estimation error related to the required run-time have been examined. While it has been shown that model-based methods require fewer measurements for certain control problems in general [163],[1] data collection required for sysID can be expensive or impractical due to resource limitations and safety constraints. Furthermore, some of the aforementioned studies rely on *a priori* information about the system, such as estimates of system parameters [51], an initial stabilizing controller [9,27,66,95], or a stable open-loop system [126,140]. In addition, persistently exciting inputs are not practical for control and identification of unstable systems even in low dimensions without recourse to resets [49].[2] Hence, existing data-guided methods are not directly applicable to time and safety critical control for applications such as flight control [107] or infrastructure recovery [74]. Our work is motivated by the desire to remove the reliance on having access to the initial stabilizing controller for data-guided control synthesis. In this direction, we focus on instances where input parameters in the LTI system are "effective." This point of view then facilitates imposing a satisfactory performance guarantee on data-guided synthesis based on a single trajectory–even when the underlying system is unstable.

In the first part of the analysis, we provide deterministic non-asymptotic error bounds

[1] That is, first finding a model estimate from data and then use that estimate for control design.

[2] For instance, injecting white noise into an unstable system can result in ill-conditioned data matrices.

for fitting a linear model to observed time-series data, with a particular attention to the role of symmetry and eigenvalue multiplicity in the underlying system matrix. Then we shift towards the control problem in a similar setup and examine online regulation of (possibly unstable) partially unknown linear systems with no *a priori* assumptions on the initial controller. we introduce and characterize the notion of "regularizability" for linear systems that gauges the capacity of a system to be regulated in finite-time in contrast to its asymptotic behavior (commonly characterized by stabilizability/controllability). Next, having access only to the input matrix, we propose the DGR synthesis that—as its name suggests—regulates the underlying states while also generating informative data that can subsequently be used for data-driven stabilization or sysID. We further elucidate our results by considering special structures for system parameters as well as boosting the performance of the algorithm via a rank-one matrix modification.

## *4.2 Linear Model Regression on Time-series Data*

For a wide range of real-world systems, the underlying complex dynamics makes deriving the corresponding models from first principles difficult if not infeasible. This can be due to a range of factors from the unpredictable nature of the environment to perturbations and uncertainties in the complex system [10, 50]. However, with the availability of sensing technologies and high performance computing, time-series data can be collected from these systems. Hence, it becomes natural to consider to what extent the observed time-series can be used to reason about the underlying dynamic model. In its most basic form, linear regression is used to find the system parameters by solving a least-squares minimization problem constructed on the observed time-series and examples of such an approach can be found in a wide variety of applications [11, 128, 132]. In the case when this data has been generated by simulations (a model, albeit complex, already exists), one might still be interested to reason about the dynamics using "simple" models. The adoption of this approach involves using prior knowledge of the underlying dynamics to choose a particular basis or library, and then postulating that the dynamic system is some combination of these basis elements. This problem then reduces to a parameter optimization problem - with respect to these basis or library- for their combination that best fits the given data,

with respect to a suitable norm or metric. In the absence of any prior assumption on the dynamics, however, it is often desirable to explore simple models. We examine non-asymptotic error bounds for doing such a linear fit, for the case when the data had been generated by a linear system; generically, it is the case that if the data is rich enough and the system does not have degeneracies, exact model is obtained after the number of data snapshots is the dimension of the system.[3] In fact, we show that even in this most streamlined case, and even in the absence of noise or uncertainty in the collected data, understanding non-asymptotic behavior of the error requires some non-trivial analysis. Consider the discrete linear time-invariant system described by the state equation,

$$x_{t+1} = Ax_t, \qquad t = 0, 1, 2, \ldots, \tag{4.1}$$

where $A \in \mathbb{R}^{n\times n}$ is the unknown system matrix, $x_t \in \mathbb{R}^n$ is the state of the system at time $t$, and the system has been initialized from $x_0$. The state snapshots collected up to (and including) time $k$ can now be grouped as,

$$X_k = [x_0 \quad x_1 \quad \ldots \quad x_k], \qquad Y_k = [x_1 \quad x_2 \quad \ldots \quad x_{k+1}]. \tag{4.2}$$

This data collection approach is analogous to the first step of the so-called DMD algorithm [162], where this step is followed by the parameterization of the eigenvalues and eigenvectors of the underlying system model.[4] Now to estimate the underlying system matrix at each $k$, we consider the the least-squares minimization, $\hat{A}_k = \text{argmin}_A \|Y_k - AX_k\|_F$, whose solution is of the form $\hat{A}_k = Y_k X_k^\dagger$; see Fig. 4.1. Note that when $\mathbf{rank}(X_k) = n$, $\hat{A}_k = AX_kX_k^\dagger = AX_kX_k^\top(X_kX_k^\top)^{-1} = A$.

[3]The model has $n^2$ unknown entries; as such, $n^2$ observations are generically needed for its exact recovery. One of course can get away with less data by invoking sparsity (say, using the $\ell_1$ regularization) or structure on the model, e.g, assuming an underlying pattern for the system matrix.

[4]The main objective of DMD is however not "model" regression per se, as it is "modal" fitting, in order to provide useful insights into the underlying (possibly nonlinear) dynamics.

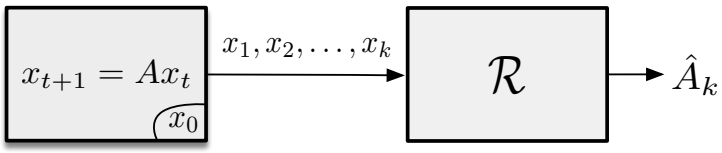



Figure 4.1: Estimating the underlying dynamics $A$ after $k$ data snapshots using the model regression $\mathcal{R}$

The focus of this part of the thesis is on the analysis of the error $\|A - \hat{A}_k\|$, *i.e.*, the non-asymptotic error between the original and estimated models, using linear regression when $\mathbf{rank}(X_k) < n$. Our work considers an *online* estimation of the model $A$, where each new data snapshot is added to the previously collected set. At any time-step $k$, an estimate for $A$ is found based on the received data up to $k$. The resulting data-driven recursive minimization is depicted in Fig. 4.1. Although not discussed further here, we note that diagram in Fig. 4.1 can–in principle–be augmented with a model-based control or filtering scheme that utilizes $\hat{A}_k$ after a suitable number of steps. we first introduce an upper bound on the estimation error as a function of the system dynamics $A$, the iteration count $k$, dimension of the system $n$, and the initial conditions $x_0$. In particular, we show that for each time-step, the left-singular vectors of the Singular Value Decomposition (SVD) of the data matrix dictate the estimation error bound. Next, we focus on symmetric system matrices. In this case, it is shown that the model regression error can be characterized by the multiplicities in the spectrum of the underlying system.

### *4.2.1 Non-asymptotic Error Analysis*

In this section, we examine the error bound for the linear system regression in (4.1) based on the system characteristics and the observed data snapshots. We assume that $k < n$, *i.e.*, the number of data snapshots is less than the size of the system. The next results, characterizes the regression estimation error as a function of the time step $k$.

**Theorem 21.** *Consider the system in* (4.1) *and the corresponding data matrix. Let* $X_k = U_k \Sigma_k V_k^\top$ *be the SVD of* $X_k$. *Then the model estimate at time-step* $k$ *is given by* $\hat{A}_k =$

$A(I - E_k)$ *where,*

$$E_k = \left( I - \frac{S_k P_k}{\mathrm{Tr}(S_k P_k)} \right) S_k \,, \tag{4.3}$$

*with,*

$$S_k = I - U_{k-1} U_{k-1}^\top, \qquad P_k = x_k x_k^\top = A^k x_0 x_0^\top A^{k\top} \,. \tag{4.4}$$

*Moreover,*

$$\|E_k\|_2 \leq \left\| I - \frac{S_k P_k}{\mathrm{Tr}(S_k P_k)} \right\|_2 . \tag{4.5}$$

*Proof.* From (4.2), $X_k = \begin{bmatrix} X_{k-1} & A^k x_0 \end{bmatrix}$, and $Y_k = AX_k = A \begin{bmatrix} X_{k-1} & A^k x_0 \end{bmatrix}$. Then the estimate of the system matrix after the $k$-th snapshot is given by $\hat{A}_k = Y_k X_k^\dagger$, where $\hat{A}_k$ is the least-squares solution to $AX_k = Y_k$. Thus,

$$\hat{A}_k = Y_k X_k^\dagger = A X_k X_k^\dagger = A \begin{bmatrix} X_{k-1} & A^k x_0 \end{bmatrix} X_k^\dagger \,. \tag{4.6}$$

Hence we need to characterize $X_k^\dagger$. To this end, we start from $X_k^\dagger = \left( X_k^\top X_k \right)^{-1} X_k^\top$. We first note that,

$$X_k^\top X_k = \begin{bmatrix} X_{k-1}^\top \\ x_0^\top A^{k\top} \end{bmatrix} \begin{bmatrix} X_{k-1} & A^k x_0 \end{bmatrix} = \begin{bmatrix} X_{k-1}^\top X_{k-1} & X_{k-1}^\top A^k x_0 \\ x_0^\top A^{k\top} X_{k-1} & x_0^\top A^{k\top} A^k x_0 \end{bmatrix} .$$

Then

$$\left( X_k^\top X_k \right)^{-1} = \frac{1}{\zeta} \begin{bmatrix} \Phi & -X_{k-1}^\dagger A^k x_0 \\ -x_0^\top A^{k\top} X_{k-1}^{\dagger\top} & 1 \end{bmatrix},$$

where

$$\begin{aligned} \Phi &= \left( X_{k-1}^\top X_{k-1} \right)^{-1} \left[ \zeta I + X_{k-1}^\top A^k x_0 x_0^\top A^{k\top} X_{k-1}^{\dagger\top} \right], \\ \zeta &= x_0^\top A^{k\top} \left[ I - X_{k-1} \left( X_{k-1}^\top X_{k-1} \right)^{-1} X_{k-1}^\top \right] A^k x_0 \\ &= -x_0^\top A^{k\top} \left[ X_{k-1} X_{k-1}^\dagger - I \right] A^k x_0 \,. \end{aligned}$$

In the meantime, $X_k^\dagger = \left( X_k^\top X_k \right)^{-1} X_k^\top \begin{bmatrix} \Psi_1 \\ \Psi_2 \end{bmatrix}$, with,

$$\begin{aligned} \Psi_1 &= X_{k-1}^\dagger + \frac{1}{\zeta} X_{k-1}^\dagger A^k x_0 x_0^\top A^{k\top} \left( U_{k-1} U_{k-1}^\top - I \right), \\ \Psi_2 &= -\frac{1}{\zeta} x_0^\top A^{k\top} \left( U_{k-1} U_{k-1}^\top - I \right), \end{aligned}$$

where we have used the fact $X_{k-1}X_{k-1}^{\dagger} = U_{k-1}U_{k-1}^{\top}$. Thereby, we can expand $\hat{A}_k$ from (4.6) as,

$$\begin{aligned}\hat{A}_k = AX_kX_k^{\dagger} &= A\left(U_{k-1}U_{k-1}^{\top}\right) \\ &\quad + A\left(\frac{1}{\zeta}\left(U_{k-1}U_{k-1}^{\top} - I\right)A^k x_0 x_0^{\top} A^{k^{\top}}\left(U_{k-1}U_{k-1}^{\top} - I\right)\right) \\ &= A\left(U_{k-1}U_{k-1}^{\top}\right) \\ &\quad - A\left(\frac{\left(U_{k-1}U_{k-1}^{\top} - I\right)A^k x_0 x_0^{\top} A^{k^{\top}}\left(U_{k-1}U_{k-1}^{\top} - I\right)}{x_0^{\top} A^{k^{\top}}\left(U_{k-1}U_{k-1}^{\top} - I\right)A^k x_0}\right),\end{aligned}$$

and from (4.4), the estimated model $\hat{A}_k$ at time-step $k$ is given by $\hat{A}_k = A\left(I - E_k\right)$ with,

$$E_k = \frac{\left(\mathrm{Tr}(S_kP_k)I - S_kP_k\right)S_k}{\mathrm{Tr}(S_kP_k)} = \left(I - \frac{S_kP_k}{\mathrm{Tr}(S_kP_k)}\right)S_k.$$

The magnitude of this error simplifies for the case of the spectral norm as,

$$\|E_k\|_2 = \left\|\left(I - \frac{S_kP_k}{\mathrm{Tr}(S_kP_k)}\right)S_k\right\|_2 \leq \left\|I - \frac{S_kP_k}{\mathrm{Tr}(S_kP_k)}\right\|_2,$$

since $\|S_k\|_2 = 1$ for $k < n$. Lastly, we note that $S_k$ is the projection onto the null space of $X_{k-1}$ and $P_k$ is the covariance matrix of the data at time-step $k$; as such both matrices are positive-semidefinite. □

We note that the relation (4.3) captures–in a succinct way–the dependency of the model regression error on how new modes are revealed by the data stream over time.

### *4.2.2 Non-asymptotic Error Analysis for Symmetric Systems*

In this section, we consider linear systems with symmetric dynamics with the aim of characterizing fundamental bounds on the regression error in terms of the spectral properties of the system. This insight into the regression error is achieved through the spectral decomposition of the system matrix,

$$A = Q\Lambda Q^{\top} = \sum_{i=1}^{r} \lambda_i q_i q_i^{\top}, \tag{4.7}$$

where $Q$ is the unitary matrix containing the eigenvectors corresponding to nonzero eigenvalues of $A$, $\Lambda$ is the diagonal matrix of nonzero eigenvalues, and $r = \mathbf{rank}(A)$. Symmetric

system matrices appear in a wide range of applications where interactions leading to the dynamics is bidirectional; such systems are of interest in biological networks [176], social interactions [8], robotic swarms [32], and networks security [5]. Using this spectral decomposition of symmetric systems, we show that the regression error is dependent on the multiplicity of eigenvalues in $A$. In particular, we show that if $m(\lambda) = 1, \forall \lambda \in \mathbf{\Lambda}(A)$, then the upper bound (4.5) is a function of the largest and smallest singular values of the system matrix as well as its rank. Otherwise, the regression error is $\max_{i:m(\lambda_i)>1} |\lambda_i|$. We provide the details of the approach for each case.

*Simple Eigenvalues*

We first consider the case when the symmetric system matrix has simple eigenvalues and rank $r$. In reference to Equation (4.7), consider the entire set of eigenvectors of $A$ consisting of $Q = [q_1 \ \cdots \ q_r]$ and $\bar{Q} = [\bar{q}_{r+1} \ \cdots \ \bar{q}_n]$, where $\{q_1, \cdots, q_r, \bar{q}_{r+1}, \cdots, \bar{q}_n\}$ forms a basis for the entire $\mathbb{R}^n$. Then the nonzero random initial state $x_0$ can be written as,

$$x_0 = Q\nu + \bar{Q}\mu = \sum_{i=1}^{r} \nu_i q_i + \sum_{i=r+1}^{n} \mu_i \bar{q}_i \qquad \nu \neq 0. \tag{4.8}$$

**Lemma 18.** *For the symmetric linear system decomposed as (4.7), we have,* $A - \hat{A}_k = A\Big(I - \frac{S_k Q \Lambda^k \nu \nu^\top \Lambda^k Q^\top}{\|S_k Q \Lambda^k \nu\|^2}\Big) S_k$.

*Proof.* From (4.8) and (4.4) we have, $P_k = A^k x_0 x_0^\top {A^k}^\top$

$$\begin{aligned} &= (Q\Lambda^k Q^\top)(Q\nu + \bar{Q}\mu)(Q\nu + \bar{Q}\mu)^\top (Q\Lambda^k Q^\top) \\ &= Q\Lambda^k \nu\nu^\top \Lambda^k Q^\top, \end{aligned}$$

and, $\mathrm{Tr}(S_k P_k) = \mathrm{Tr}(S_k Q \Lambda^k \nu\nu^\top \Lambda^k Q^\top) = \|S_k Q \Lambda^k \nu\|^2$, where we have used that $S_k$ is a symmetric projection. Substituting these in (4.3) completes the proof. □

We now show that when $k < n$, the error depends on the largest and smallest eigenvalues of $A$.

**Theorem 22.** *Consider the linear dynamical system with symmetric system matrix $A$ as in (4.7) and the initial state $x_0$ as in (4.8). If $k < n$, then*

$$\|A - \hat{A}_k\|_F^2 \leq \Big(n - \min\{k, |\nu| + \min\{|\mu|, 1\}\}\Big)\lambda_1^2 - \lambda_n^2 \,,$$

*where* $\lambda_n = \lambda_{\min}(A)$, $\lambda_1 = \lambda_{\max}(A)$, *and* $|\nu|$ *and* $|\mu|$ *are the number of nonzero* $\nu_i$*'s and* $\mu_i$*'s, respectively.*

*Proof.* From Lemma 18 we observe that,

$$\begin{aligned}\|A - \hat{A}_k\|_F^2 &= \mathrm{Tr}\Big((A - \hat{A}_k)^\top (A - \hat{A}_k)\Big) \\ &= \mathrm{Tr}(\Lambda^2 Q^\top S_k Q) - \frac{\nu^\top \Lambda^k Q^\top S_k Q \Lambda^2 Q^\top S_k Q \Lambda^k \nu}{\nu^\top \Lambda^k Q^\top S_k Q \Lambda^k \nu} \\ &= \|AS_k\|^2 - \frac{\|AS_k Q\Lambda^k \nu\|^2}{\|S_k Q \Lambda^k \nu\|^2}\,.\end{aligned}$$

In the meantime,

$$\|AS_k\|_F^2 \le \mathbf{rank}(S_k)\|AS_k\|_2^2 \le \mathbf{rank}(S_k)\|A\|_2^2\|S_k\|_2^2 = \mathbf{rank}(S_k)\lambda_1^2(A);$$

moreover, since $\lambda_n(A) = \inf_{y\neq 0} \|Ay\|_2/\|y\|_2$, we have

$$\frac{\|AS_k Q\Lambda^k \nu\|^2}{\|S_k Q \Lambda^k \nu\|^2} \ge \lambda_n^2(A).$$

Since $Aq_i = \lambda_i q_i$ and $A\bar{q_i} = 0$, we have,

$$X_{k-1} = [x_0 \quad x_1 \quad \cdots \quad x_{k-1}] = \left[\sum_{i=1}^{r} \nu_i q_i + \sum_{i=r+1}^{n} \mu_i \bar{q}_i \quad \sum_{i=1}^{r} \lambda_i \nu_i q_i \quad \cdots \quad \sum_{i=1}^{r} \lambda_i^{k-1} \nu_i q_i\right].$$

Thus,

$$\mathbf{rank}(X_{k-1}) = \min\{k, |\nu| + \min\{|\mu|, 1\}\}\,, \tag{4.9}$$

and,

$$\mathbf{rank}(S_k) = n - \mathbf{rank}(X_{k-1}) = n - \min\{k, |\nu| + \min\{|\mu|, 1\}\}\,.$$

Hence,

$$\|A - \hat{A}_k\|_F^2 \le \Big(n - \min\{k, |\nu| + \min\{|\mu|, 1\}\}\Big)\lambda_1^2 - \lambda_n^2,$$

which completes the proof. □

*Effect of Eigenvalues Multiplicity on the Regression Error*

In this section, we consider the symmetric systems whose eigenvalues are not necessary simple. We will see that for such systems, the regression error $\|E_k\|$ converges to the largest

eigenvalue with multiplicity greater than one, i.e., $\|E_k\| = \max_{i:m(\lambda_i)>1} |\lambda_i|$ for $k \geq n$. In order to show this, we will pursue the convention adopted in (4.2), (4.7), and (4.8). As in the previous section, let $r = \mathbf{rank}(A)$ and define $\tilde{Q} = [Q \mid \bar{Q}] = [q_1 \ \dots \ q_r \ \ \bar{q}_{r+1} \ \dots \ \bar{q}_n]$, where the columns of $\tilde{Q}$ span the entire $\mathbb{R}^n$. Furthermore, let $\alpha = [\nu^\top \ \mu^\top]^\top$, where $\nu$ and $\mu$ are from (4.8). The data matrix can now be re-written as,

$$X_k = [x_0 \quad Ax_0 \quad A^2x_0 \quad \dots \quad A^kx_0] = \tilde{Q}\,[\tilde{Q}^\top x_0 \quad \Lambda\tilde{Q}^\top x_0 \quad \dots \quad \Lambda^k\tilde{Q}^\top x_0]. \tag{4.10}$$

From (4.8) and the fact that the columns of $\tilde{Q}$ are orthonormal, we have $\tilde{Q}^\top x_0 = [\alpha_1 \ \alpha_2 \ \dots \ \alpha_n]^\top$ and $\Lambda^j\tilde{Q}^\top x_0 = [\alpha_1\lambda_1^j \quad \alpha_2\lambda_2^j \quad \dots \quad \alpha_n\lambda_n^j]^\top$. Moreover, in light of (4.10) we can decompose the data matrix into $X_k = \tilde{Q}\Gamma V$ where,

$$\Gamma = \begin{bmatrix} \alpha_1 & 0 & \dots & 0 \\ 0 & \alpha_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \alpha_n \end{bmatrix}, \qquad V = \begin{bmatrix} 1 & \lambda_1 & \dots & \lambda_1^k \\ 1 & \lambda_2 & \dots & \lambda_2^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_n & \dots & \lambda_n^k \end{bmatrix}. \tag{4.11}$$

The matrix $[V]_{ij} = \lambda_i^{j-1}$ is the Vandermonde matrix formed by the eigenvalues of $A$ and $\Gamma = \mathbf{Diag}([\alpha_i]_{i=1}^n)$.[5] Assume now that the system matrix $A$ contains $s$ distinct eigenvalues and let $\mathbf{\Lambda}^*(A) = \{\lambda_{t_1}, \lambda_{t_2}, \dots, \lambda_{t_s}\}$ be the set of these eigenvalues; as such, the other $n - s$ eigenvalues are repetitions of the elements in $\mathbf{\Lambda}^*(A)$.

