Springer Theses
Recognizing Outstanding Ph.D. Research

Airlie Chapman

Semi-Autonomous Networks

Effective Control of Networked Systems through Protocols, Design, and Modeling

Springer

Springer Theses

Recognizing Outstanding Ph.D. Research

More information about this series at http://www.springer.com/series/8790

Aims and Scope

The series "Springer Theses" brings together a selection of the very best Ph.D. theses from around the world and across the physical sciences. Nominated and endorsed by two recognized specialists, each published volume has been selected for its scientific excellence and the high impact of its contents for the pertinent field of research. For greater accessibility to non-specialists, the published versions include an extended introduction, as well as a foreword by the student's supervisor explaining the special relevance of the work for the field. As a whole, the series will provide a valuable resource both for newcomers to the research fields described, and for other scientists seeking detailed background information on special questions. Finally, it provides an accredited documentation of the valuable contributions made by today's younger generation of scientists.

Theses are accepted into the series by invited nomination only and must fulfill all of the following criteria

- They must be written in good English.
- The topic should fall within the confines of Chemistry, Physics, Earth Sciences, Engineering and related interdisciplinary fields such as Materials, Nanoscience, Chemical Engineering, Complex Systems and Biophysics.
- The work reported in the thesis must represent a significant scientific advance.
- If the thesis includes previously published material, permission to reproduce this must be gained from the respective copyright holder.
- They must have been examined and passed during the 12 months prior to nomination.
- Each thesis should include a foreword by the supervisor outlining the significance of its content.
- The theses should have a clearly defined structure including an introduction accessible to scientists not expert in that particular field.

Airlie Chapman

Semi-Autonomous Networks

Effective Control of Networked Systems through Protocols, Design, and Modeling

Doctoral Thesis accepted by University of Washington

Springer

Airlie Chapman
Department of Aeronautics and Astronautics
University of Washington
Seattle, WA, USA

ISSN 2190-5053 ISSN 2190-5061 (electronic)
Springer Theses
ISBN 978-3-319-15009-3 ISBN 978-3-319-15010-9 (eBook)
DOI 10.1007/978-3-319-15010-9

Library of Congress Control Number: 2014960279

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media
(www.springer.com)

For my family and my Eric

Supervisor's Foreword

The theme of Dr. Chapman's Ph.D. dissertation is a foundational approach to the design and analysis of control and estimation algorithms for coordinated networked platforms, as well as an effective means of interacting with such complex networked systems. Such multi-platform systems have been identified as the next important milestone for future space, defense, and civilian aerospace and robotic missions, for example, next generation spacecraft architectures collectively referred to as "distributed spacecraft," multiple unmanned aerial systems for search and rescue missions, delivery of medicine in remote areas, and tracking and classification in hazardous and complex environments using multiple ground robots. Specifically, this dissertation aims to present, at a mathematically rigorous level, how one can interface with these networked systems, how secure they are with respect to external influence, how to reason about networks composed of smaller factor networks, and how the network of information exchange between autonomous robots/vehicles can adaptively reconfigure itself such that the ensemble as a whole is more secure and optimally managed.

The dissertation addresses various facets of these issues and much more. For example, Part I clarifies how the mathematical machinery for developing cooperative behavior in linear networks can be extended to nonlinear scenarios, a topic that is of great relevance to areas such as density control, coverage, and air traffic management. Another research area addressed by the dissertation exemplified in Part II is the notion of adaptation in dynamic networks and how it can be implemented using only local information such that the network as a whole resists or promotes an externally injected signal to the network. This work has opened up a new research direction in the field, including how one can reason about self-organization in distributed networked systems: large-scale dynamic systems where local interactions among its subsystems lead to system-wide behaviors and properties. Yet in Part III of the dissertation, a mathematically elegant approach to compose large-scale networks from smaller "atomic" networks such that the system theoretic properties of the composite network can be inferred from those of the atomic ones has been proposed. Such a contribution has far-reaching consequences as it allows systems and control engineers to utilize modularity and scalability in the design and analysis of large-scale dynamic networks, not only in aerospace robotics, but also in completely different contexts such as energy networks and networks-of-networks. Part IV of the dissertation, on the other hand, delves into the notion of structural controllability of networks and important implications for security of networks—once again, a topic of great current technological interest.

The dissertation is a testimony to the creativity of a rising star in control and system theory who, once motivated by a relevant engineering problem, is able to masterfully abstract its essential features and then proceed to address the fundamental issues with mathematical rigor and elegance.

Seattle, WA, USA Mehran Mesbahi

Acknowledgments

I had the pleasure to spend the 5 years of my PhD in Prof. Mehran Mesbahi's lab at the University of Washington (UW). The lab's members and name may have changed, but the warmth and supportive spirit of the group remained throughout. This is a testimony to the atmosphere of camaraderie and encouragement that Prof. Mesbahi exudes. Prof. Mesbahi has inspired me academically, by example, through discussion and, if all of these failed, the offer of a free lunch for "proving him wrong." As an advisor, I still marvel at his ability to correct, guide, and invigorate without me noticing. Prof. Mesbahi is a great mentor, a master at trading compliments, and I am honored to say a great friend. For these reasons and many more, I'd like to thank him for his integral part in making me the researcher and person I am today. I am so grateful that I'll have the opportunity to continue to work with him as a Postdoctoral fellow—it is too early to part!

Thank you to all my lab mates. It has been a joy to share both an office and ideas. Much of the work in this thesis would not have been possible without the collaboration of Prof. Ran Dai, Saghar Hosseini, Dr. Marzieh Nabi-Abdolyousefi, and Eric Schoof. A special thank you to Marzieh for her example of academic excellence and hard work and for not kicking me under the table; Ran for her openness and sharing personality which spread throughout the lab; and Saghar for her gentle spirit, willingness to listen to my constant chatter, and again for the lack of kicks.

I have been fortunate to have taken classes taught by many fantastic UW professors that convey the wonders of teaching. I would especially like to thank my reading and doctoral committee, Profs. Jim Burke, Eric Klavins, Kristi Morgansen, Eric Shea-Brown, and Rekha Thomas. Their enthusiasm and insightful comments opened many new avenues for exciting research. I would like to give special thanks to Prof. Burke for his hours spent sharing with me his contagious passion for optimization.

Finally, I cannot say enough about my family in Australia. My parents have always been a supportive phone call away no matter what the time difference. They have organized my life when I did not have the energy and have proofed thousands of pages of unintelligible garble. My brother, Jason, has provided me with an annual

road trip as an incentive to work toward and, when things got difficult, a flight home to recuperate. My brother, Troy, has been an unwavering model of both a