For our subsequent error analysis, we will use the rank of $X_k$ at each time-step. The next result characterizes the rank of $X_k$ based on the number of the collected data.

**Lemma 19.** *Given $k < s$, the $s \times k$ Vandermonde matrix defined by $[V_s]_{ij} = \lambda_i^{j-1}$, $i \in \{1, \dots, s\}$, $j \in \{1, \dots, k\}$, formed by the elements of $\mathbf{\Lambda}^*(A)$, has full-rank.*

*Proof.* Let $v_i$ be the $i$th column of $V_s$ and assume that $c_1v_1 + c_2v_2 + \cdots + c_kv_k = 0$. Consider row $p$ of the equation $c_1 + c_2\lambda_p + \cdots + c_k\lambda_p^k = 0$. Since $\lambda_i \neq \lambda_j$ for $i \neq j$, there exist $s$ solutions to the $k$-degree polynomial,

$$P(x) = c_0 + c_1x + \cdots + c_kx^k = 0.$$

[5] Note that it is assumed that for all $i$, $x_0 \not\perp q_i$; in this case, $\alpha_i \neq 0$ for all $i$ and $\mathbf{rank}(X_k) = \mathbf{rank}(V)$.

Hence $c_1 = c_2 = \cdots = c_k = 0$ and $v_i$'s are linearly independent and since $k < s$, $V_s$ has full-rank. □

**Theorem 23.** *Let $k$ be the number of collected data snapshots and $s = |\mathbf{\Lambda}^*(A)|$. Then $\mathbf{rank}(X_k) = k$ when $k < s$ and $\mathbf{rank}(X_k) = s$ when $k \geq s$.*

*Proof.* For $k < s$, it is straightforward to show that from Lemma 19, $\mathbf{rank}(X_k) = \mathbf{rank}(V) = \mathbf{rank}(V_s) = k$. For $k \geq s$, we know that $\mathbf{rank}(V) = \mathbf{rank}(V_s)$, where $[V_s]_{ij} = \lambda_i^{j-1}$, $i \in \{1, \ldots, s\}$, $j \in \{1, \ldots, k+1\}$. Since $V_s$ has full-rank (this can be shown using the nonzero sub-matrix determinants for the first $s \times s$ block), $\mathbf{rank}(X_k) = \mathbf{rank}(V) = \mathbf{rank}(V_s) = s$. □

Let $k \geq s$ and $X_k = U\Sigma V^\top$ be the SVD of $X_k$ where,

$$X_k = [U_1 \quad U_2] \begin{bmatrix} \Sigma_1 & \mathbf{0}_{s\times(k-s)} \\ \mathbf{0}_{(n-s)\times s} & \mathbf{0}_{(n-s)\times(k-s)} \end{bmatrix} \begin{bmatrix} V_1 & V_2 \end{bmatrix}^\top, \tag{4.12}$$

with $U_1 \in \mathbb{R}^{n\times s}$, $U_2 \in \mathbb{R}^{n\times(n-s)}$, $\Sigma_1 \in \mathbb{R}^{s\times s}$, $V_1 \in \mathbb{R}^{k\times s}$, $V_2 \in \mathbb{R}^{k\times(k-s)}$. The existence of $U_2$ and $V_2$ is due to the fact that $X_k$ is degenerate. W.l.o.g, we re-arrange the columns of $\tilde{Q}$ and $\Lambda$ such that,

$$\Lambda_R = \left[\begin{array}{c|c} \Lambda_1 & \mathbf{0} \\ \hline \mathbf{0} & \Lambda_2 \end{array}\right], \quad \tilde{Q}_R = \left[\begin{array}{c|c} \tilde{Q}_1 & \tilde{Q}_2 \end{array}\right], \tag{4.13}$$

where $\Lambda_1 = \mathbf{Diag}([\lambda_{t_1} \quad \lambda_{t_2} \quad \ldots \quad \lambda_{t_s}])$ contains the element of $\mathbf{\Lambda}^*(A)$ with the corresponding eigenvectors in $\tilde{Q}_1 \in \mathbb{R}^{n\times s}$. The remaining eigenvalues and the corresponding eigenvectors are stacked in $\Lambda_2 \in \mathbb{R}^{(n-s)\times(n-s)}$ and $\tilde{Q}_2 \in \mathbb{R}^{n\times(n-s)}$, respectively.

A crucial term in our analysis is $U_2^\top \tilde{Q}_R \in \mathbb{R}^{(n-s)\times n}$. This matrix multiplication combines a submatrix of $U$ corresponding to the repeated eigenvalues, with the orthonormal eigenvectors of the system. In the next result, we show that this term has a specific row structure.

**Lemma 20.** *Given the convention in (4.13), we have $U_2^\top \tilde{Q}_1 = \boldsymbol{0}$, i.e., $U_2^\top \tilde{Q}_R = U_2^\top [\tilde{Q}_1 \mid \tilde{Q}_2] = [\ \boldsymbol{0}_{(n-s)\times s} \mid U_2^\top \tilde{Q}_2\ ]$.*

*Proof.* From (4.12) we know $U_2^\top X_k = \mathbf{0}$. Then,

$$\begin{aligned} U_2^\top X_k &= U_2^\top \tilde{Q}\tilde{Q}^\top X_k = U_2^\top \tilde{Q}\tilde{Q}^\top [x_0 \quad Ax_0 \quad \dots \quad A^k x_0] \\ &= U_2^\top \tilde{Q} [\tilde{Q}^\top x_0 \quad \Lambda \tilde{Q}^\top x_0 \quad \dots \quad \Lambda^k \tilde{Q}^\top x_0] \\ &= U_2^\top \tilde{Q} \Gamma V = \mathbf{0}, \end{aligned}$$

with $\Gamma$ and $V$ defined as in (4.11). Define $B = U_2^\top \tilde{Q}$ and let $b_i = [b_{i_1} \quad b_{i_2} \quad \dots \quad b_{i_n}]$ and $v_i = [1 \quad \lambda_i \quad \dots \quad \lambda_i^k]$ be the $i$'th rows of $B$ and $V$, respectively. Then for all $i \in \{1, 2, \dots, n-s\}$ we have $B\Gamma V = \mathbf{0}$ implying that $b_i \Gamma V = \mathbf{0}$; as such, $\sum_{j=1}^{n} b_{i_j} \alpha_j v_j = \mathbf{0}$ implies that $\sum_{t=1}^{s} c_t v_t^* = \mathbf{0}$, where $v_t^*$'s are the rows of $V$ corresponding to $s$ distinct eigenvalues and $c_t$'s are some combinations of $b_{i_j}\alpha_j$'s. Since the vectors $v_t^*$'s are linearly independent, we get $c_i = 0$ for all $i = 1, 2, \dots, s$. Considering that $c_i = b_{i_j}$ for the elements with simple eigenvalues, the result implies that for any row $v_j$ corresponding to a simple eigenvalue $\lambda_j$ the corresponding coefficient $b_{i_j} = 0$. Hence, the structure $\left[\, \mathbf{0}_{(n-s)\times s} \mid U_2^\top \tilde{Q}_2 \,\right]$ follows, implying that $U_2^\top \tilde{Q}_1 = \mathbf{0}$. □

**Definition 6.** *Define $U_2'$ by permuting the columns of $U_2$ such that we obtain block diagonal matrix $U_2'^\top \tilde{Q}_2 = \mathbf{Diag}([P_1, P_2, \dots, P_\ell])$, where $\ell$ is the number of eigenvalues $\lambda$ with $m(\lambda) > 1$ and $P_i = U_2'^{i\top} \tilde{Q}_2^i \in \mathbb{R}^{(m(\lambda_i)-1)\times m(\lambda_i)}$; as such $U_2'^i \in \mathbb{R}^{n\times(m(\lambda_i)-1)}$ is the matrix containing vectors in $U_2'$ corresponding to $\lambda_i$ and $\tilde{Q}_2^i \in \mathbb{R}^{n\times m(\lambda_i)}$ is the matrix of eigenvectors corresponding to $\lambda_i$.*

**Remark 9.** *To justify the existence of such a matrix $U_2'$, notice that the SVD factorization in terms of $U$ and $V$ are not unique and for any such factorization, $U_1 \perp U_2$.*

We are now well positioned to prove the main theorem of this section.

**Theorem 24.** *Consider the dynamics represented by (4.1) and (4.2). Let $s = |\mathbf{\Lambda}^*(A)|$ be the number of distinct eigenvalues of $A$. Assume that $k \geq s$ and let $\lambda^* = \max_{i:m(\lambda_i)>1} |\lambda_i|$ be the largest eigenvalue of $A$ with multiplicity greater than one. Then $\|E_k\|_2 = \|A - \hat{A}_k\|_2 = \lambda^*$.*

*Proof.* We will show that $\mathbf{\Lambda}(E_k) = \mathbf{\Lambda}(A) \backslash \mathbf{\Lambda}^*(A)$. The error can then be re-written as,

$$\begin{aligned}
\|E_k\|_2 &= \|A - \hat{A}_k\|_2 = \|A - AX_kX_k^\dagger\|_2 = \|A(I - X_kX_k^\dagger)\|_2 \\
&= \left\|A\left(I - U_1\Sigma_1V_1^\top V_1\Sigma_1^{-1}U_1^\top\right)\right\|_2 = \|A(I - U_1U_1^\top)\|_2 \\
&= \|AU_2'U_2'^\top\|_2 = \|\tilde{Q}\Lambda\tilde{Q}^\top U_2'U_2'^\top\|_2 = \|\Lambda\tilde{Q}^\top U_2'U_2'^\top\tilde{Q}\|_2 \\
&= \|\Lambda(U_2'^\top\tilde{Q})^\top(U_2'^\top\tilde{Q})\|_2,
\end{aligned}$$

where $U_2'$ pertains to definition 6 and $I - U_1U_1^\top = U_2'U_2'^\top$ is the projection matrix onto $\mathcal{N}(X_k^\top)$. Note that since the columns of $U_2'$ are linearly independent, we have $\mathbf{rank}(E_k) = \mathbf{rank}(AU_2'U_2'^\top) = n - s$. From Lemma 20,

$$\begin{aligned}
\Lambda(U_2'^\top\tilde{Q})^\top(U_2'^\top\tilde{Q}) &= \Lambda\begin{bmatrix}\mathbf{0} \\ (U_2'^\top\tilde{Q}_2)^\top\end{bmatrix}\begin{bmatrix}\mathbf{0} & U_2'^\top\tilde{Q}_2\end{bmatrix} \\
&= \left[\begin{array}{c|c} \mathbf{0}_{s\times s} & \mathbf{0}_{s\times(n-s)} \\ \hline \mathbf{0}_{(n-s)\times s} & \Lambda_2(U_2'^\top\tilde{Q}_2)^\top(U_2'^\top\tilde{Q}_2)\end{array}\right].
\end{aligned} \tag{4.14}$$

Then from definition 6, $\Lambda_2(U_2'^\top\tilde{Q}_2)^\top(U_2'^\top\tilde{Q}_2) = \mathbf{Diag}([\lambda_1P_1^\top P_1\ ,\ \lambda_2P_2^\top P_2\ ,\ \ldots\ ,\ \lambda_\ell P_\ell^\top P_\ell])$. Consider the $i$th block $\lambda_iP_i^\top P_i$. Notice that from definition $\mathbf{rank}(P_i^\top P_i) = m(\lambda_i) - 1$, and since ${U_2'}^{i\top}$ and $\tilde{Q}_2^i$ are orthonormal, $\lambda_iP_i^\top P_i = \lambda_i\tilde{Q}_2^{i\top}U_2'^iU_2'^{i\top}\tilde{Q}_2^i$ has the spectrum $\mathbf{\Lambda}(\lambda_iP_i^\top P_i) = \{0, \lambda_i, \ldots, \lambda_i\}$. Then having $\ell$ of these blocks $\mathbf{\Lambda}(E_k) = \mathbf{\Lambda}(A)\backslash\mathbf{\Lambda}^*(A)$, i.e., the spectrum of $E_k$ contains all repeated eigenvalues of $A$. Since both $\Lambda$ and $(U_2'^\top\tilde{Q})^\top(U_2'^\top\tilde{Q})$ are symmetric square block diagonal matrices, the product $\Lambda(U_2'^\top\tilde{Q})^\top(U_2'^\top\tilde{Q})$ is symmetric and therefore $\|E_k\|_2 = \|\Lambda(U_2'^\top\tilde{Q})^\top(U_2'^\top\tilde{Q})\|_2 = \max\limits_{i=1,\ldots,\ell}\lambda_i = \lambda^*$.[6] □

### *4.2.3 Example: Model Regression on Networks*

We now provide an example to demonstrate the applicability of the error bounds on networked systems. Consider the Petersen graph on 10 nodes and 15 edges as shown in Figure 4.2. We use the weighted version of this specific structure to find error bounds on a system with simple eigenvalues. The dynamics of this system is defined using the *graph*

[6] It is straightforward to show an analogous result for the Frobenius norm $\|E_k\|_F^2 = \|A - \hat{A}_k\|_F^2 = \sum_{i=1}^{\ell}[m(\lambda_i) - 1]\lambda_i^2$.

*Laplacian*, defined as $\mathcal{L} = D - A$, where $A$ is the *adjacency* matrix that defines the connections in the network and $D$ is the *degree* matrix defined as $D_{ii} = \sum_j |W_{ij}|$; in this case $W_{ij}$ is the weight of the edge between nodes $i$ and $j$. Network symmetries typically induce eigenvalue multiplicities in the corresponding adjacency and Laplacian matrices. Hence to make the system more generic, we add weights $w_{1,6} = 1$, $w_{2,7} = 2$, $w_{3,8} = 3$, $w_{4,9} = 4$, and $w_{5,10} = 5$ and for all other weights we have $w_{i,j} = 1$. For each component $i$ the dynamics depend on the adjacent nodes in the graph $\dot{x}_i = \sum_j |W_{ij}|(x_i - x_j)$. Then the overall dynamics can be written as $\dot{x} = -\mathcal{L}x$. The model regression algorithm discussed here leads to the error shown in Figure 4.2. For this simulation, the initial condition has been chosen as a (normalized) random vector $x_0 \in \mathbb{R}^{10}$. The upper subfigure shows the comparison of the bound for general case using the spectral norm and the lower subfigure demonstrates the same setup for the symmetric results. It can be seen that the error converges to zero after $k = 10$ steps, since the system matrix has simple eigenvalues.

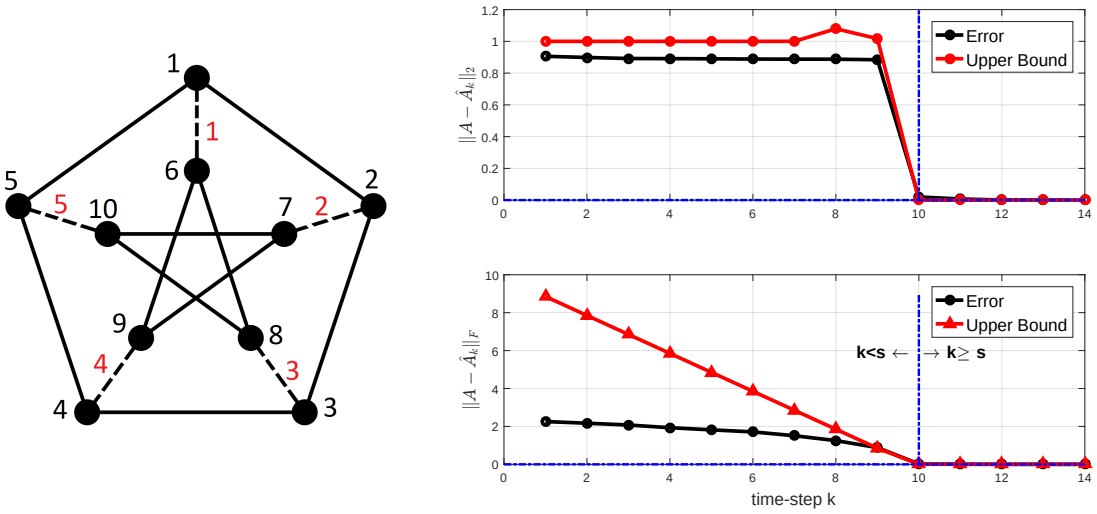



Figure 4.2: (left) weighted Petersen graph; (right): model regression error the corresponding theoretical error bound

### *4.3 Online Regulation of Unstable Linear Systems*

In this section, we introduce the problem setup and highlight its unique feature through an example. Consider a discrete-time LTI system of the form,

$$\boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + B\boldsymbol{u}_t, \qquad \boldsymbol{x}_0 \text{ given}, \tag{4.15}$$

where $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are the system parameters and $\boldsymbol{x}_t \in \mathbb{R}^n$ and $\boldsymbol{u}_t \in \mathbb{R}^m$ denote the state and control inputs at iteration $t$, respectively. We assume that the system matrix $A$ is unknown and (possibly) unstable, and that the input matrix $B$ is known. The problem of interest is to design $\boldsymbol{u}_t$ from state measurements (and not the system matrix $A$) such that: 1) the system is regulated, with a norm uniformly bounded during the learning process, *e.g.*, $\boldsymbol{x}_t$ evolves in a safe region with a quantifiable bounded norm, and the corresponding data matrix does not become ill-condition, and 2) the system generates informative data for post-processing such as data-driven stabilization or system identification.[7]

Considering regulation by having access to input matrix is of interest in applications where it is known a priori how various control inputs effect the dynamic states, *e.g.*, the effect of elevator deflection on the pitch dynamics for longitudinal aircraft dynamics. Intuitively, this assumption allows an online regulation mechanism to have a chance of stabilizing an unknown unstable system in real-time. The data-guided regulation scenario where in addition to the uncertain system matrix, the input matrix is unknown will be examined in our future works.

The following example motivates our setup and underscores why the data-guided perspective requires introducing new system theoretic notions.

**Example 1.** *For any positive integer $n$, define the system matrix $A \in \mathbb{R}^{n\times n}$ and the input*

[7] Inspired by [164], we interchangeably use the terms linear independence and *informativity* of data to emphasize that the collected data has useful information content for decision-making.

*matrix* $B \in \mathbb{R}^n$ *as,*

$$A = \begin{pmatrix} \lambda_1 & 1 & 0 & \dots & 0 \\ 0 & \lambda_2 & 1 & & \vdots \\ 0 & 0 & \lambda_3 & \ddots & 0 \\ \vdots & & \ddots & \ddots & 1 \\ 0 & \dots & & 0 & \lambda_n \end{pmatrix}, \qquad B = \begin{pmatrix} 0 \\ \\ \vdots \\ \\ 0 \\ 1 \end{pmatrix}.$$

*Note that for any choice of* $\lambda_i \in \mathbb{R}$*, the pair* $(A, B)$ *is controllable (and therefore stabilizable). Furthermore, since the set* $\{\lambda_i\}$ *coincides with the spectrum of* $A$*, if any subset of* $\{\lambda_i\}$ *are equal, then* $A$ *contains a Jordan block corresponding to that subset. Moreover, when* $\lambda_i \neq \lambda_j$ $(i \neq j)$*, then* $A$ *is diagonalizable. Let* $\boldsymbol{x}_0 = \boldsymbol{e}_1$ *and observe that under* (4.15)*, we have* $\boldsymbol{e}_1^\top \boldsymbol{x}_t = \lambda_1^t$ *for all* $0 \leq t < n$ *regardless of the input* $\boldsymbol{u}_t$*. This implies that, for "any" choice of input, for the first* $(n-1)$ *iterations, the first state of the system grows exponentially fast with the rate* $\lambda_1$ *whenever* $|\lambda_1| > 1$*.*

**Remark 10.** *Example 1 constructs a family of controllable systems where no controller can regulate their respective first states–at least for the first* $n$ *iterations. That is, a system state will grow exponentially fast regardless of the choice of* $\boldsymbol{u}_t$*, even when all eigenvalues of* $A$ *except* $\lambda_1$ *are stable (*e.g.*,* $|\lambda_i| < 1$ *for* $i = 2, \cdots, n$*). Note that in this example, the (right) eigenvector associated with the unstable mode of* $A$ *(i.e., the eigen-pair* $(\lambda_1, \boldsymbol{e}_1)$*) is orthogonal to* $\mathcal{R}(B) = \mathcal{R}(\boldsymbol{e}_n)$*. This is despite the fact that the PBH controllability test holds (i.e., for any left eigenvector* $\boldsymbol{v}$ *of* $A$ *we have* $\boldsymbol{v}^\top B \neq 0$*). This example highlights that controllability of a pair* $(A, B)$ *does not capture regularizability of an unstable linear system, specially when the regulation has been achieved in a data-guided manner and performance of the controller matters from the initialization. Finally, we point out that in the particular case when* $\lambda_i = 0$ *for* $i = 2, \dots, n$*, the controllability matrix corresponding to* $(A, B)$ *is anti-diagonal with all anti-diagonal elements equal to identity. Therefore, it has singular values/eigenvalues all equal to* $\pm 1$*. This implies that the controllability matrix has condition number equal to identity; as such modes that are difficult to regularize are not distinguished in the controllability matrix.*

In order to formalize the behavior of the class of systems mentioned above, we introduce a system theoretic notion to capture the effectiveness of the input characteristics pertinent to online regulation. In order to motivate this notion, note that the dynamics in equation (4.15) can be rewritten as,

$$\begin{aligned}\boldsymbol{x}_{t+1} &= \Pi_{\mathcal{R}(B)^\perp} A\boldsymbol{x}_t + \Pi_{\mathcal{R}(B)} A\boldsymbol{x}_t + B\boldsymbol{u}_t \\ &= \Pi_{\mathcal{R}(B)^\perp} A\boldsymbol{x}_t + B(B^\dagger A\boldsymbol{x}_t + \boldsymbol{u}_t).\end{aligned}$$

Setting $u_t = -B^\dagger A\boldsymbol{x}_t + \bar{u}_t$, the dynamics equation (4.15) can be rewritten as,

$$\boldsymbol{x}_{t+1} = \widetilde{A}\boldsymbol{x}_t + B\bar{\boldsymbol{u}}_t,$$

where,

$$\widetilde{A} := \Pi_{\mathcal{R}(B)^\perp} A, \tag{4.16}$$

and $\bar{\boldsymbol{u}}_t$ is yet to be designed. Note that the signals $\widetilde{A}\boldsymbol{x}_t$ and $B\bar{\boldsymbol{u}}_t$ are orthogonal. This implies that the control signal would not effect the part of dynamics that is generated by $\Pi_{\mathcal{R}(B)^\perp} A$. As such, in order to have even the possibility of achieving some online performance for this system in finite-time, we require that this part of the dynamics be stable. This observation thereby motivates the following definition.

**Definition 7.** *The pair $(A, B)$ is called regularizable if $\widetilde{A} := \Pi_{\mathcal{R}(B)^\perp} A$ is Schur stable.*

As we will show subsequently, regularizability of a pair $(A, B)$ is related to the stabilizability of $(A, B)$ as well as detectability of $(A, B^\top)$; a combination that is not typically encountered in control theoretic LTI analysis. This connection is intuitive, as regulation of a system in finite-time requires the states to be accessible through the input matrix $B$. It also provides a new perspective on LTI models which will be utilized as a basis for the analysis.

### *4.3.1 Regularizable Systems*

In order to get a better sense of the notion of regularizability, we study the spectral properties of $\widetilde{A}$ in equation (4.16) and its relation with system matrices $A$ and $B$. First, the following example highlights why regularizability of a system is distinct from its controllability.

**Example 2.** *Consider the linear system with $A$ defined as in Example 1 such that $|\lambda_1| > 1$ and $|\lambda_i| < 1$ for $i = 2, \dots, n$. Note that the pair $(A, \boldsymbol{e}_n)$ is controllable (and thus stabilizable); however this pair not regularizable. On the other hand, the pair $(A, \boldsymbol{e}_1)$ is regularizable but not controllable.*

Recall that a pair $(A, B)$ is stabilizable if and only if $(A^\top, B^\top)$ is detectable [82]; the detectability of $(A, B^\top)$ is seldom of interest in linear system theory; however, we show that it is indeed, a necessary condition for $(A, B)$ to be regularizable. To this end, we first connect regularizability to the spectral properties of the pair $(A, B)$.

**Lemma 21.** *Let $\widetilde{A} = \Pi_{\mathcal{R}(B)^\perp} A$. Then for each right eigenpair $(\lambda, \boldsymbol{v})$ of $A$ the following hold:*

- *$(\lambda, \boldsymbol{v})$ is a right eigenpair of $\widetilde{A}$ whenever $\boldsymbol{v} \in \mathcal{R}(B)^\perp$ or $\lambda = 0$.*
- *$(0, \boldsymbol{v})$ is a right eigenpair of $\widetilde{A}$ whenever $\boldsymbol{v} \in \mathcal{R}(B)$.*

The proof of Lemma 21 directly follows from the definitions and therefore omitted. Note that the above lemma does not address the scenario where $(\lambda, \boldsymbol{v})$ is an eigenpair of $A$, with $\lambda \neq 0$, and $\boldsymbol{v} = \boldsymbol{v}_1 + \boldsymbol{v}_2$, with nontrivial $\boldsymbol{v}_1 \in \mathcal{R}(B)$ and $\boldsymbol{v}_2 \in \mathcal{R}(B)^\perp$. The following example illustrates the although $\widetilde{A}$ is a product of matrix $A$ with an orthogonal projection operator (which is non-expansive), its spectral radius can be drastically different from $A$.

**Example 3.** *Consider the system in Example 1, where the identity off-diagonal elements of $A$ are replaced with 10, $\lambda_1 = 0.9$ and $\lambda_i = 0$ for all $i = 2, \dots, n$, and $B = \mathbb{1}$. It is straightforward to show that for all $n \geq 2$, $A$ is Schur stable with spectral radius of 0.9 while $\widetilde{A}$ is not, i.e., $(A, B)$ is not regularizable. In this case, in spite of $A$ being Schur stable, its operator norm is about 10. Furthermore, the spectral radius of $\tilde{A}$ would be 4.55 for $n = 2$ and increases to about 10 as $n$ increases. This results in a pathological behavior despite the fact that the system is originally stable, e.g., any infinite horizon closed-loop LQR controller for this system would demonstrate undesirable behavior —similar to Example 1— when*

*initialized from* $\boldsymbol{x}_0 = \mathbb{1}$.[8] *Finally, it is worth mentioning that the controllability matrix of this pair would be ill-conditioned in contrast to Example 1.*

The preceding example illustrates that even for a stable system, it is nontrivial to assert that state trajectories over a finite time horizon are "well-regulated." It is no surprise then that most of the recent works on data-guided control focus on contractible systems as they streamline composition rules and analysis for consecutive iterations [2, 96]. The following remark shows that regularizability is less restrictive than contractibility, while also ensuring regulation of state trajectories.

**Remark 11.** *A pair* $(A, B)$ *is said to be contractible if there exists a controller* $K$ *such that* $\|A - BK\| < 1$. *Noting that*

$$A - BK = \widetilde{A} + \ \Pi_{\mathcal{R}(B)}(A - BK),$$

*for any vector* $\boldsymbol{x} \in \mathbb{R}^n$,

$$\begin{aligned}\|\widetilde{A}\boldsymbol{x}\|^2 &= \|(A - BK)\boldsymbol{x}\|^2 - \| \ \Pi_{\mathcal{R}(B)}(A - BK)\boldsymbol{x}\|^2 \\ &\leq \|(A - BK)\boldsymbol{x}\|^2 \\ &\leq \|(A - BK)\|^2 \, \|\boldsymbol{x}\|^2.\end{aligned}$$

*This, in turn, implies that a contractible system is regularizable (as* $\|\widetilde{A}\| < 1$*). In particular, if the original system matrix* $A$ *is non-expansive (at least on the subspace* $A^{-1}\{\mathcal{R}(B)^{\perp}\}$*), then* $(A, B)$ *is regularizable.*

The following results further clarifies the relation between regularizable systems with their system theoretic twins.

**Proposition 5.** *If* $(A, B)$ *is regularizable, then*

- $(A, B)$ *is stabilizable, and*

[8] One practical remedy to this problem is to split the dynamics into multiple time-scales using, say, a sampling heuristics [110] However, time-scale separation often requires physical insights, non-trivial to identify in general [121], let alone for a system with unknown dynamics. We address time-scale identification from finite-time collected data as a potential future direction of our work.

- $(A, B^\top)$ *is detectable.*

*Proof.* For the first claim, note that $\widetilde{A} = A - \Pi_{\mathcal{R}(B)}A = A + BK$, where $K := -B^\dagger A$. Thus if $(A, B)$ is regularizable then $K$ is a stabilizing closed loop controller. For the second claim, we establish a contrapositive. Suppose that $(A, B^\top)$ is not detectable. Hence there exists a right eigenpair $(\lambda, \boldsymbol{v})$ of $A$, where $|\lambda| \geq 1$ and $\boldsymbol{v} \in \mathcal{N}(B^\top) = \mathcal{R}(B)^\perp$. Then, Lemma 21 implies that $(\lambda, \boldsymbol{v})$ must be a right eigenpair of $\widetilde{A}$. Since $|\lambda| \geq 1$, $\widetilde{A}$ is not Schur stable and therefore $(A, B)$ is not regularizable. □

Note that the consequents of Proposition 5 are equivalent whenever $A$ is symmetric, simply because detectability of $(A, B^\top)$ is the same as stabilizability of $(A^\top, B)$. Also, note that Proposition 5 equivalently states that for the system in equation (4.16) to be Schur stable (i.e., the original pair $(A, B)$ to be regularizable), the original system $A$ has to be stabilizable through $B$ and also detectable through $B^\top$. This provides a necessary condition for regularizability. The following counter-example underscores why the stabilizability of $(A, B)$ even when combined with detectability of $(A, B^\top)$, is not sufficient for regularizability.

**Example 4.** *Let the system matrices $A, B$ be as defined in Example 1 and consider the pair $(A_1, B_1) := (A + A^\top, B)$. By the structure of $A_1$, note that $(A_1, B_1)$ is controllable. Since $A_1$ is symmetric, $(A_1, B_1^\top)$ is also observable. By direct computation we observe that,*

$$\widetilde{A} = \Pi_{\mathcal{R}(B)^\perp}A = \begin{pmatrix} 2\lambda_1 & 1 & 0 & \dots & 0 \\ 1 & 2\lambda_2 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & & 0 \\ 0 & & 1 & 2\lambda_{n-1} & 1 \\ 0 & \dots & 0 & 0 & 0 \end{pmatrix}.$$

*Now if any of $\lambda_i$ for $i = 1, \dots, n-1$ is say, larger than $1/2$, then $\widetilde{A}$ would be unstable, implying that $(A_1, B_1)$ is not regularizable.*

In order to complete our understanding of regularizability, we provide several characterizations using LMIs.

**Proposition 6.** *Consider a pair $(A, B)$, and denote $\Pi_\perp := \Pi_{\mathcal{R}(B)^\perp}$. Then the following are equivalent:*

1. *The pair* $(A, B)$ *is regularizable.*

2. $\exists P \succ 0$ *such that* $\rho(A^\top \ \Pi_\perp P \ \Pi_\perp A P^{-1}) < 1$.

3. $\exists P \succ 0$ *such that* $\|P^{1/2} \ \Pi_\perp A P^{-1/2}\| < 1$.

4. $\exists P \succ 0$ *such that*
$$A^\top \ \Pi_\perp P \ \Pi_\perp A - P \prec 0.$$

5. $\exists W \succ 0$ *such that,*
$$\begin{pmatrix} W & \Pi_\perp A W \\ W A^\top \ \Pi_\perp & W \end{pmatrix} \succ 0.$$

6. $\exists P \succ 0$ *and* $G \in \mathbb{R}^{n\times n}$ *such that,*
$$\begin{pmatrix} P & A^\top \ \Pi_\perp G^\top \\ G \ \Pi_\perp A & G + G^\top - P \end{pmatrix} \succ 0.$$

7. $\exists P \succ 0$, *and* $G, H \in \mathbb{R}^{n\times n}$ *such that,*
$$\begin{pmatrix} GA + A^\top G^\top - P & A^\top H^\top - G \\ HA - G^\top & \Pi_\perp P \ \Pi_\perp - H - H^\top \end{pmatrix} \prec 0.$$

*Proof.* Noting that regularizability of $(A, B)$ is equivalent to Schur stability of $\Pi_\perp A$, the first four equivalences are direct consequences of Theorem 7.7.7 in [85]. By Schur complement and by constructing, following by a congruence induced by $\mathrm{diag}(I, P^{-1})$, (iv) becomes equivalent to (v). The last two equivalences are due to Theorem 1 in [47] and Theorem 1 in [48], respectively. □

We conclude this section by providing a sufficient condition for guaranteeing when an uncertain polytopic LTI system is regularizable.

**Proposition 7.** *Consider* $A_i \in \mathbb{R}^{n\times n}$ *for* $i = 1, \ldots, N$ *and suppose there exist matrices* $P_i \succ 0$ *and* $G, H \in \mathbb{R}^{n\times n}$ *satisfying,*
$$\begin{pmatrix} GA_i + A_i^\top G^\top - P_i & A_i^\top H^\top - G \\ HA_i - G^\top & \Pi_S P_i \ \Pi_S - H - H^\top \end{pmatrix} \prec 0,$$

*for some linear subspace of* $S \subseteq \mathbb{R}^n$. *Then a pair* $(A_\alpha, B)$ *is regularizable whenever* $A_\alpha \in$ convhull$\{A_i\}_1^N$ *and*

$$\begin{pmatrix} P_i & P_i\, \Pi_{\mathcal{R}(B)^\perp} \\ \Pi_{\mathcal{R}(B)^\perp} P_i & \Pi_S P_i\, \Pi_S \end{pmatrix} \succeq 0, \quad \forall i = 1, \ldots, N.$$

*Proof.* Since $A_\alpha \in \text{convhull}\{A_i\}_1^N$, there exists scalars $\alpha_i \in [0,1]$ with $\sum_1^N \alpha_i = 1$ such that $A_\alpha = \sum_1^N \alpha_i A_i$. By defining $P_\alpha = \sum_1^N \alpha_i P_i$ and taking the convex combinations of the negative definite matrices in the hypothesis with weights $\alpha_i$ we get that,

$$\begin{pmatrix} GA_\alpha + A_\alpha^\top G^\top - P_\alpha & A_\alpha^\top H^\top - G \\ HA_\alpha - G^\top & \Pi_S P_\alpha\, \Pi_S - H - H^\top \end{pmatrix} \prec 0. \tag{4.17}$$

Now by taking the Schur complement of the LMIs in the hypothesis involving the input matrix $B$ we obtain,

$$\Pi_S P_i\, \Pi_S \succeq\ \Pi_{\mathcal{R}(B)^\perp} P_i\, \Pi_{\mathcal{R}(B)^\perp}, \quad \forall i = 1, \ldots, N.$$

Convex combinations of these LMIs now lead to,

$$\Pi_S P_\alpha\, \Pi_S \succeq\ \Pi_{\mathcal{R}(B)^\perp} P_\alpha\, \Pi_{\mathcal{R}(B)^\perp}.$$

This, together with the LMI equation (4.17) imply that

$$\begin{pmatrix} GA_\alpha + A_\alpha^\top G^\top - P_\alpha & A_\alpha^\top H^\top - G \\ HA_\alpha - G^\top & \Pi_{\mathcal{R}(B)^\perp} P_\alpha\, \Pi_{\mathcal{R}(B)^\perp} - H - H^\top \end{pmatrix} \prec 0.$$

Noting that $P_\alpha \succ 0$, by part (vii) of Proposition 6, we conclude that the pair $(A_\alpha, B)$ is regularizable. □

Note that the proof above also shows that the last LMI in the statement of Proposition 7 is equivalent to

$$\Pi_S P_i\, \Pi_S \succeq\ \Pi_{\mathcal{R}(B)^\perp} P_i\, \Pi_{\mathcal{R}(B)^\perp}, \quad \forall i = 1, \ldots, N;$$

this is certainly satisfied when $S = \mathcal{R}(B)^\perp$. In particular, Proposition 7 implies that regularlizability is a monotonic system theoretic property with respect to the input, in the sense that enlarging $\mathcal{R}(B)$ would not destroy its regularizability. In fact, enlarging $\mathcal{R}(B)$ for a system would make the system more regularizable (as $\widetilde{A}$ will have smaller modulus eigenvalues).

### *4.3.2 Data-Guided Regulation (DGR) Algorithm*

The primary focus of the present work is devising an online, data-driven feedback controller to regulate the system state trajectories, quantified in terms of some signal norm. In this direction, we propose an iterative algorithm for updating the feedback gain (policy); the form of the controller is motivated by considering, at each time-step $t$, the optimization problem,[9]

$$\begin{aligned} \min_{\boldsymbol{u}_t} \quad & \|\boldsymbol{x}_{t+1}\|^2 + \alpha\|\boldsymbol{u}_t\|^2 \\ \text{s.t.} \quad & \boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + B\boldsymbol{u}_t, \end{aligned} \tag{4.18}$$

where $\boldsymbol{x}_t$ can be measured over time but the system matrix is unknown, and $\alpha \geq 0$ is a regularization factor of the controller to be designed.[10] In the case of known $A$, it is straightforward to characterize the set of minimizers of the above optimization problem through the first order optimality condition,

$$(\alpha I_m + B^\top B)\boldsymbol{u}_t + B^\top A\boldsymbol{x}_t = 0;$$

as such the corresponding input belongs to a linear subspace in $\mathbb{R}^m$ parameterized by the system matrices and data. This observation forms the basis for the proposed algorithm when $A$ is unknown and potentially unstable; essentially, we aim to control the system in an online fashion to become more regulated, using the history of the state data and a regulated input. The corresponding synthesis procedure is detailed in Algorithm 6. Specifically, at time-step $t$, DGR sets

$$\boldsymbol{u}_t^* = -K_t^*\boldsymbol{x}_t, \qquad K_t^* := G_\alpha \mathcal{Y}_t \mathcal{X}_{t-1}^\dagger, \tag{4.19}$$

where $\mathcal{X}_{t-1}, \mathcal{Y}_t \in \mathbb{R}^{n\times t}$ are the measured data matrices with $\mathcal{Y}_t = A\mathcal{X}_{t-1}$, and $G_\alpha = (\alpha I + B^\top B)^\dagger B^\top$ for $\alpha \geq 0$. Note that $G_0 = (B^\top B)^\dagger B^\top = B^\dagger$; on the other hand, $G_\alpha \to 0$ as $\alpha \to \infty$ implying that $\boldsymbol{u}_t \to 0$ for all $t$. Intuitively, collecting more data results in

[9] The setup resembles dead-beat control design, with the caveat that the synthesis is data-guided.

[10] We note that assuming a more elaborate form of cost (e.g., finite/infinite horizon LQR cost) for this optimization problem would not be advantageous as $A$ is assumed to be unknown.

[11] The stopping criterion can be application specific. For instance, for sysID generating $n$ linearly independent data is sufficient, while mere stabilization may require less; see [164]

**Algorithm 6** Data-Guided Regulation (DGR)

1: **Initialization** (at $t = 0$)
2: Measure $\boldsymbol{x}_0$; set $K_0 = \mathbf{0}$, $G_\alpha = (\alpha I + B^\top B)^\dagger B^\top$
3: Set $\mathcal{X}_0 = [\boldsymbol{x}_0]$ and $\mathcal{Y}_0 = [\,]$
4: **While stopping criterion not met**[11]
5: Compute $\boldsymbol{u}_t = -K_t \boldsymbol{x}_t$
6: Run system (4.15) and measure $\boldsymbol{x}_{t+1}$
7: Update $\mathcal{Y}_{t+1} = [\mathcal{Y}_t \quad \boldsymbol{x}_{t+1} - B\boldsymbol{u}_t]$
8: $K_{t+1} = G_\alpha \mathcal{Y}_{t+1} \mathcal{X}_t^\dagger$
9: $\mathcal{X}_{t+1} = [\mathcal{X}_t \quad \boldsymbol{x}_{t+1}]$
10: $t = t + 1$

capturing the essential (*e.g.*, unstable) modes in the dynamics. As such, it is important to note that DGR is particularly relevant for online regulation of an unstable system, when the controller does not have access to enough state data for the purpose of system identification or optimal control synthesis. Note that DGR actively guides the ongoing process, and the data generation and the dynamics are interdependent; this setup would therefore not be suited for traditional system identification. The proposed technique is close in spirit to modal analysis where regression-based methods are leveraged to extract and control the dominant modes of the system [133, 146]. The emphasis of DGR, however, is on the significance of each temporal action for safety-critical applications; in these scenarios, it might be rather unrealistic to generate enough data from the inherent unstable modes.

From an implementation perspective, the DGR algorithm can become computationally expensive for large systems. The reason lies within steps 7-9 of Algorithm 6 where the entire history of data is stored in $\mathcal{X}_{t+1}$ and $\mathcal{Y}_{t+1}$; the pseudoinverse operation in the meantime has complexity $\mathcal{O}(n^2 t)$ required at iteration $t$. While for the purpose of analysis, we present the basic form of DGR (as in Algorithm 6), in the upcoming sections we will propose Fast Data-Guided Regulation (F-DGR) to circumvent the complexity of storing and computing on large datasets using rank-one update on the data matrices, resulting in a recursive

evaluation of $\mathcal{Y}_t\mathcal{X}_{t-1}^{\dagger}$.

### *4.3.3 Analysis of DGR*

In this section, we provide performance analysis for DGR. As pointed out previously, DGR is particularly relevant when $t \leq n$, where $n$ denotes the dimension of the underlying system. Subsequently, we examine the effects of DGR on the system state trajectory when $\mathcal{R}(A) \subset \mathcal{R}(B)$. In addition, we will see how a particular structure of the system matrix $A$, such as its diagonalizability, further facilitates deriving more insights into the operation of DGR.

First, we show why regularizability is essential for the analysis of the trajectory generated under Algorithm 6; in hindsight, justifying its introduction in the first place.

**Lemma 22.** *For all $t > 0$, the trajectory generated by Algorithm 6 satisfies,*

$$\boldsymbol{x}_{t+1} = \ \Pi_{\mathcal{R}(B)^{\perp}} A\boldsymbol{x}_t + \ \Pi_{\mathcal{R}(B)}\ A\boldsymbol{z}_t + \Delta_{\alpha}\boldsymbol{w}_t$$

*where $\Delta_{\alpha} \coloneqq B(B^{\dagger} - G_{\alpha})A$, $\boldsymbol{z}_0 \coloneqq \boldsymbol{x}_0$, $\boldsymbol{w}_0 = 0$, and $\boldsymbol{z}_t \coloneqq \ \Pi_{\mathcal{R}(\mathcal{X}_{t-1})^{\perp}}\boldsymbol{x}_t$ , $\boldsymbol{w}_t \coloneqq \ \Pi_{\mathcal{R}(\mathcal{X}_{t-1})}\boldsymbol{x}_t$ . Furthermore, $\{\boldsymbol{z}_0, \boldsymbol{z}_1, \cdots, \boldsymbol{z}_t\}$ is a set of orthogonal vectors, and $\Delta_0 = 0$.*

*Proof.* Let $B = U_r\Sigma_r V_r^{\top}$ be the "thin" SVD of $B$ where $r = \text{rank}(B)$. Since $BB^{\dagger} = U_r U_r^{\top} = \Pi_{\mathcal{R}(U_r)}$,

$$\begin{aligned}
\boldsymbol{x}_{t+1} &= A\boldsymbol{x}_t + B\boldsymbol{u}_t \\
&= \left[A - BG_{\alpha}\mathcal{Y}_t\mathcal{X}_{t-1}^{\dagger}\right]\boldsymbol{x}_t \\
&= \left[A - \ BG_{\alpha}\ A\ \ \Pi_{\mathcal{R}(\mathcal{X}_{t-1})}\right]\boldsymbol{x}_t \\
&= \ \Pi_{\mathcal{R}(U_r)^{\perp}} A\boldsymbol{x}_t + BB^{\dagger}A(\boldsymbol{z}_t + \boldsymbol{w}_t) - BG_{\alpha}\ A\ \boldsymbol{w}_t \\
&= \ \Pi_{\mathcal{R}(U_r)^{\perp}} A\boldsymbol{x}_t + \ \Pi_{\mathcal{R}(U_r)}\ A\boldsymbol{z}_t + B(B^{\dagger} - G_{\alpha})A\boldsymbol{w}_t.
\end{aligned}$$

Thus, the first claim follows as $\mathcal{R}(U_r) = \mathcal{R}(B)$. For the second claim, note that the definition of $\boldsymbol{z}_t$ implies that $\boldsymbol{z}_t \perp \mathcal{R}(\mathcal{X}_{t-1})$, and $\boldsymbol{z}_k \in \mathcal{R}(\mathcal{X}_{t-1})$ for all $k = 1, \ldots, t-1$. Hence $\{\boldsymbol{z}_0, \boldsymbol{z}_1, \cdots, \boldsymbol{z}_t\}$ consists of orthogonal vectors. Finally, by definition of $G_{\alpha}$ it follows that $\Delta_0 = 0$. □

The preceding lemma implies that in the case of $\alpha = 0$, the time series generated by Algorithm 6 can be considered as the trajectory of a linear system with parameters $(\widetilde{A}, \widetilde{B})$ and input $\boldsymbol{z}_t$ where,

$$\widetilde{A} \coloneqq \Pi_{\mathcal{R}(B)^{\perp}} A, \qquad \widetilde{B} \coloneqq \Pi_{\mathcal{R}(B)} A, \tag{4.20}$$

and $\boldsymbol{z}_t = \widetilde{K}_t \boldsymbol{x}_t$, with the time-varying, state-dependent feedback gain $\widetilde{K}_t = \Pi_{\mathcal{R}(\mathcal{X}_{t-1})^{\perp}}$.

Note that, in this case, if $\Pi_{\mathcal{R}(B)}$ and $A$ commute,[12] then $\widetilde{A}\widetilde{B} = 0$ and $\boldsymbol{x}_{t+1} = \widetilde{A}^{t+1}\boldsymbol{x}_0 + \widetilde{B}\boldsymbol{z}_t$. Moreover, $\widetilde{A} = 0$ whenever $\mathcal{R}(A) \subset \mathcal{R}(B)$, i.e., the system dynamics will only be driven by the feedback signal $\boldsymbol{z}_t$; this case will be examined subsequently. In case of general $\alpha$, the system trajectories evolve as,

$$\boldsymbol{x}_{t+1} = \widetilde{A}^{t+1}\boldsymbol{x}_0 + \sum_{r=0}^{t} \widetilde{A}^{t-r} \left[ \widetilde{B}\boldsymbol{z}_r + \Delta_{\alpha}\boldsymbol{w}_r \right]. \tag{4.21}$$

Finally, an attractive feature of DGR hinges upon the orthogonality of the "hidden" states $\boldsymbol{z}_t$ generated during the process.

*Bounding the State Trajectories Generated by DGR*

In the open loop setting, the generated data from an unstable system can grow exponentially fast with a rate dictated by the largest unstable mode. We show that DGR can prevent this undesirable phenomenon for unstable systems when the system is regularizable. The key property for such an analysis involves the notion of the instability number.

**Definition 8.** *Given the matrix $A \in \mathbb{R}^{n\times n}$, its instability number of order $t$ is defined as,*

$$M_t(A) \coloneqq \sup_{\{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_t\}\in\mathcal{O}_t^n} \|A\boldsymbol{v}_1\| \, \|A\boldsymbol{v}_2\| \, \cdots \, \|A\boldsymbol{v}_t\|, \tag{4.22}$$

*where $\mathcal{O}_t^n$ is the collection of all sets of $t$ orthonormal vectors in $\mathbb{R}^n$.*

Note that $M_t(A) \leq \|A\|^t$, where $\|.\|$ denotes the induced operator norm. In general, the instability number of a matrix is distinct from products of any subset of its eigenvalues. Consider for example, a $t$-dimensional hypercube in the domain with its image as a

[12] This is the case if (and only if) both matrices are simultaneously diagonalizable (Theorem 1.3.21 in [85]). If $A$ is symmetric, then these matrices commute if (and only if) they are congruent (Theorem 4.5.15 in [85]).

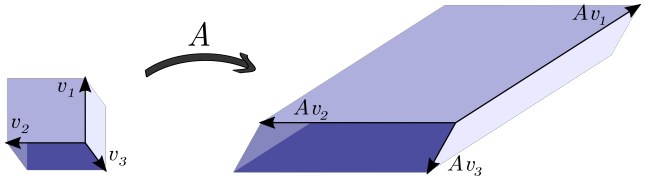



Figure 4.3: A unit cube in the domain of $A$ that is mapped to a parallelepiped in its range space.

parallelotope (see Figure 4.3 for a 3D schematic). The instability number is related to the multiplication of the lengths of edges radiating from one vertex of the parallelotope, while $\det(A^\top A)$ (the multiplication of eigenvalues) is related to its volume. In fact, the instability number of a matrix might be difficult to compute. In what follows, we first provide upper and lower bounds on $M_t(A)$ characterizing its growth rate with respect to the largest singular value of $A$. Subsequently, these bounds will be used to provide a bound on the norm of the state trajectory generated by DGR.

**Lemma 23.** *Let $\sigma_1, \cdots, \sigma_n$ denote the singular values of $A \in \mathbb{R}^{n\times n}$ in a descending order. Then for $t \leq n$,*

$$\left[\frac{\sigma_1^2}{t}\right]^t \leq M_t^2(A) \leq \left[\frac{\sigma_1^2}{t}\right]^t + \sum_{j=1}^{t-1} \left[\frac{\sigma_1^2}{t-j}\right]^{t-j} \binom{t}{j} \delta^j + \delta^t,$$

*where $\delta \coloneqq \sum_{i=2}^t \sigma_i^2$, with $M_t(A)$ as in Definition 8.*

*Proof.* Let $A = W\Sigma U^\top$ be the SVD of $A$ where $\Sigma$ is diagonal containing the singular values in descending order and both $W, U \in \mathbb{R}^{n\times n}$ are unitary. This implies that,

$$\begin{aligned} M_t(A) &= \sup_{\{\boldsymbol{v}_i\}_1^t \in \mathcal{O}_t^n} \|\Sigma U^\top \boldsymbol{v}_1\| \, \|\Sigma U^\top \boldsymbol{v}_2\| \, \cdots \, \|\Sigma U^\top \boldsymbol{v}_t\| \\ &= \sup_{\{\boldsymbol{v}_i\}_1^t \in \mathcal{O}_t^n} \|\Sigma \boldsymbol{v}_1\| \, \|\Sigma \boldsymbol{v}_2\| \, \cdots \, \|\Sigma \boldsymbol{v}_t\|, \end{aligned}$$

where the last equality is due to the fact that $\{U^\top \boldsymbol{v}_i\}_1^t \in \mathcal{O}_t^n$ only if $\{\boldsymbol{v}_i\}_1^t \in \mathcal{O}_t^n$, since $U$ is unitary. For the lower-bound if $t \leq n$, we can choose $\{\boldsymbol{v}_i\}_1^t \in \mathcal{O}_t^n$ such that $|\langle \boldsymbol{e}_1, \boldsymbol{v}_i\rangle| = 1/\sqrt{t}$ for all $i = 1, \cdots, t$. This choice is certainly possible as a result of applying Parseval's identity in a $t$-dimensional subspace with orthonormal basis $\{\boldsymbol{v}_i\}_1^t$ containing the unit vector $\boldsymbol{e}_1$, in

which, $\boldsymbol{e}_1$ is represented with all coordinates equal to $1/\sqrt{t}$ with respect to this basis. We thus conclude that,

$$M_t(A) \geq |\sigma_1 \langle \boldsymbol{e}_1, \boldsymbol{v}_1 \rangle| \cdots |\sigma_1 \langle \boldsymbol{e}_1, \boldsymbol{v}_t \rangle| \geq \left[ \frac{\sigma_1}{\sqrt{t}} \right]^t,$$

where the left inequality follows from the fact that $\|\Sigma \boldsymbol{v}\| \geq |\sigma_1 \langle \boldsymbol{e}_1, \boldsymbol{v} \rangle|$ for any $\boldsymbol{v} \in \mathbb{R}^n$. For the upper-bound, define $\Sigma_t = \mathrm{diag}(\sigma_1, \ldots, \sigma_t)$ and since singular values are in descending order we have,

$$\begin{aligned} M_t(A) &\leq \sup_{\{\boldsymbol{v}_i\}_1^t \in \mathcal{O}_t^t} \prod_{i=1}^t \|\Sigma_t \boldsymbol{v}_i\| \\ &= \sup_{\{\boldsymbol{v}_i\}_1^t \in \mathcal{O}_t^t} \prod_{i=1}^t \left[ \sigma_1^2 \left| \langle \boldsymbol{e}_1, \boldsymbol{v}_i \rangle \right|^2 + \sum_{j=2}^t \left| \sigma_j \langle \boldsymbol{e}_j, \boldsymbol{v}_i \rangle \right|^2 \right]^{\frac{1}{2}} \\ &\leq \sup_{\{\boldsymbol{v}_i\}_1^t \in \mathcal{O}_t^t} \prod_{i=1}^t \left[ \sigma_1^2 \left| \langle \boldsymbol{e}_1, \boldsymbol{v}_i \rangle \right|^2 + \delta \right]^{\frac{1}{2}}. \end{aligned}$$

Define $\gamma_i = \langle \boldsymbol{e}_1, \boldsymbol{v}_i \rangle$; then by Bessel's inequality $\sum_{i=1}^t \gamma_i^2 \leq 1$ whenever $\{\boldsymbol{v}_i\}_1^t \in \mathcal{O}_t^t$. Thereby, by denoting $\boldsymbol{\gamma} \coloneqq [\gamma_1 \ \ldots \ \gamma_t]^\top$, we can conclude that

$$\begin{aligned} M_t(A) &\leq \sup_{\boldsymbol{\gamma} \in \mathcal{B}_2^t} \prod_{i=1}^t \left[ \sigma_1^2 \gamma_i^2 + \delta \right]^{\frac{1}{2}} \\ &= \sup_{\boldsymbol{\gamma} \in \mathcal{B}_2^t} \left[ \sum_{i=1}^{t+1} \sigma_1^{2(t+1-i)} \, \delta^{i-1} \sum_{\substack{|\alpha| = t+1-i \\ \alpha \in \mathcal{B}_\infty^t}} (\gamma_1^2)^{\alpha_1} \cdots (\gamma_t^2)^{\alpha_t} \right]^{\frac{1}{2}}, \end{aligned}$$

where $\alpha$ is a multi-index of dimension $t$, and the last equality follows by direct computation. Now it is easy to see that for fixed $\alpha$, if $|\alpha| = m > 0$ and $\alpha \in \mathcal{B}_\infty^t$ then,

$$\sup_{\boldsymbol{\gamma} \in \mathcal{B}_2^t} (\gamma_1^2)^{\alpha_1} \cdots (\gamma_t^2)^{\alpha_t} \leq (\frac{1}{m})^m,$$

that follows by the symmetry in optimization variables. Therefore, we can conclude that

$$M_t(A) \leq \left[ \delta^t + \sum_{i=1}^t \left[ \frac{\sigma_1^2}{t+1-i} \right]^{(t+1-i)} \delta^{i-1} \binom{t}{t+1-i} \right]^{\frac{1}{2}},$$

implying the claimed upper bound. □

The lower and upper bounds in Lemma 23 show that, particularly when $\delta < 1$, $M_t(A)$ would initially grow similar to $(\sigma_1/\sqrt{k})^k$ for $k \leq t$, in contrast to the exponential growth $\sigma_1^k$. The following result provides an upper bound for the most general case.

**Theorem 25.** *For any pair $(A, B)$, the trajectory generated by Algorithm 6 satisfies,*

$$\|\boldsymbol{x}_{t+1}\| \leq L_{t+1}\|\boldsymbol{x}_0\|\,,$$

*where $L_t$ satisfies the recursion,*

$$L_{t+1} = a_t + \sum_{r=1}^{t} b_{t,r} L_r, \qquad L_1 = \|A\overline{\boldsymbol{z}}_0\|,$$

*with*

$$b_{t,r} = \sqrt{\|\widetilde{A}^{t-r}\widetilde{B}\overline{\boldsymbol{z}}_r\|^2 + \|\widetilde{A}^{t-r}\Delta_\alpha\overline{\boldsymbol{w}}_r\|^2}$$

*and $a_t = \|\widetilde{A}^t A\overline{\boldsymbol{z}}_0\|$, where $\overline{\boldsymbol{z}}_r = \boldsymbol{z}_r/\|\boldsymbol{z}_r\|$ (if $\boldsymbol{z}_r \neq 0$, otherwise $\overline{\boldsymbol{z}}_r = 0$), and $\overline{\boldsymbol{w}}_r$ is similarly defined.*

*Proof.* Knowing that $\boldsymbol{x}_1 = A\boldsymbol{x}_0$, it follows that $\|\boldsymbol{x}_1\| \leq L_1\|\boldsymbol{x}_0\|$. Furthermore, for $t \geq 1$, equation (4.21) leads to,

$$\boldsymbol{x}_{t+1} = \widetilde{A}^t A\boldsymbol{x}_0 + \textstyle\sum_{r=1}^{t} \widetilde{A}^{t-r}\left[\widetilde{B}\boldsymbol{z}_r + \Delta_\alpha \boldsymbol{w}_r\right],$$

since $\widetilde{A} + \widetilde{B} = A$ and $\boldsymbol{w}_0 = 0$. This implies that,

$$\begin{aligned}\|\boldsymbol{x}_{t+1}\| &\leq \|\widetilde{A}^t A\boldsymbol{x}_0\| + \sum_{r=1}^{t} \|\widetilde{A}^{t-r}\widetilde{B}\overline{\boldsymbol{z}}_r\|\|\boldsymbol{z}_r\| + \|\widetilde{A}^{t-r}\Delta_\alpha\overline{\boldsymbol{w}}_r\|\|\boldsymbol{w}_r\|,\\ &\leq a_t\|\boldsymbol{x}_0\| + \sum_{r=1}^{t} b_{t,r}\|\boldsymbol{x}_r\|,\end{aligned}$$

where we have used Cauchy–Schwarz inequality in conjunction with the equality $\|\boldsymbol{z}_r\|^2 + \|\boldsymbol{w}_r\|^2 = \|\boldsymbol{x}_r\|^2$. Using this recursive bound, the rest of the proof follows by induction. □

**Remark 12.** *Note that in the analysis above, when the system is regularizable, $a_t$ eventually decreases exponentially fast as $t$ increases. Furthermore, the term $b_{t,r}$ in the sum increases as $r$ approaches a fixed $t$. Additionally in the case of $\alpha = 0$, since $b_{r,r} = \|\widetilde{B}\overline{\boldsymbol{z}}_r\| \leq \|A\overline{\boldsymbol{z}}_r\|$, for any set $T = \{r_i\}_1^n \subset \mathbb{N}$,*

$$b_{r_n,r_n} b_{r_{n-1},r_{n-1}} \cdots\ b_{r_1,r_1} \leq M_n(A).$$

*Finally, we can show that the obtained upper bound is tight by considering Example 1 with* $\lambda_1 > 0$ *and* $\lambda_i = 0$ *for* $i > 1$.

In order to shed light on the intuition behind the upper bound in Theorem 25, we next study simpler cases with $\alpha = 0$, where there exists small enough $\kappa$ for which $b_{t,r} \leq \|\widetilde{A}^{t-r}\widetilde{B}\| \leq \kappa$ for all $r < t$. In particular, we can show that if the system is regularizable and $\widetilde{A}\widetilde{B} = 0$, then the trajectories of the closed loop system will be bounded by a combination of instability number of different orders. This is stated in the next corollary; the proof is omitted for brevity.

**Corollary 2.** *For any regularizable pair* $(A, B)$ *with* $\widetilde{A}\widetilde{B} = 0$*, and* $M_t(A)$ *as in Definition 8, the system trajectory generated by Algorithm 6 with* $\alpha = 0$ *satisfies,*

$$\frac{\|\boldsymbol{x}_{t+1}\|}{\|\boldsymbol{x}_0\|} \leq M_{t+1}(A) + a_t + \sum_{r=1}^{t-1} M_r(A) a_{t-r}\,.$$

The above observation further highlights the importance of the instability number in the context of DGR.

### *4.3.4 Informativity of the DGR Generated Data*

In the sequel, we show that DGR generates linearly independent data; we then proceed to make a connection between this independence structure and the number of excited modes in the system. Before we proceed, let us define $L_k^t(A)$, that is based on $k$ modes of a matrix $A$, as,

$$L_k^t(A) \coloneqq \begin{pmatrix} 1 & \lambda_1 & \cdots & (\lambda_1)^{t-1} \\ 1 & \lambda_2 & \cdots & (\lambda_2)^{t-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_k & \cdots & (\lambda_k)^{t-1} \end{pmatrix}, \quad 1 \leq t \leq n. \tag{4.23}$$

**Remark 13.** *Note that* $L_k^t(A)$ *has a specific structure that hints at its invertibility. In fact, for* $t = k$*,* $L_k^k(A)$ *is the Vandermonde matrix formed by* $k$ *eigenvalues of* $A$ *which would be invertible if and only if* $\lambda_1, \cdots, \lambda_k$ *are distinct. More generally, if* $\{\lambda_1, \cdots, \lambda_k\}$ *consists of* $r$ *distinct eigenvalues (where* $r \leq k$*), then* $L_k^r(A)$ *is a full column rank.*

Note that informativity implies that the available data is sufficient for regulating the state trajectory of the system initiated from $\boldsymbol{x}_0$. In particular, if $\boldsymbol{x}_0$ results in all the modes of $A$ being excited, one should be able to conclude that the generated data is informative for regulation. This can be formalized as follows.

**Theorem 26.** *Let $\boldsymbol{x}_0$ excite $k_1 + k_2$ modes of $A$, such that $k_1$ modes are in $\mathcal{R}(B)$ and $k_2$ modes are in $\mathcal{R}(B)^\perp$. If $\alpha = 0$ and the excited modes correspond with distinct eigenvalues, then $\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{r-1}\}$, generated by Algorithm 6, is a set of linearly independent vectors for any $r \leq \max\{k_1, k_2\}$.*

*Proof.* Without loss of generality, let $\lambda_1, \ldots, \lambda_{k_1}$ be the eigenvalues corresponding to the excited modes $u_1, \ldots, u_{k_1} \in \mathcal{R}(B)$, and similarly $\lambda_{k_1+1}, \ldots, \lambda_{k_1+k_2}$ be corresponding to $u_{k_1+1}, \ldots, u_{k_1+k_2} \in \mathcal{R}(B)^\perp$. Recall that $\mathcal{X}_{t-1} = [\boldsymbol{x}_0 \ \boldsymbol{x}_1 \ \ldots \ \boldsymbol{x}_{t-1}]$; then by definition of $\boldsymbol{z}_t$ in Lemma 22, for $t \geq 1$ there exists scalar coefficients $\zeta_0^t, \cdots, \zeta_{t-1}^t \in \mathbb{R}$ such that $\boldsymbol{z}_t = \boldsymbol{x}_t - \sum_{j=0}^{t-1} \zeta_j^t \boldsymbol{x}_j$. This together with the dynamics in Lemma 22 imply that $\boldsymbol{x}_1 = A\boldsymbol{x}_0$ and for $t \geq 2$,

$$\boldsymbol{x}_t = A\boldsymbol{x}_{t-1} - \Pi_{\mathcal{R}(B)} \sum_{j=0}^{t-2} \zeta_j^t A\boldsymbol{x}_j. \tag{4.24}$$

Since $\boldsymbol{x}_0$ excites $k_1 + k_2$ modes of the system, we have $\boldsymbol{x}_0 = \sum_{\ell=1}^{k_1+k_2} \beta_\ell \boldsymbol{u}_\ell$, where $\beta_\ell$ are some nonzero real coefficients and $(\lambda_\ell, \boldsymbol{u}_\ell)$ are eigenpairs of $A$. Hence $\boldsymbol{x}_1 = A\boldsymbol{x}_0 = \sum_{\ell=1}^{k_1+k_2} \beta_\ell \lambda_\ell \boldsymbol{u}_\ell$, and we claim that for $t \geq 2$ there exist scalar coefficients $\xi_1^t, \cdots, \xi_{t-1}^t \in \mathbb{R}$ such that,

$$\boldsymbol{x}_t = \sum_{\ell=1}^{k_1} \beta_\ell \Big[ (\lambda_\ell)^t - \sum_{i=1}^{t-1} \xi_i^t (\lambda_\ell)^i \Big] \boldsymbol{u}_\ell + \sum_{\ell=k_1+1}^{k_1+k_2} \beta_\ell (\lambda_\ell)^t \boldsymbol{u}_\ell. \tag{4.25}$$

The proof of the last claim is by induction. Note that $A^t\boldsymbol{x}_0 = \sum_{\ell=1}^{k_1+k_2} \beta_\ell (\lambda_\ell)^t \boldsymbol{u}_\ell$, and by substituting this into equation (4.24) for $t = 2$ we have that,

$$\begin{aligned}
\boldsymbol{x}_2 &= A\boldsymbol{x}_1 - \zeta_0^2 \Pi_{\mathcal{R}(B)} A\boldsymbol{x}_0 \\
&= \Pi_{\mathcal{R}(B)} \left[ A^2\boldsymbol{x}_0 - \zeta_0^2 A\boldsymbol{x}_0 \right] + \Pi_{\mathcal{R}(B)^\perp} A^2\boldsymbol{x}_0 \\
&= \textstyle\sum_{\ell=1}^{k_1} \beta_\ell \left[ (\lambda_\ell)^2 - \zeta_0^2 \lambda_\ell \right] \boldsymbol{u}_\ell + \sum_{\ell=k_1+1}^{k_1+k_2} \beta_\ell (\lambda_\ell)^2 \boldsymbol{u}_\ell,
\end{aligned}$$

where the last equality is due to the fact that $\boldsymbol{u}_\ell \in \mathcal{R}(B)$ for $\ell \leq k_1$ and $\boldsymbol{u}_\ell \in \mathcal{R}(B)^\perp$ for $\ell > k_1$. By choosing $\xi_1^2 = \zeta_0^2$, we have shown that equation (4.25) holds for $t = 2$. Now

suppose that equation (4.25) holds for all $2, \ldots, t-1$; it now suffices to show that this relation also holds for $t$. By substituting the hypothesis for $2, \ldots, t-1$ into equation (4.24),

$$\begin{aligned}\boldsymbol{x}_t = &\sum_{\ell=1}^{k_1} \beta_\ell \Big[ (\lambda_\ell)^t - \sum_{i=1}^{t-2} \xi_i^{t-1} (\lambda_\ell)^{i+1} \Big] \boldsymbol{u}_\ell + \sum_{\ell=k_1+1}^{k_1+k_2} \beta_\ell (\lambda_\ell)^t \boldsymbol{u}_\ell \\ &- \sum_{\ell=1}^{k_1} \beta_\ell \zeta_0^t \lambda_\ell \boldsymbol{u}_\ell - \sum_{\ell=1}^{k_1} \beta_\ell \zeta_1^t (\lambda_\ell)^2 \boldsymbol{u}_\ell \\ &- \sum_{j=2}^{t-2} \zeta_j^t \sum_{\ell=1}^{k_1} \beta_\ell \Big[ (\lambda_\ell)^{j+1} - \sum_{i=1}^{j-1} \xi_i^j (\lambda_\ell)^{i+1} \Big] \boldsymbol{u}_\ell.\end{aligned}$$

Therefore,

$$\boldsymbol{x}_t = \textstyle\sum_{\ell=1}^{k_1} \beta_\ell \, [\star] \, \boldsymbol{u}_\ell + \sum_{\ell=k_1+1}^{k_1+k_2} \beta_\ell (\lambda_\ell)^t \boldsymbol{u}_\ell,$$

where $\star$ replaces the expression,

$$(\lambda_\ell)^t - \sum_{i=1}^{t-2} \xi_i^{t-1} (\lambda_\ell)^{i+1} - \sum_{j=0}^{t-2} \zeta_j^t (\lambda_\ell)^{j+1} + \sum_{j=2}^{t-2} \sum_{i=1}^{j-1} \zeta_j^t \xi_i^j (\lambda_\ell)^{i+1}.$$

By appropriate choices of $\xi_1^t, \cdots, \xi_{t-1}^t \in \mathbb{R}$, we can rewrite $\star = (\lambda_\ell)^t - \sum_{i=1}^{t-1} \xi_i^t (\lambda_\ell)^i$. This completes the proof of the claim in equation (4.25) by induction. Now, let $\widehat{\boldsymbol{x}} = \sum_{j=0}^{r-1} \alpha_j \boldsymbol{x}_j$ for some $\alpha_j \in \mathbb{C}$ and some $r \leq \max\{k_1, k_2\}$. Then, by substituting $\boldsymbol{x}_j$ from (4.25) and exchanging the sums over $j$ and $\ell$ we have,

$$\begin{aligned}\widehat{\boldsymbol{x}} = &\sum_{\ell=1}^{k_1} \beta_\ell \left[ \alpha_0 + \alpha_1 \lambda_\ell + \sum_{j=2}^{r-1} \alpha_j \Big[ (\lambda_\ell)^j - \sum_{i=1}^{j-1} \xi_i^j (\lambda_\ell)^i \Big] \right] \boldsymbol{u}_\ell \\ &+ \sum_{\ell=k_1+1}^{k_1+k_2} \beta_\ell \sum_{j=0}^{r-1} \alpha_j (\lambda_\ell)^j \boldsymbol{u}_\ell.\end{aligned}$$

Now, by exchanging the sums over $i$ and $j$, it follows that,

$$\begin{aligned}\widehat{\boldsymbol{x}} = &\sum_{\ell=1}^{k_1} \beta_\ell \left[ \alpha_0 + \sum_{i=1}^{r-2} \Big[ \alpha_i - \sum_{j=i+1}^{r-1} \alpha_j \xi_i^j \Big] (\lambda_\ell)^i + \alpha_{r-1} (\lambda_\ell)^{r-1} \right] \boldsymbol{u}_\ell \\ &+ \sum_{\ell=k_1+1}^{k_1+k_2} \beta_\ell \Big[ \sum_{j=0}^{r-1} \alpha_j (\lambda_\ell)^j \Big] \boldsymbol{u}_\ell.\end{aligned}$$

Since $\{\boldsymbol{u}_\ell\}_1^{k_1+k_2}$ are eigenvectors associated with distinct eigenvalues, they are linearly independent. Thus, noting that $\beta_\ell \neq 0$ for all $\ell = 1, \cdots, k_1 + k_2$, then $\widehat{\boldsymbol{x}} = 0$ implies that,

$$\alpha_0 + \sum_{i=1}^{r-2} \Big[\alpha_i - \sum_{j=i+1}^{r-1} \alpha_j \xi_i^j\Big](\lambda_\ell)^i + \alpha_{r-1}(\lambda_\ell)^{r-1} = 0,$$

for all $\ell = 1, \ldots, k_1$; and $\sum_{j=0}^{r-1} \alpha_j (\lambda_\ell)^j = 0$, for all $\ell = k_1 + 1, \ldots, k_1 + k_2$. By rewriting the last two sets of equations in matrix form we get,

$$\begin{pmatrix} L_{k_1}^r(A)(I - \Xi) \\ \widehat{L}_{k_2}^r(A) \end{pmatrix} \boldsymbol{\alpha} = 0, \tag{4.26}$$

where $\widehat{L}_{k_2}^r(A)$ is the last $k_2$ rows of $L_{k_1+k_2}^r(A)$ and

$$\Xi = \begin{pmatrix} 0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & \xi_1^2 & \xi_1^3 & \cdots & \xi_1^{r-1} \\ 0 & 0 & 0 & \xi_2^3 & \cdots & \xi_2^{r-1} \\ 0 & 0 & 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \xi_{r-1}^{r-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}, \qquad \boldsymbol{\alpha} = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \\ \vdots \\ \\ \alpha_{r-1} \end{pmatrix}.$$

Note that $I - \Xi$ is invertible by construction. Since the excited modes correspond to distinct eigenvalues, if $r \leq \max\{k_1, k_2\}$, then either $L_{k_1}^r(A)$ or $\widehat{L}_{k_2}^r(A)$ is full column rank. Either way, equation (4.26) implies that $\alpha = 0$ and thus $\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{r-1}\}$ is a set of linearly independent vectors. This observation completes the proof as $r \leq \max\{k_1, k_2\}$ was chosen arbitrary. □

### *4.3.5 Special Case of $\mathcal{R}(A) \subset \mathcal{R}(B)$*

In order to better understand the behavior of DGR, in this section, we study the more restricted case where $\mathcal{R}(A) \subset \mathcal{R}(B)$. This case includes the case where $\text{rank}(B) = n$, i.e., one can directly control each state of the system (e.g. see [69,143]). Note that $\mathcal{R}(A) \subset \mathcal{R}(B)$ implies that $\widetilde{A} = 0$ which, in turn, results in regularizability of $(A, B)$. This, together with Corollary 2, results in the following corollary.

**Corollary 3.** *For any matrix $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$, where $\mathcal{R}(A) \subseteq \mathcal{R}(B)$, the trajectory generated by Algorithm 6 with $\alpha = 0$ satisfies,*

$$\|\boldsymbol{x}_t\| \leq M_t(A)\|\boldsymbol{x}_0\|\,,$$

*with $M_t(A)$ as in Definition 8.*

*Proof.* Note that $\widetilde{A} = \Pi_{\mathcal{R}(B)^\perp} A = 0$ whenever $\mathcal{R}(A) \subseteq \mathcal{R}(B)$. The claim now follows by Corollary 2 since $\widetilde{A}\widetilde{B} = 0$ and thus $a_k = 0$ for all $k = 1, \ldots, t-1$. □

The latter bound becomes more structured for a symmetric $A$ by combining the results from Corollary 3 and Lemma 23 which we skip for brevity.

**Remark 14.** *In order to further illustrate the bound stated in Corollary 3, assume that $\delta e \leq 1$. Then, from Lemma 23,*

$$\frac{\|\boldsymbol{x}_t\|^2}{\|\boldsymbol{x}_0\|^2} \leq \left[\frac{\sigma_1^2}{t}\right]^t + \sum_{j=1}^{\lfloor t/2\rfloor}\left[\frac{\sigma_1^2}{t-j}\right]^{t-j}\left[\frac{t}{j}\right]^j + \sum_{j=\lfloor t/2\rfloor+1}^{t-1}\left[\frac{t\,\sigma_1^2}{(t-j)^2}\right]^{t-j} + 1,$$

*where we have also used $\binom{t}{j} \leq (e\,t/j)^j$. This implies that as $t$ gets larger than $\sigma_1^2$, the terms with large powers admit smaller bases and those with large bases will gain smaller powers comparing to $\sigma_1^{2t}$. This is despite the fact that for small $t$, the relative norm of the state might grow.*

In the sequel, as a result of linear independence established in Theorem 26 we show how the simplified bounds clarify the elimination of the unstable modes in the system.

**Corollary 4.** *Suppose $A$ is diagonalizable with $\mathcal{R}(A) \subseteq \mathcal{R}(B)$ and let $\boldsymbol{x}_0$ excite $k$ modes of $A$. If $r$ eigenvalues corresponding to the $k$ excited modes are distinct for any $r \leq k$, then $\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{r-1}\}$, generated by Algorithm 6 with $\alpha = 0$, is a set of linearly independent vectors.*

*Proof.* Given that $\mathcal{R}(A) \subseteq \mathcal{R}(B)$, all the modes of $A$ are contained in $\mathcal{R}(B)$, so without loss of generality, let $\lambda_1, \ldots, \lambda_k$ be the eigenvalues corresponding to the excited modes $u_1, \ldots, u_k \in \mathcal{R}(B)$. Then, following the proof of Theorem 26, Equation (4.25) reduces to,

$$\boldsymbol{x}_t = \textstyle\sum_{\ell=1}^{k} \beta_\ell \left[(\lambda_\ell)^t - \sum_{j=1}^{t-1} \xi_j^t (\lambda_\ell)^j\right] \boldsymbol{u}_\ell.$$

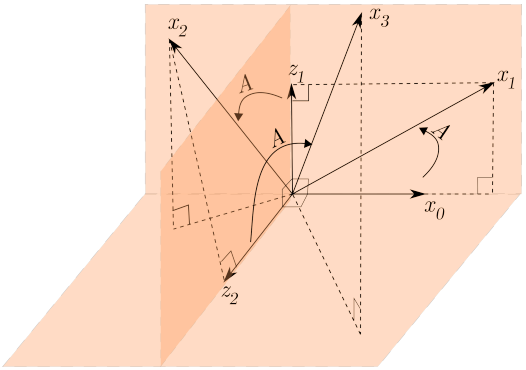



Figure 4.4: A geometric schematic of DGR when $\mathcal{R}(A) \subseteq \mathcal{R}(B)$. Since $\boldsymbol{z}_0 \coloneqq \boldsymbol{x}_0$, $\boldsymbol{z}_t \perp \mathcal{R}(\mathcal{X}_{t-1})$ and $\boldsymbol{z}_t \in \mathcal{R}(\mathcal{X}_t)$ for $t = 1, 2$, the set $\{\boldsymbol{z}_0, \boldsymbol{z}_1, \boldsymbol{z}_2\}$ consists of orthogonal vectors.

Now, let $\widehat{\boldsymbol{x}} = \sum_{j=0}^{r-1} \alpha_j \boldsymbol{x}_j$ for some $\alpha_j \in \mathbb{C}$ and $r \leq k$. Then following the same argument in the proof of theorem 26 about $\widehat{\boldsymbol{x}}$, equation (4.26) reduces to,

$$L_k^r(A)(I - \Xi)\boldsymbol{\alpha} = 0,$$

with similar definitions of $\Xi$ and $\boldsymbol{\alpha}$, and $L_k^r(A)$ as defined in equation (4.23). Since $r$ eigenvalues corresponding to $k$ excited modes are distinct, $L_k^r(A)$ has full column rank. As $I - \Xi$ is invertible, we conclude that $\boldsymbol{\alpha} = 0$ meaning that $\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{r-1}\}$ are linearly independent. □

An immediate consequence of the above corollary is that DGR generates data that is effective for simultaneous identification of modes with multiplicity greater than one.

**Proposition 8.** *Suppose that $A$ is diagonalizable with $\mathcal{R}(A) \subseteq \mathcal{R}(B)$, and let $\boldsymbol{x}_0$ excite $k$ modes of $A$ corresponding to $r$ distinct eigenvalues (where possibly $r \leq k$). Then, in $r$ iterations of Algorithm 6 with $\alpha = 0$, $\operatorname{span}\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{r-1}\}$ coincides with the subspace containing these excited modes; furthermore, $\boldsymbol{x}_{r+1} = 0$.*

*Proof.* Without loss of generality, let $\boldsymbol{x}_0$ excite the $k$ modes of $A$ corresponding to $\lambda_1, \cdots, \lambda_r$. Since $A$ is diagonalizable, let $A = U\Lambda U^{-1}$ be its eigen-decomposition and so $\boldsymbol{x}_0$ excite $\{\boldsymbol{u}_1, \cdots, \boldsymbol{u}_k\}$, i.e., $\boldsymbol{x}_0 = \sum_{i=1}^{k} \beta_i \boldsymbol{u}_i$, with $\beta_i \neq 0$. Define

$$\mathcal{M}(\lambda_i) = \{j : \boldsymbol{u}_j \text{ is the eigenvector corresponding to } \lambda_i\},$$

for $i = 1, \cdots, r$. Furthermore, define the $r$-dimensional subspace,

$$S \coloneqq \text{span}\left\{\sum_{j\in\mathcal{M}(\lambda_1)} \beta_j \boldsymbol{u}_j,\ \cdots, \sum_{j\in\mathcal{M}(\lambda_r)} \beta_j \boldsymbol{u}_j\right\},$$

where the span is taken over the complex field. We prove by induction that $\boldsymbol{x}_t \in S$ for all $t = 1, \cdots, r$. Notice that $\boldsymbol{x}_0 \in S$ and suppose that $\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{t-1}\} \subset S$; recall from the proof of Corollary 4 that $\boldsymbol{x}_t = A\boldsymbol{z}_{t-1}$, where $\boldsymbol{z}_{t-1} = \Pi_{\mathcal{R}(\mathcal{X}_{t-2})^\perp}(\boldsymbol{x}_{t-1})$. Since $\boldsymbol{x}_{t-1} \in S$ and $\text{span}\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{t-2}\} \subset S$, one can conclude that $\boldsymbol{z}_{t-1} \in S$, and from the definition of $S$, $\boldsymbol{x}_t = A\boldsymbol{z}_{t-1} \in S$. On the other hand, since $\lambda_1, \lambda_2, \cdots, \lambda_r$ are distinct eigenvalues, by Corollary 4, $\mathbf{dim}\,(\text{span}\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{r-1}\}) = r$. By hypothesis of the induction $\text{span}\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{r-1}\} \subset S$, and since $\mathbf{dim}(S) = r$, we conclude that $\text{span}\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{r-1}\}$ must be the entire $S$, i.e. $\text{span}\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{r-1}\} = S$, proving the first claim. Lastly, since $\boldsymbol{x}_r \in S$, $\boldsymbol{z}_r = \Pi_{\mathcal{R}(\mathcal{X}_{r-1})^\perp}(\boldsymbol{x}_r) = 0$, and thus $\boldsymbol{x}_{r+1} = A\boldsymbol{z}_r = 0$, thereby completing the proof. □

Following Proposition 8, if $\boldsymbol{x}_0$ excites $k$ modes of the system corresponding to distinct eigenvalues with trivial algebraic multiplicities, then Algorithm 6 identifies all the excited modes of the system in $k$ iterations. Furthermore, this implies that $\boldsymbol{x}_{k+1} = 0$, i.e., DGR eliminates the unstable modes and regulates the unknown system in exactly $k+1$ iterations.

### *4.3.6 Boosting the Performance of DGR*

The DGR algorithm as introduced in Algorithm 6 can become computationally heavy for large-scale systems. This is mainly due to storing the entire history of data in $\mathcal{X}_t$ and $\mathcal{Y}_t$ followed by the update of the controller that finds the pseudoinverse as well as multiplication of these data matrices (steps 7-9). Assuming the SVD-based computation of pseudoinverse, the complexity of the method is[13] $\mathcal{O}(n^2 t)$. In this section, we show that such computational burden can be circumvented using rank-1 modifications of data matrices as a result of the discrete nature of data collection in our setup. Note that for computing $K_{t+1}$ from (4.19) we only need to access $\mathcal{Y}_{t+1}\mathcal{X}_t^\dagger$ (rather than $\mathcal{X}_t^\dagger$). To this end, we leverage the results of [117] in order to find $\mathcal{Y}_{t+1}\mathcal{X}_t^\dagger$ recursively as a function of $\mathcal{Y}_t\mathcal{X}_{t-1}^\dagger$, $\mathcal{X}_{t-1}\mathcal{X}_{t-1}^\dagger$, and $\boldsymbol{x}_t$.

[13] The multiplication $\mathcal{Y}_{t+1}\mathcal{X}_t^\dagger$ enforces another $\mathcal{O}(n^2 t)$ complexity that can be significant for large $n$.

**Proposition 9.** *Define $\mathcal{X}_{t-1}$ as in Algorithm 6 and let $\boldsymbol{x}_t$ be the collected data at time-step $t$. Then, if $\boldsymbol{x}_t \notin \mathcal{R}(\mathcal{X}_{t-1})$,*

$$\mathcal{X}_t^\dagger = \begin{pmatrix} \mathcal{X}_{t-1}^\dagger - \boldsymbol{\gamma}_t \boldsymbol{z}_t^\dagger \\ \boldsymbol{z}_t^\dagger \end{pmatrix}. \tag{4.27}$$

*Otherwise,*

$$\mathcal{X}_t^\dagger = \begin{pmatrix} \mathcal{X}_{t-1}^\dagger - \epsilon_t \boldsymbol{\gamma}_t \boldsymbol{\zeta}_t^\top \\ \epsilon_t \boldsymbol{\zeta}_t^\top \end{pmatrix}, \tag{4.28}$$

*where $\epsilon_t \in \mathbb{R}$, $\boldsymbol{\gamma}_t \in \mathbb{R}^t$, and $\boldsymbol{\zeta}_t \in \mathbb{R}^n$ are defined as,*

$$\epsilon_t = \frac{1}{\|\boldsymbol{\gamma}_t\|^2 + 1}, \qquad \boldsymbol{\gamma}_t = \mathcal{X}_{t-1}^\dagger \boldsymbol{x}_t, \qquad \boldsymbol{\zeta}_t = \left(\mathcal{X}_{t-1}^\dagger\right)^\top \boldsymbol{\gamma}_t, \tag{4.29}$$

*and $\boldsymbol{z}_t \in \mathbb{R}^n$ is defined in Lemma 22.*

*Proof.* Rearrange $\mathcal{X}_t$ into,

$$\mathcal{X}_t = \begin{pmatrix} \mathcal{X}_{t-1} & 0 \end{pmatrix} + \boldsymbol{x}_t \boldsymbol{e}_{t+1}^\top.$$

Then, it is implied from Theorem 1 in [117] that

$$\mathcal{X}_t^\dagger = \begin{pmatrix} \mathcal{X}_{t-1} & 0 \end{pmatrix}^\dagger + \left[\boldsymbol{e}_{t+1} - \begin{pmatrix} \mathcal{X}_{t-1} & 0 \end{pmatrix}^\dagger \boldsymbol{x}_t\right] \boldsymbol{z}_t^\dagger,$$

whenever $\boldsymbol{x}_t \notin \mathcal{R}(\mathcal{X}_{t-1})$. To simplify, note from the SVD of $\mathcal{X}_{t-1} = U\Sigma V^\top$ that,

$$\begin{aligned} \begin{pmatrix} \mathcal{X}_{t-1} & 0 \end{pmatrix}^\dagger = \left[U\Sigma \begin{pmatrix} V^\top & 0 \end{pmatrix}\right]^\dagger &= \left[U \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V^\top & 0 \\ 0 & 1 \end{pmatrix}\right]^\dagger \\ &= \begin{pmatrix} V & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \Sigma^\dagger & 0 \\ 0 & 0 \end{pmatrix} U^\top = \begin{pmatrix} \mathcal{X}_{t-1}^\dagger \\ 0 \end{pmatrix}. \end{aligned}$$

Hence,

$$\mathcal{X}_t^\dagger = \begin{pmatrix} \mathcal{X}_{t-1}^\dagger \\ 0 \end{pmatrix} + \left[\boldsymbol{e}_{t+1} - \begin{pmatrix} \mathcal{X}_{t-1}^\dagger \\ 0 \end{pmatrix} \boldsymbol{x}_t\right] \boldsymbol{z}_t^\dagger = \begin{pmatrix} \mathcal{X}_{t-1}^\dagger - \boldsymbol{\gamma}_t \boldsymbol{z}_t^\dagger \\ \boldsymbol{z}_t^\dagger \end{pmatrix}.$$

For the case when $\boldsymbol{x}_t \in \mathcal{R}(\mathcal{X}_{t-1})$, Theorem 3 in [117] gives,

$$\mathcal{X}_t^\dagger = \begin{pmatrix} \mathcal{X}_{t-1}^\dagger \\ 0 \end{pmatrix} + \boldsymbol{e}_{t+1} \boldsymbol{\gamma}_t^\top \mathcal{X}_{t-1}^\dagger - \frac{1}{\sigma} \boldsymbol{p}\boldsymbol{q}^\top,$$

where,

$$\sigma = \|\boldsymbol{\gamma}_t\|^2 + 1, \qquad \boldsymbol{p} = -\|\boldsymbol{\gamma}_t\|^2 \boldsymbol{e}_{t+1} - \begin{pmatrix} \boldsymbol{\gamma}_t \\ 0 \end{pmatrix}, \qquad \boldsymbol{q} = -\boldsymbol{\zeta}_t.$$

The rest of the proof follows from rearranging the terms and using the definitions in (4.29). □

As mentioned earlier, the update of the controller requires $\mathcal{Y}_{t-1}\mathcal{X}_{t-1}^{\dagger}$ that could become prohibitive for large $n$. However, we can take advantage of Proposition 9 to find this term recursively in order to avoid memory usage as well as computational burden.

**Theorem 27.** *Let $\mathcal{X}_{t-1}$ be defined as in Algorithm 6 and $\boldsymbol{x}_t$ be the collected data at time-step $t$. Define $\mathcal{P}_{t-1} = \mathcal{X}_{t-1}\mathcal{X}_{t-1}^{\dagger}$, $\mathcal{Q}_{t-1} = \mathcal{Y}_t\mathcal{X}_{t-1}^{\dagger}$, $\boldsymbol{z}_t = \big[\mathrm{I} - \mathcal{P}_{t-1}\big]\boldsymbol{x}_t$, and $\boldsymbol{y}_t = A\boldsymbol{x}_t$. Then,*

$$\begin{aligned} \mathcal{Q}_t &= \mathcal{Q}_{t-1} - \mathcal{Q}_{t-1}\boldsymbol{x}_t\boldsymbol{z}_t^{\dagger} + \boldsymbol{y}_t\boldsymbol{z}_t^{\dagger}, \\ \mathcal{P}_t &= \mathcal{P}_{t-1} + \boldsymbol{z}_t\boldsymbol{z}_t^{\dagger}. \end{aligned}$$

*Proof.* For the case $\boldsymbol{x}_t \notin \mathcal{R}(\mathcal{X}_{t-1})$ we get from Proposition 9,

$$\begin{aligned} \mathcal{P}_t = \mathcal{X}_t\mathcal{X}_t^{\dagger} &= \begin{pmatrix} \mathcal{X}_{t-1} & \boldsymbol{x}_t \end{pmatrix} \begin{pmatrix} \mathcal{X}_{t-1}^{\dagger} - \boldsymbol{\gamma}_t\boldsymbol{z}_t^{\dagger} \\ \boldsymbol{z}_t^{\dagger} \end{pmatrix} \\ &= \mathcal{P}_{t-1} + \boldsymbol{z}_t\boldsymbol{z}_t^{\dagger}. \end{aligned} \tag{4.30}$$

Observing that $\mathcal{Q}_t = A\mathcal{P}_t$ and $\boldsymbol{y}_t = A\boldsymbol{x}_t$, the recursive relation for $\mathcal{Q}_t$ can be derived from (4.30). When $\boldsymbol{x}_t \in \mathcal{R}(\mathcal{X}_{t-1})$, note that $\mathcal{X}_{t-1}\boldsymbol{\gamma}_t = \mathcal{X}_{t-1}\mathcal{X}_{t-1}^{\dagger}\boldsymbol{x}_t = \boldsymbol{x}_t$ which results in $\mathcal{P}_{t+1} = \mathcal{P}_t$. However, in this case $\boldsymbol{z}_t = (\mathrm{I} - \mathcal{P}_{t-1})\boldsymbol{x}_t = 0$ and therefore, the same results hold when $\boldsymbol{x}_t \in \mathcal{R}(\mathcal{X}_{t-1})$. □

Given the recursive behavior introduced in Theorem 27, the refined (fast) version of DGR is displayed in Algorithm 7. At each time-step $t$, we update $\mathcal{Q}_t$ based on the information from the new data and the projection $\mathcal{P}_{t-1}$ (hidden in $\boldsymbol{z}_t^{\dagger}$), which itself gets updated as a part of the recursion. The $n \times n$ matrix $\mathcal{Q}_t$ is then employed for the controller's update. Notice that $\boldsymbol{z}_t$ has the same meaning as in Lemma 22; however, here we compute it using $\mathcal{P}_{t-1}$—which

is obtained recursively—and put it in the matrix form to emphasize its computation (since $\boldsymbol{z}_t^\dagger = \boldsymbol{z}_t^\top / \|\boldsymbol{z}_t\|^2$ is simply a $1 \times n$ vector).

**Algorithm 7** Fast Data-Guided Regulation (F-DGR)

1: **Initialization**
2: Measure $\boldsymbol{x}_0$, set $K_0 = 0$ and $G_\alpha = (\alpha I + B^\top B)^\dagger B^\top$
3: Run system (4.15) and measure $\boldsymbol{x}_1$
4: Set $\mathcal{P}_0 = (\boldsymbol{x}_0 \boldsymbol{x}_0^\top)/\|\boldsymbol{x}_0\|^2$
5: $\mathcal{Q}_0 = (\boldsymbol{x}_1 \boldsymbol{x}_0^\top)/\|\boldsymbol{x}_0\|^2$
6: $K_1 = G_\alpha \mathcal{Q}_0$
7: $t = 1$
8: **While stopping criterion not met**
9: Compute $\boldsymbol{u}_t = -K_t \boldsymbol{x}_t$
10: Run system (4.15) and measure $\boldsymbol{x}_{t+1}$
11: Set $\boldsymbol{z}_t = (\mathrm{I} - \mathcal{P}_{t-1})\boldsymbol{x}_t$
12: $\mathcal{Q}_t = \mathcal{Q}_{t-1} + (\boldsymbol{y}_t - \mathcal{Q}_{t-1}\boldsymbol{x}_t)\boldsymbol{z}_t^\dagger$
13: $\mathcal{P}_t = \mathcal{P}_{t-1} + \boldsymbol{z}_t \boldsymbol{z}_t^\dagger$
14: $K_{t+1} = G_\alpha \mathcal{Q}_t$
15: $t = t + 1$

When $\boldsymbol{x}_t \in \mathcal{R}(\mathcal{X}_{t-1})$, using the definition, $\boldsymbol{z}_t = 0$ and $\mathcal{Q}_t$ remains the same. Otherwise, $\mathcal{Q}_t$ deviates from its previous value according to,

$$\mathcal{Q}_t - \mathcal{Q}_{t-1} = (\boldsymbol{y}_t - \mathcal{Q}_{t-1}\boldsymbol{x}_t)\boldsymbol{z}_t^\dagger = A\boldsymbol{z}_t \boldsymbol{z}_t^\dagger.$$

Recall that $\boldsymbol{z}_t$ reflects the informativity of the newly generated data $\boldsymbol{x}_t$. In fact, based on its definition, $\mathcal{Q}_t = \mathcal{Y}_{t+1}\mathcal{X}_t$ provides an estimate of $A$ up to time-step $t$. Hence, the update of $\mathcal{Q}_t$ on line 12 of Algorithm 7 essentially adjusts the prior estimate of $A$ based on the new information encoded in the term $A\boldsymbol{z}_t \boldsymbol{z}_t^\dagger$. Moreover, $\boldsymbol{y}_t$ captures the effect of the unknown dynamics at each iteration. All in all, the machinery provided in this section circumvents the computational load of finding pseudo-inverses by leveraging the recursive nature of the solution methodology.

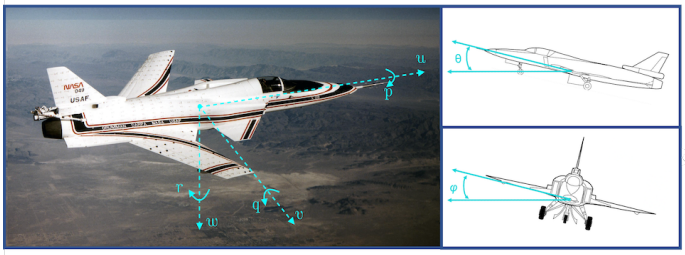


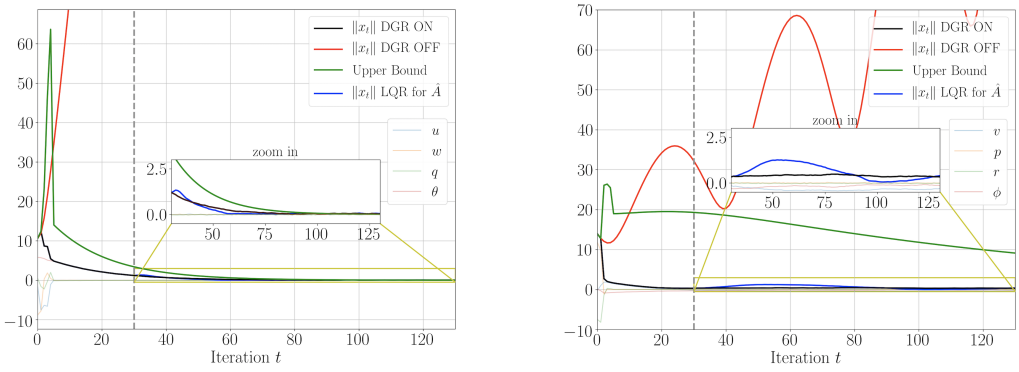

Figure 4.5: Grumman X-29A *(Credits: NASA Photo)*, mainly known for its extreme instability while providing high-quality maneuverability. The longitudinal and lateral-directional states are also illustrated.

### *4.3.7 Example*

In order to showcase the advantages of the proposed method in practical settings, we have implemented DGR on data collected from the X-29A aircraft. The Grumman X-29A is an experimental aircraft initially tested for its forward-swept wing; it was designed with a high degree of longitudinal static instability (due to the location of the aerodynamic center on the wings) for maneuverability, where linear models were leveraged to determine the closed-loop stability (Figure 4.5). The primary task of the control laws is to stabilize the longitudinal motion of the aircraft. To this end, the dynamic elements of the flight control system is designed for two general modes: 1) the Normal Digital Powered Approach (ND-PA) used in the takeoff and landing phase of the flight, and 2) the Normal Digital Up-and-Away (ND-UA) used otherwise.

For both flight modes, we study the case where the dynamics of the aircraft has been perturbed and unknown. This can be due to a mis-estimation of system parameters and/or any unpredicted flaw in the flight dynamics due to malfunction/damage. In this setting, the control laws designed for the original system fail and the system can become highly unstable. We then let DGR to regulate the system; in this case, since aircraft continues to operate safely, one can use any data-driven identification, stabilization, or robust control methods once enough data has been collected.

For both longitudinal and lateral-directional dynamics in each operating mode, the nom-




(a) (b)

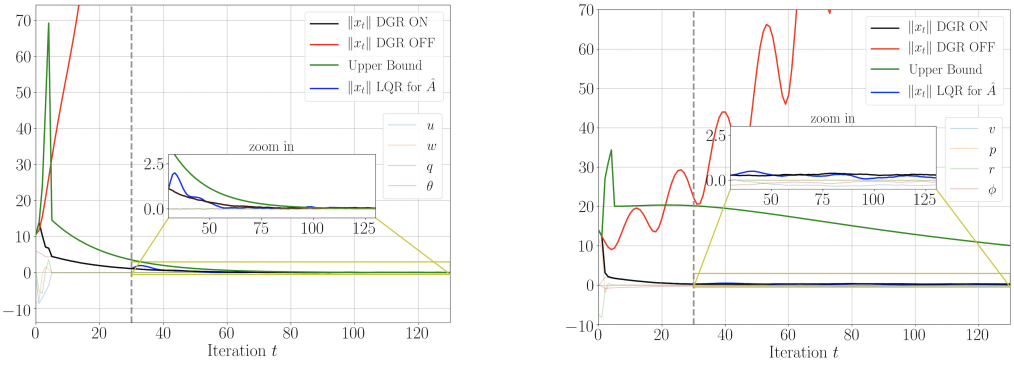
Figure 4.6: The state trajectory of X-29 in ND-PA mode with and without DGR for a) Longitudinal control, b) Lateral-Directional control.

inal system parameters are obtained from Tables 9-10 and 13-14 in [26] (with fixed discretization step-size $\gamma = 0.05$), whereas perturbation $\Delta A$ is assumed to shift the dynamics to,

$$\boldsymbol{x}_{t+1} = (A + \Delta A)\boldsymbol{x}_t + B\boldsymbol{u}_t + \boldsymbol{\omega}_t,$$

where the elements of $\Delta A$ are sampled from a normal distribution $\mathcal{N}(0, 0.05)$ and $\boldsymbol{\omega}_t \sim \mathcal{N}(0, 0.01)$ denotes the process noise. Note that even though the nominal dynamics is known in this example, the proposed machinery makes no such *a priori* estimate, and assumes a completely unknown dynamics $A + \Delta A$. The original controller for the unperturbed system in each mode is assumed to be a closed-loop infinite horizon LQR with states and inputs weights $Q = R = I$ (since the dynamics is already normalized).

We now aim to regulate the unstable system $A_{\text{new}} \coloneqq A + \Delta A$ from random initial states (where each state is sampled from $\mathcal{N}(0, 10.0)$). Note that both the original system and the perturbed system have effective input characteristic that make them regularizable (with $\rho(\widetilde{A}) = 0.998$ and $\rho(\widetilde{A}_{\text{new}}) = 0.881$ for ND-PA mode, and $\rho(\widetilde{A}) = 0.998$ and $\rho(\widetilde{A}_{\text{new}}) = 0.876$ for ND-UA mode). The resulting state trajectories for ND-PA and ND-UA modes are demonstrated in Figure 4.6 and 4.7, respectively. Without using DGR the norm of the state




(a) (b)

Figure 4.7: The state trajectory of X-29 in ND-UA mode with and without DGR for a) Longitudinal control, b) Lateral-Directional control.

$\|\boldsymbol{x}_t\|$ would grow rapidly (red curve) as the unknown system is unstable and the original control laws fail.[14] As the plots suggest, with DGR in the feedback loop (with choice of $\alpha = 0$),[15] the unstable modes can be suppressed resulting in stabilization of the system (norm of the states in this case is demonstrated in black and each the state is depicted in faded color). Up to step $t = 30$ (shown with vertical dashed-line), enough data is generated in order to estimate the new system dynamics since the dimension of the augmented system is only 8 and the dynamic noise is relatively small. Now for the sake of comparison, one can replace DGR with a closed-loop infinite horizon LQR controller, with some cost-weights $Q$ and $R$ (here we set $Q = R = I$) which is obtained using the new estimate of the system dynamics; in this case, only norm of the states is depicted in blue for comparison.

In contrast to the original unstable LQR controller, it is shown that the blue curve is stabilizing since we now have a more accurate estimate of the (perturbed) system parameters using the data generated safely by DGR in the loop.

[14] Since the LQR solution, in general, may have small stability margins for general parameter perturbations [178].

[15] The positive choice for $\alpha$ adjusts the algorithm to compromise the regulation of states in order to decrease 2-norm of the input. This may lead to larger upper bound specially when the system is unstable.

In these examples, the bound derived in Theorem 25 is plotted in green for comparison. The behavior of the bound follows our observations in Remark 12; the bound increases as the algorithm initially tries to "detect" the unstable modes, followed by suppressing these modes for regulation. We finally note that for large enough iterations, the rate of change of the upper bound is dictated by $\rho(\widetilde{A}_{\text{new}})$ which in this case, is slightly less than one. [16]

### *4.3.8 Example: Data-Driven Stabilization of Perturbed Signed Networks*

Signed networks refer to systems that admit negative edges as well as positive ones implying antagonistic relationship in networks such as friendship/adversary, enemy/ally, etc. From the dynamics viewpoint, this setup has been well-studied for the linear case in the past few years.[17] Notably, it has been shown that *structural balance* is a key notion in such networks that leads to stable behavior and convergence to opposite clusters. Perturbations are induced in social networks due to distortions in existence, nature, or intensity of interactions among the individuals.

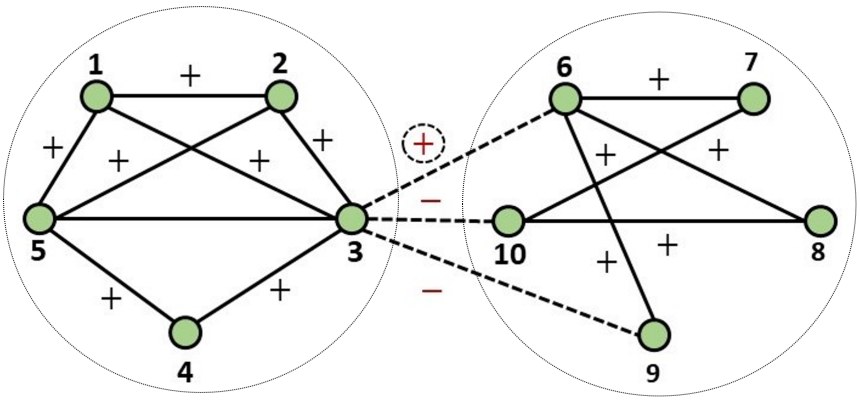



Figure 4.8: Model perturbations in signed networks such as sign flips can lead to instability.

The resulting unstable behavior generally have unfavorable ramifications such as the advent of clustering or community cleavage [16, 68]. It is shown that a model perturbation

[16] The code for this simulation can be found at https://github.com/shahriarta/Data-Guided-Regulation.

[17] For more on clustering and controllability in signed networks the reader is referred to [8, 13]

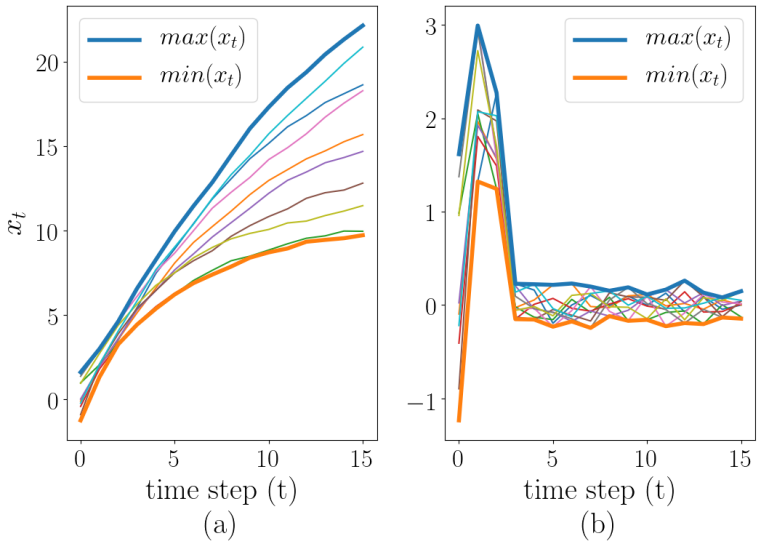



Figure 4.9: State trajectories of an unstable signed network resulted from a perturbed consensus dynamic (4.31), (a) without DGR, (b) with DGR.

as simple as a sign flip can cause instability in signed networks [10]. In this example, we consider a similar case with two groups of 5 nodes with all positive edges within each group and negative edges in between. The system is perturbed as shown in Figure 4.8 with a change of sign introducing one unstable mode in the dynamics. The states' constraint is set to $\|x_t\|_\infty < 5$ and the noise is a Gaussian signal with $\mu = 2$ and $\sigma^2 = 0.5$ , forming the dynamics,

$$x_{t+1} = (\mathcal{L} + \Delta\mathcal{L})x_t + u_t + \omega_t, \tag{4.31}$$

where $\mathcal{L}$ and $\Delta\mathcal{L}$ refer to the Laplacian and the sign perturbation, respectively,[18] and are defined based on the structure given in Figure 4.8. The results of applying DGR to this scenario are demonstrated in Figure 4.9. The comparison of the trajectories clearly shows how DGR captures the unstable mode of the system in 2 times-steps respecting the specified safety bound in the problem setup.

[18]For example, the perturbations in the case of sign flips are in the form of $e_i e_j^\top + e_j e_i^\top$ for $i \neq j$ only affecting the adjacency matrix. More on graph theoretical notation can be found in [116]

# Chapter 5

# CONCLUSION AND FUTURE WORK

## *5.1 Concluding Remarks*

This thesis is mainly focused on answering the general question of how can networked dynamical systems help to get a better understanding of the control and analysis of highly complex and large systems that are otherwise difficult—if not impossible—to control. Based on a variety of applications, I touched upon different aspects of networks, their characteristics, and how they lend their structure into the dynamics of those applications. To this end, the development of ideas in this manuscript are based on a wide range of research topics including distributed control, game theory, graph theory, machine learning, linear algebra, and dynamical systems.

In the series of works that were presented in this dissertation, the network control is tackled from a variety of perspectives. In the first part of the thesis, we start the analysis from a system theoretic viewpoint and consider networks as systems that enjoy a fixed underlying graph structure. Then based on prior knowledge of the consensus protocol, we tackle different problem in control and game theory based on the newly introduced structure to the system. First, we study the controllability and stabilizability of signed consensus networks. Most importantly, we show how the underlying symmetry in the system can be sufficient for its uncontrollability. Next, the attention is shifted to the application of networked systems on games and, specifically, those types of games that can be modeled as a network game. Therein, we provide an algorithm to optimize the cost of each team towards the Nash equilibrium using a dual averaging scheme. Lastly, in this part, we considered the application of layered network on influence models in social networks. Our approach to this problem was a guaranteed-cots design as a remedy for the potential model perturbations, where we extended previous results in robust control design to multi-layered complex systems.

Generalizing the results of the first section, in the second part, we switch the scope of the analysis such that the agents in the network own their absolute dynamics and exchange information possibly through their states, costs, etc., enabling us to reason about distributed control schemes in such a framework. First, we provide a distributed scheme based on individual control setup where each unit tries to achieve some optimality while interacting with the rest of the network. We show that while such a scheme is computationally efficient, it does not take into consideration the (sub)optimality of the entire system. Therefore, we provide another setup in the second section of this part that finds an LQR-based structured controller based on a similar setup, where we let some auxiliary links in the system to compensate for the unknown dynamics. Our results show that the proposed controller is stabilizing at all steps of the algorithm and the final outcome is suboptimal compared to the best unstructured LQR solution.

In the final chapter of the thesis, we study some general aspects of non-asymptotic control and analysis of linear systems that, in prospect, can be helpful in resolving some the shortcomings of the methods in the previous sections. In particular, online regulation is proposed which is one remedy to the impractical assumption of starting a policy iteration algorithm from a stabilizing point. Then using the DGR algorithm, one can run the black-box system and wait for some time-steps until the unstable modes of the (unknown) system are annihilated. We also provide examples at each section to demonstrate the usefulness of the results in a network setting.

### *5.2 Future Directions*

As a view in hindsight, the study proposed in this thesis shows the potentials of the network control theory and its flexibility to accommodate a variety of applications. As the content of this dissertation suggest, distributed control of complex system is in general an NP-hard constrained optimization problem, yet the computational complexity makes finding a solution to such problems vital. This problem becomes even more difficult when least is known about the underlying dynamics and parameters of the system—demanding the need for a data-guided mechanism. Hence, finding and employing underlying structures of such systems is inevitable for real applications in large scales. In this dissertation, we are

mainly interested in those structure that arise from interaction between multiple agents in one or many interconnected systems. This area is a current topic of research and intersects many fields of study including control theory, graph theory, machine learning, mathematical analysis, and linear algebra. To conclude, we mention a few of potential extensions that make the provided theory more adaptable for real applications.

Viewing networks as systems suggests that a control designer can think of almost every system theoretic advancement over the rich history of control theory such as robust and adaptive control into networks. Thinking about antagonistic networks, one such direction is to examine the limitation of performance for controlled networks in case additional clusters (more than two) and also reflect the signed structure into a game theoretic framework. Regarding the analysis of game networks, stochastic access to subdifferential of cost functions as well as noisy observations of opponent's action is considered as the next immediate extension of the work. Additionally, analysis of the step-size for the dual variables and equipping the algorithm with feasible second-order information are yet other achievable furtherance that can improve the convergence behavior of this method close to equilibrium. Also, to further extend the layered control segment of this section, some future directions include generalization of the dynamics such that every layer can potentially contribute to the control mechanism for the entire system. One can also aim to provide a more topological structure to the design parameters $Q_{\otimes}$ and $R_{\otimes}$.

Distributed control for a system with unknown dynamics is novel research area with many directions for improvement. First, as discussed in the second part of the thesis the observation can be further extended to more elaborate cost structure, highlighting the trade-off between local and global optimality in large-scale distributed systems. This can be achieved if other types of interconnections such as dynamics or feedback coupling as well as consensus through the $Q$-function are adopted for the analysis. Another line of work is to consider other types of data-guided distributed control mechanisms for structures such as layered systems or systems with switching dynamics. Regarding the D3PI algorithm, extension of the current results to a heterogeneous system of agents is considered as an immediate future direction of our work. Moreover, D3PI builds upon parameter estimation techniques that represent an end-to-end policy prediction directly from observed data. In

a black-box linear setup, model-based methods—*i.e.*, extracting an approximate model from data and use it for control—can be integrated in this multiagent setup to further reduce sample and time complexities of the algorithm and is addressed as a potential future direction of this project.

Finally, the extensions of the results presented in the final technical chapter to noisy dynamics as well as an unknown input matrix are deferred for future work. Furthermore, state regulation becomes more challenging when one only relies on partial observation of system's trajectory or when the system is known to have multi-scale dynamics. Also, our setup would be more practical considering input constraints, *e.g.*, rate limits. While it is straightforward to address such extensions via convex constraints in the proposed procedure, analysis of the resulting closed loop trajectory is more involved.

# BIBLIOGRAPHY

[1] Antonio Acín, J. Ignacio Cirac, and MacIej Lewenstein. Entanglement percolation in quantum networks. *Nature Physics*, 3(4):256–259, 4 2007.

[2] Naman Agarwal, Brian Bullins, Elad Hazan, Sham Kakade, and Karan Singh. Online Control with Adversarial Disturbances. In *International Conference on Machine Learning*, pages 111–119, 2019.

[3] Ricardo Aguilar-López, M Isabel Neria-González, Rafael Martinez-Guerra, and Juan L Mata-Machuca. Nonlinear estimation in a class of gene transcription process. *Applied Mathematics and Computation*, 226:131–144, 2014.

[4] J Akiyama, D Avis, V Chvatal, and H Era. Balancing Signed Graphs. *Discrete Applied Mathematics*, 3:227–233, 1981.

[5] A Alaeddini, K Morgansen, and M Mesbahi. Adaptive communication networks with privacy guarantees. In *American Control Conference (ACC)*, pages 4460–4465, 5 2017.

[6] Atiye Alaeddini, Siavash Alemzadeh, Afshin Mesbahi, and Mehran Mesbahi. Linear Model Regression on Time-series Data: Non-asymptotic Error Bounds and Applications. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 2259–2264. IEEE, 2018.

[7] Atiye Alaeddini, Siavash Alemzadeh, Afshin Mesbahi, and Mehran Mesbahi. Linear Model Regression on Time-series Data: Non-asymptotic Error Bounds and Applications. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2259–2264, 2019.

[8] Siavash Alemzadeh, Mathias Hudoba De Badyn, and Mehran Mesbahi. Controllability and stabilizability analysis of signed consensus networks. In *1st Annual IEEE Conference on Control Technology and Applications, CCTA 2017*, pages 55–60, 2017.

[9] Siavash Alemzadeh and Mehran Mesbahi. Distributed Q-learning for dynamically decoupled systems. In *Proceedings of the American Control Conference*, pages 772–777, 2019.

[10] Siavash Alemzadeh and Mehran Mesbahi. Influence Models on Layered Uncertain Networks: A Guaranteed-Cost Design Perspective. In *Proceedings of the IEEE Conference on Decision and Control*, pages 5251–5256, 2019.

[11] Siavash Alemzadeh, Hesam Talebiyan, Shahriar Talebi, Leonardo Duenas-Osorio, and Mehran Mesbahi. Resource Allocation for Infrastructure Resilience using Artificial Neural Networks. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 617–624, 2020.

[12] Tansu Alpcan and Tamer Basar. Distributed Algorithms for Nash Equilibria of Flow Control Games. In *Advances in Dynamic Games: Applications to Economics, Finance, Optimization, and Stochastic Control*, pages 473–498. Birkhäuser Boston, Boston, MA, 2005.

[13] Claudio Altafini. Dynamics of opinion forming in structurally balanced social networks. *PloS one*, 7(6):5876–5881, 2012.

[14] Claudio Altafini. Consensus problems on networks with antagonistic interactions. *IEEE Transactions on Automatic Control*, 58(4):935–946, 2013.

[15] Zahra Askarzadeh, Rui Fu, Abhishek Halder, Yongxin Chen, and Tryphon T Georgiou. Stability Theory in l1 for Nonlinear Markov Chains and Stochastic Models for Opinion Dynamics over Influence Networks. *arXiv preprint arXiv:1706.03158*, 2017.

[16] Zahra Askarzadeh, Rui Fu, Abhishek Halder, Yongxin Chen, and Tryphon T Georgiou. Stability theory of stochastic models in opinion dynamics. *IEEE Transactions on Automatic Control*, 2019.

[17] Karl J Åström and Björn Wittenmark. *Adaptive control*. Courier Corporation, 2013.

[18] Lubomír Bakule. Decentralized control: An Overview. *Annual Reviews in Control*, 32(1):87–98, 2008.

[19] Bassam Bamieh, Fernando Paganini, and Munther A Dahleh. Distributed control of spatially invariant systems. *IEEE Transactions on Automatic Control*, 47(7):1091–1107, 2002.

[20] T Başar and GJ Olsder. *Dynamic noncooperative game theory*. 1998.

[21] Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

[22] Julian Berberich, Anne Romer, Carsten W Scherer, and Frank Allgöwer. Robust data-driven state-feedback design. *arXiv preprint arXiv:1909.04314*, 2019.

[23] Béla Bollobás. *Modern graph theory*, volume 184. Springer Science & Business Media, 2013.

[24] Francesco Borrelli and Tamás Keviczky. Distributed LQR design for identical dynamically decoupled systems. *IEEE Transactions on Automatic Control*, 53(8):1901–1912, 2008.

[25] Francesco Borrelli, Tamás Keviczky, Gary J Balas, Greg Stewart, Kingsley Fregene, and Datta Godbole. Hybrid decentralized control of large scale systems. In *International Workshop on Hybrid Systems: Computation and Control*, pages 168–183. Springer, 2005.

[26] John T Bosworth. *Linearized aerodynamic and control law models of the X-29A airplane and comparison with flight data*, volume 4356. NASA, 1992.

[27] S.J. Bradtke, B.E. Ydstie, and a.G. Barto. Adaptive linear quadratic control using policy iteration. *Proceedings of American Control Conference*, 3(2):3475–3479, 1994.

[28] Steven J Bradtke. *Incremental dynamic programming for on-line adaptive optimal control*. PhD thesis, Citeseer, 1994.

[29] Jingjing Bu, Afshin Mesbahi, Maryam Fazel, and Mehran Mesbahi. LQR through the lens of first order methods: Discrete-time case. *arXiv preprint arXiv:1907.08921*, 2019.

[30] Jingjing Bu, Afshin Mesbahi, and Mehran Mesbahi. On Topological Properties of the Set of Stabilizing Feedback Gain. *IEEE Transactions on Automatic Control*, 2020.

[31] Marko Budi***v***sić, Ryan Mohr, and Igor Mezić. Applied koopmanism. *Chaos*, 22(4):47510, 10 2012.

[32] Francesco Bullo, Jorge Cortes, and Sonia Martinez. *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton University Press, 2009.

[33] Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

[34] Dorwin Cartwright and Frank Harary. Structural balance: a generalization of Heider's theory. *Psychological Review*, 63(5):277, 1956.

[35] SSSL Chang and T Peng. Adaptive Guaranteed Cost Control of Systems with Uncertain Parameters. *IEEE Transactions on Automatic Control*, 17(4):474–483, 1972.

[36] Ting-Jui Chang and Shahin Shahrampour. Distributed Online Linear Quadratic Control for Linear Time-invariant Systems. *arXiv preprint arXiv:2009.13749*, 2020.

[37] Airlie Chapman, Andrés D González, Mehran Mesbahi, Leonardo Dueñas-Osorio, and Raissa M D'Souza. Data-guided Control: Clustering, Graph Products, and Decentralized Control. In *IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 493–498, 2017.

[38] Airlie Chapman and Mehran Mesbahi. State Controllability, Output Controllability and Stabilizability of Networks : A Symmetry Perspective. In *Proc. 54th IEEE Conference on Decision and Control*, pages 4776–4781, Osaka, Japan, 2015.

[39] Airlie Chapman, Marzieh Nabi-Abdolyousefi, and Mehran Mesbahi. Controllability and observability of network-of-networks via cartesian products. *IEEE Transactions on Automatic Control*, 59(10):2668–2679, 2014.

[40] Xinyi Chen and Elad Hazan. Black-Box Control for Linear Dynamical Systems. *arXiv preprint arXiv:2007.06650*, 2020.

[41] Andrew Clark, Andrew Clark, Qiqiang Hou, Linda Bushnell, and Radha Poovendran. A Submodular Optimization Approach to Leader- Follower Consensus in Networks with Negative Edges. In *Proc. American Control Conference*, Seattle, USA, 2017.

[42] Pierre Comon and Gene H Golub. Tracking a few extreme singular values and vectors in signal processing. *Proceedings of the IEEE*, 78(8):1327–1343, 1990.

[43] Jean Pierre Corfmat and A Stephen Morse. Decentralized control of linear multivariable systems. *Automatica*, 12(5):479–495, 1976.

[44] Marco Cremonini and Francesca Casamassima. Controllability of Social Networks and the Strategic Use of Random Information. *Computational Social Networks*, 4(1):10, 2017.

[45] Mathias Hudoba de Badyn, Siavash Alemzadeh, and Mehran Mesbahi. Controllability and data-driven identification of bipartite consensus on nonlinear signed networks. In *IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 3557–3562, 2017.

[46] Mathias Hudoba De Badyn, Siavash Alemzadeh, and Mehran Mesbahi. Controllability and data-driven identification of bipartite consensus on nonlinear signed networks. In *IEEE 56th Annual Conference on Decision and Control*, pages 3557–3562, 2018.

[47] M C de Oliveira, J Bernussou, and J C Geromel. A new discrete-time robust stability condition. *Systems & Control Letters*, 37(4):261–265, 1999.

[48] M C de Oliveira, J C Geromel, and Liu Hsu. LMI characterization of structural and robust stability: the discrete-time case. *Linear Algebra and its Applications*, 296(1):27–38, 1999.

[49] C De Persis and P Tesi. Formulas for Data-Driven Control: Stabilization, Optimality, and Robustness. *IEEE Transactions on Automatic Control*, 65(3):909–924, 2020.

[50] Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. On the sample complexity of the linear quadratic regulator. *Foundations of Computational Mathematics*, pages 1–47, 2019.

[51] Sarah Dean, Stephen Tu, Nikolai Matni, and Benjamin Recht. Safely learning to control the constrained linear quadratic regulator. In *2019 American Control Conference (ACC)*, pages 5582–5588. IEEE, 2019.

[52] Paresh Deshpande, P P Menon, Christopher Edwards, and Ian Postlethwaite. Suboptimal distributed control law with H2 performance for identical dynamically coupled linear systems. *IET Control Theory & Applications*, 6(16):2509–2517, 2012.

[53] Paresh Deshpande, Prathyush P Menon, Christopher Edwards, and Ian Postlethwaite. A distributed control law with guaranteed LQR cost for identical dynamically coupled linear systems. In *American Control Conference (ACC)*, pages 5342–5347. IEEE, 2011.

[54] Persi Diaconis and Daniel Stroock. Geometric bounds for eigenvalues of Markov chains. *The Annals of Applied Probability*, 1(1):36–61, 1991.

[55] Thinh T. Doan, Subhonmesh Bose, D. Hoa Nguyen, and Carolyn L. Beck. Convergence of the Iterates in Mirror Descent Methods. *IEEE Control Systems Letters*, 3(1):114–119, 2019.

[56] P Dorato, C T Abdallah, and V Cerone. *Linear Quadratic Control: An Introduction*. Krieger Publishing Company, 1995.

[57] Peter Dorato, Vito Cerone, and Chaouki Abdallah. *Linear-quadratic control: an introduction*. Simon & Schuster, Inc., 1994.

[58] John C. Duchi, Alekh Agarwal, and Martin J. Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. In *IEEE Transactions on Automatic Control*, volume 57, pages 592–606, 2012.

[59] Jacob Engwerda. *LQ Dynamic Optimization & Differential Games*. John Wiley & Sons, 2005.

[60] Giulietti Fabrizio, Lorenzo Pollini, and Mario Innocenti. Autonomous Formation Flight. *IEEE Control Systems Magazine*, (December):34–44, 2000.

[61] Giuseppe Facchetti, Giovanni Iacono, and Claudio Altafini. Computing global structural balance in large-scale signed social networks. *Proceedings of the National Academy of Sciences of the United States of America*, 108(52):20953–20958, 2011.

[62] Francisco. Facchinei and Jong-Shi. Pang. *Finite-dimensional Variational Inequalities and Complementarity Problems*. Springer, 2003.

[63] Mohamad Kazem Shirani Faradonbeh, Ambuj Tewari, and George Michailidis. Finite time adaptive stabilization of LQ systems. *IEEE Transactions on Automatic Control*, 2018.

[64] S Fattahi, N Matni, and S Sojoudi. Learning Sparse Dynamical Systems from a Single Sample Trajectory. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 2682–2689, 2019.

[65] J Alexander Fax and Richard M Murray. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9):1465–1476, 2004.

[66] Maryam Fazel, Rong Ge, Sham Kakade, and Mehran Mesbahi. Global Convergence of Policy Gradient Methods for the Linear Quadratic Regulator. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, 2018.

[67] Claude-Nicolas Fiechter. PAC adaptive control of linear systems. In *Conference on Computational learning theory*, pages 72–80. ACM, 1997.

[68] Noah E Friedkin. The Problem of Social Control and Coordination of Complex Systems in Sociology: A Look at the Community Cleavage Problem. *IEEE Control Systems Magazine*, 35(3):40–51, 2015.

[69] Noah E Friedkin and Eugene C Johnsen. Social influence and opinions. *Journal of Math. Sociology*, 15(3-4):193–206, 1990.

[70] P Frihauf, M Krstic, and T Basar. Nash Equilibrium Seeking in Noncooperative Games. *IEEE Transactions on Automatic Control*, 57(5):1192–1207, 5 2012.

[71] Yaser Ghaedsharaf, Milad Siami, Christoforos Somarakis, and Nader Motee. Eminence in presence of time-delay and structured uncertainties in linear consensus networks. In *IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 3218–3223, 2017.

[72] B Gharesifard and J Cortés. Distributed convergence to Nash equilibria in two-network zero-sum games. *Automatica*, 49(6):1683–1692, 2013.

[73] Chris Godsil and Gordon F Royle. *Algebraic graph theory*, volume 207. Springer Science & Business Media, 2013.

[74] Andrés D González, Airlie Chapman, Leonardo Dueñas-Osorio, Mehran Mesbahi, and Raissa M D'Souza. Efficient infrastructure restoration strategies using the recovery operator. *Computer-Aided Civil and Infrastructure Engineering*, 32(12):991–1006, 2017.

[75] Graham C Goodwin, Stefan F Graebe, and Mario E Salgado. *Control system design*. Upper Saddle River, NJ: Prentice Hall,, 2001.

[76] Graham C Goodwin and Kwai Sang Sin. *Adaptive filtering prediction and control*. Courier Corporation, 2014.

[77] S Grammatico. Proximal Dynamics in Multiagent Network Games. *IEEE Transactions on Control of Network Systems*, 5(4):1707–1716, 12 2018.

[78] Vijay Gupta, Babak Hassibi, and Richard M Murray. A sub-optimal algorithm to synthesize control laws for a network of dynamic agents. *International Journal of Control*, 78(16):1302–1313, 2005.

[79] Frank Harary and Jerald A Kabell. A simple algorithm to detect balance in signed graphs. *Mathematical Social Sciences*, 1(1):131–136, 1980.

[80] Simon Haykin. Adaptive filter theory. *Information and System Science. Prentice Hall*, 2002.

[81] Rainer Hegselmann and Ulrich Krause. Opinion dynamics and bounded confidence models, analysis and simulation. *Journal of Artificial Societies and Social Simulation*, 5(3), 2002.

[82] João P Hespanha. *Linear Systems Theory*. Princeton Press, Princeton, New Jersey, 2 2018.

[83] G Hewer. An iterative technique for the computation of the steady state gains for the discrete optimal regulator. *IEEE Transactions on Automatic Control*, 16(4):382–384, 1971.

[84] RA Horn and CR Johnson. *Matrix Analysis Second Edition*. 2012.

[85] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[86] Zhongsheng Hou, Huijun Gao, and Frank L Lewis. Data-driven control and learning systems. *IEEE Transactions on Industrial Electronics*, 64(5):4070–4075, 2017.

[87] Yeung Sam Hung and A G MacFarlane. *Multivariable feedback: a quasi-classical approach*. Springer-Verlag New York, Inc., 1982.

[88] W Imrich and S Klavzar. Product graphs: structure and recognition. 2000.

[89] P Ioannou. Decentralized adaptive control of interconnected systems. *IEEE Transactions on Automatic Control*, 31(4):291–298, 1986.

[90] Peng Jia, Anahita MirTabatabaei, Noah E Friedkin, and Francesco Bullo. Opinion Dynamics and the Evolution of Social Power in Influence Networks. *SIAM review*, 57(3):367–397, 2015.

[91] Yu Jiang and Zhong-Ping Jiang. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. *Automatica*, 48(10):2699–2704, 2012.

[92] Merten Jung, Keith Glover, and Urs Christen. Comparison of uncertainty parameterisations for H-inf robust control of turbocharged diesel engines. *Control Engineering Practice*, 13(1):15–25, 2005.

[93] Eugenius Kaszkurewicz and Amit Bhaya. *Matrix diagonal stability in systems and computation*. Springer Science & Business Media, 2012.

[94] Pramod P Khargonekar, Ian R Petersen, and Kemin Zhou. Robust stabilization of uncertain linear systems: quadratic stabilizability and H-infinity control theory. *IEEE Transactions on Automatic control*, 35(3):356–361, 1990.

[95] H J Kim and Andrew Y Ng. Stable adaptive control with online learning. In *Advances in Neural Info. Processing Systems*, pages 977–984, 2005.

[96] Sahin Lale, Kamyar Azizzadenesheli, Babak Hassibi, and Anima Anandkumar. Regret bound of adaptive control in linear quadratic gaussian (LQG) systems. *arXiv preprint arXiv:2003.05999*, 2020.

[97] Luca Lambertini. *Differential Games in Industrial Economics*. Cambridge University Press, 2018.

[98] Peter Lancaster. Explicit solutions of linear matrix equations. *SIAM review*, 12(4):544–566, 1970.

[99] W Levine, T Johnson, and M Athans. Optimal limited state variable feedback controllers for linear systems. *IEEE Transactions on Automatic Control*, 16(6):785–793, 1971.

[100] Frank L Lewis and Kyriakos G Vamvoudakis. Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(1):14–25, 2010.

[101] Frank L Lewis, Hongwei Zhang, Kristian Hengster-Movric, and Abhijit Das. *Cooperative control of multi-agent systems: optimal and adaptive design approaches*. Springer Science & Business Media, 2013.

[102] Jinna Li, Hamidreza Modares, Tianyou Chai, Frank L Lewis, and Lihua Xie. Off-policy reinforcement learning for synchronization in multiagent graphical games. *IEEE transactions on neural networks and learning systems*, 28(10):2434–2445, 2017.

[103] Na Li and Jason R. Marden. Designing games for distributed optimization. *IEEE Journal on Selected Topics in Signal Processing*, 7(2):230–242, 2013.

[104] Zhongkui Li, Zhisheng Duan, Lihua Xie, and Xiangdong Liu. Distributed robust control of linear multi-agent systems with parameter uncertainties. *International Journal of Control*, 85(8):1039–1050, 2012.

[105] Lennart Ljung. System identification. *Wiley encyclopedia of electrical and electronics engineering*, pages 1–19, 1999.

[106] Lennart Ljung. *System Identification*. Wiley Online Library, 2001.

[107] Rogelio Lozano, Pedro Castillo, Pedro Garcia, and Alejandro Dzul. Robust prediction-based control for unstable delay systems: Application to the yaw control of a mini-helicopter. *Automatica*, 40(4):603–612, 2004.

[108] David G Luenberger. Optimization by Vector Space Methods. *Students Quarterly Journal*, 41(162):207, 1970.

[109] Yuwei Luo, Zhuoran Yang, Zhaoran Wang, and Mladen Kolar. Natural actor-critic converges globally for hierarchical linear quadratic regulator. *arXiv preprint arXiv:1912.06875*, 2019.

[110] Krithika Manohar, Eurika Kaiser, Steven L Brunton, and J Nathan Kutz. Optimized sampling for multiscale dynamics. *Multiscale Modeling & Simulation*, 17(1):117–136, 2019.

[111] Karl Mårtensson and Anders Rantzer. Gradient methods for iterative distributed control synthesis. In *Proceedings of the IEEE Conference on Decision and Control*, pages 549–554, 2009.

[112] Paolo Massioni and Michel Verhaegen. Distributed control for identical dynamically coupled systems: A decomposition approach. *IEEE Transactions on Automatic Control*, 54(1):124–135, 2009.

[113] Daniel McFadden. On the controllability of decentralized macroeconomic systems: The assignment problem. In *Mathematical Systems Theory and Economics I/II*, pages 221–239. Springer, 1969.

[114] Kunal Menda, Yi-Chun Chen, Justin Grana, James W Bono, Brendan D Tracey, Mykel J Kochenderfer, and David Wolpert. Deep reinforcement learning for event-driven multi-agent decision processes. *IEEE Transactions on Intelligent Transportation Systems*, 20(4):1259–1268, 2018.

[115] Panayotis Mertikopoulos and Zhengyuan Zhou. Learning in games with continuous action sets and unknown payoff functions. *Mathematical Programming*, 173(1-2):465–507, 1 2019.

[116] Mehran Mesbahi and Magnus Egerstedt. *Graph-Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[117] Carl D Meyer Jr. Generalized inversion of modified matrices. *SIAM Journal on Applied Mathematics*, 24(3):315–323, 1973.

[118] Jacob Levy Moreno. Who Shall Survive?: A New Approach to the Problem of Human Interrelations. *Nervous and Mental Disease Publishing Company*, 1934.

[119] Takehiro Mori, Norio Fukuma, and Michiyoshi Kuwahara. On the discrete Lyapunov matrix equation. *IEEE Transactions on Automatic Control*, 27(2):463–464, 1982.

[120] Nader Motee and Ali Jadbabaie. Optimal control of spatially distributed systems. *IEEE Transactions on Automatic Control*, 53(7):1616–1629, 2008.

[121] D Subbaram Naidu and Anthony J Calise. Singular perturbations and time scales in guidance and control of aerospace systems: A survey. *Journal of Guidance, Control, and Dynamics*, 24(6):1057–1078, 2001.

[122] Kumpati S Narendra and Anuradha M Annaswamy. *Stable adaptive systems*. Courier Corporation, 2012.

[123] Yurii Nesterov and Y Nesterov. Primal-dual subgradient methods for convex problems. *Math. Program., Ser. B*, 120(1 SPEC. ISS.):221–259, 8 2009.

[124] Anh-Thu Nguyen and Mehran Mesbahi. A Factorization Lemma for the Agreement Dynamics. In *IEEE 46th Conference on Decision and Control*, pages 288–293, 2007.

[125] Ann Nowé, Peter Vrancx, and Yann-Michaël De Hauwere. Game theory and multi-agent reinforcement learning. In *Reinforcement Learning*, pages 441–470. Springer, 2012.

[126] Samet Oymak and Necmiye Ozay. Non-asymptotic identification of LTI systems from a single trajectory. In *2019 American Control Conference (ACC)*, pages 5655–5661. IEEE, 2019.

[127] John F Padgett. Introduction to "Marriage and Elite Structure in Renaissance Florence, 1282-1500". 21:33–41, 2011.

[128] H. A. Pahlavan, B. Zahraie, M. Nasseri, and A. Mahdipour Varnousfaderani. Improvement of multiple linear regression method for statistical downscaling of monthly precipitation. *International Journal of Environmental Science and Technology*, 15(9):1897–1912, 9 2018.

[129] Lulu Pan, Haibin Shao, and Mehran Mesbahi. Laplacian Dynamics on Signed Networks. In *Proc. 55th IEEE Conference on Decision and Control*, pages 891–896, Las Vegas, USA, 2016.

[130] Lulu Pan, Haibin Shao, and Mehran Mesbahi. Verification and prediction of structural balance: A data-driven perspective. In *Proc.of the American Control Conference*, pages 2858–2863, Boston, USA, 2016.

[131] Francesca Parise and Asuman Ozdaglar. A variational inequality framework for network games: Existence, uniqueness, convergence and sensitivity analysis. *Games and Economic Behavior*, (2010):1–43, 2019.

[132] Behnoosh Parsa, Keshav Rajasekaran, Franziska Meier, and Ashis G Banerjee. A Hierarchical Bayesian Linear Regression Model with Local Features for Stochastic Dynamics Approximation. *arXiv preprint arXiv:1807.03931*, 2018.

[133] Joshua L. Proctor, Steven L. Brunton, and J. Nathan Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161, 2016.

[134] Anton V Proskurnikov and Roberto Tempo. A Tutorial on Modeling and Analysis of Dynamic Social Networks. {Part I}. *Annual Reviews in Control*, 43:65–79, 2017.

[135] Amirreza Rahmani, Meng Ji, Mehran Mesbahi, and Magnus Egerstedt. Controllability of Multi-Agent Systems From a Graph-Theoretic Perspective. *SIAM Journal on Control and Optimization*, 48(1):162–186, 2009.

[136] J B Rosen. Existence and Uniqueness of Equilibrium Points for Concave N-Person Games. *Econometrica*, 33(3):520–534, 1965.

[137] Michael Rotkowitz and Sanjay Lall. A characterization of convex problems in decentralized control. *IEEE transactions on Automatic Control*, 50(12):1984–1996, 2005.

[138] Farzad Salehisadaghiani and Lacra Pavel. Distributed Nash equilibrium seeking in networked graphical games. *Automatica*, 87:17–24, 2018.

[139] Nils R Sandell Jr, Pravin Varaiya, Michael Athans, and Michael G Safonov. Survey of decentralized control methods for large scale systems. *IEEE Transactions on Automatic Control*, 23(2):108–128, 1978.

[140] Tuhin Sarkar, Alexander Rakhlin, and Munther A Dahleh. Finite-time system identification for partially observed LTI systems of unknown order. *arXiv preprint arXiv:1902.01848*, 2019.

[141] Carsten Scherer and Siep Weiland. Linear matrix inequalities in control. *Lecture Notes, Dutch Institute for Systems and Control, Delft, The Netherlands*, 3(2), 2000.

[142] Mahlagha Sedghi, George Atia, and Michael Georgiopoulos. A multi-criteria approach for fast and outlier-aware representative selection from manifolds. *arXiv preprint arXiv:2003.05989*, 2020.

[143] Miel Sharf and Daniel Zelazo. Network identification: A passivity and network optimization approach. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 2107–2113. IEEE, 2018.

[144] M Simaan, JB Cruz Journal of Optimization Theory Applications, , and undefined 1973. On the Stackelberg strategy in nonzero-sum games. *Springer*.

[145] Max Simchowitz, Stephen Tu, Michael I Jordan, Benjamin Recht, Sebastien Bubeck, Vianney Perchet, and Philippe Rigollet. Learning Without Mixing: Towards A Sharp Analysis of Linear System Identification. Technical report, 7 2018.

[146] J D Simon and Sanjoy K Mitter. A theory of modal control. *Information and Control*, 13(4):316–353, 1968.

[147] Sigurd Skogestad, Kjetil Havre, and Truls Larsson. Control limitations for unstable plants. *IFAC Proceedings Volumes*, 35(1):485–490, 2002.

[148] R Padma Sree and M Chidambaram. *Control of Unstable Systems*. Alpha Science Int'l Ltd., 2006.

[149] Vaibhav Srivastava, Jeff Moehlis, and Francesco Bullo. On bifurcations in nonlinear consensus networks. *Journal of Nonlinear Science*, 21(6):875–895, 12 2011.

[150] Gunter Stein. Respect the unstable. *IEEE Control Systems Magazine*, 23(4):12–25, 2003.

[151] Du$\boldsymbol{v}$sAn M Stipanović, GöKhan Inalhan, Rodney Teo, and Claire J Tomlin. Decentralized overlapping control of a formation of unmanned aerial vehicles. *Automatica*, 40(8):1285–1296, 2004.

[152] Peter Stone and Manuela Veloso. Multiagent systems: a survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383, 2000.

[153] Yvonne R Stürz, Annika Eichler, and Roy S Smith. Distributed control design for heterogeneous interconnected systems. *arXiv preprint arXiv:2004.04876*, 2020.

[154] Richard S Sutton, Andrew G Barto, and Ronald J Williams. Reinforcement learning is direct adaptive optimal control. *IEEE Control Systems Magazine*, 12(2):19–22, 1992.

[155] Shahriar Talebi, Siavash Alemzadeh, Niyousha Rahimi, and Mehran Mesbahi. Online Regulation of Unstable LTI Systems from a Single Trajectory. *arXiv preprint arXiv:2006.00125*, 5 2020.

[156] Shahriar Talebi, Siavash Alemzadeh, Niyousha Rahimi, and Mehrann Mesbahi. Online Regulation of Unstable Linear Systems from a Single Trajectory. In *2020 IEEE 59th Conference on Decision and Control (CDC)*, 2020.

[157] Shahriar Talebi, Siavash Alemzadeh, Lillian J. Ratliff, and Mehran Mesbahi. Distributed Learning in Network Games: A Dual Averaging Approach. In *Proceedings of the IEEE Conference on Decision and Control*, pages 5544–5549, 2019.

[158] Tatiana Tatarenko, Wei Shi, and Angelia Nedic. Accelerated Gradient Play Algorithm for Distributed Nash Equilibrium Seeking. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 3561–3566. IEEE, 2018.

[159] Michael Taylor. Towards a Mathematical Theory of Influence and Attitude Change. *Human Relations*, 21(2):121–139, 1968.

[160] Hamidou Tembine, Dario Bauso, and Tamer Basar. Robust Linear Quadratic Mean-field Games in Crowd-seeking Social Networks. In *IEEE 52nd Annual Conference on Decision and Control (CDC)*, pages 3134–3139, 2013.

[161] A Tsiamis and G J Pappas. Finite Sample Analysis of Stochastic System Identification. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 3648–3654, 2019.

[162] Jonathan H Tu, Clarence W Rowley, Dirk M Luchtenburg, Steven L Brunton, and J Nathan Kutz. On dynamic mode decomposition: theory and applications. *arXiv preprint arXiv:1312.0041*, 2013.

[163] Stephen Tu and Benjamin Recht. The Gap Between Model-Based and Model-Free Methods on the Linear Quadratic Regulator: An Asymptotic Viewpoint. In *Proceedings of the 32nd Conference on Learning Theory*, volume 99, pages 3036–3083. PMLR, 2019.

[164] Henk J Van Waarde, Jaap Eising, Harry L Trentelman, and M Kanat Camlibel. Data informativity: a new perspective on data-driven analysis and control. *IEEE Transactions on Automatic Control*, 2020.

[165] Tamas Vicsek, Andras Czirok, Eshel Ben-Jacob, Inon Cohen, and Ofer Shochet. Novel Type of Phase Transistion in a System of Self-Driven Particles. *New York*, 75(6):1226–1229, 1995.

[166] Andrew Wagenmaker and Kevin Jamieson. Active Learning for Identification of Linear Dynamical Systems. *arXiv preprint arXiv:2002.00495*, 2020.

[167] Shih Ho Wang and E J Davison. On the stabilization of decentralized control systems. *IEEE Transactions on Automatic Control*, 18(5):473–478, 1973.

[168] Wei Wang, Fangfang Zhang, and Chunyan Han. Distributed linear quadratic regulator control for discrete-time multi-agent systems. *IET Control Theory & Applications*, 11(14):2279–2287, 2017.

[169] Yuh-Shyang Wang, Nikolai Matni, and John C Doyle. A system-level approach to controller synthesis. *IEEE Transactions on Automatic Control*, 64(10):4079–4093, 2019.

[170] Christopher J C H Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[171] Matthew O. Williams, Ioannis G. Kevrekidis, and Clarence W. Rowley. A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

[172] Max A Woodbury. Inverting modified matrices. *Memorandum report*, 42(106):336, 1950.

[173] Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2543–2596, 2010.

[174] Nan Xue and Aranya Chakrabortty. Optimal Control of Large-Scale Networks Using Clustering Based Projections. *arXiv:1609.05265*, 2016.

[175] Hai Yang and Sam Yagar. Traffic assignment and signal control in saturated road networks. *Transportation Research Part A*, 29(2):125–139, 1995.

[176] M K Stephen Yeung, Jesper Tegnér, and James J Collins. Reverse engineering gene networks using singular value decomposition and robust regression. *Proceedings of the National Academy of Sciences of the United States of America*, 99(9):6163–6168, 2002.

[177] Kaiqing Zhang, Zhuoran Yang, and Tamer Ba\csar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *arXiv preprint arXiv:1911.10635*, 2019.

[178] Kemin Zhou, John Comstock Doyle, Keith Glover, and others. *Robust and optimal control*, volume 40. Prentice hall New Jersey, 1996.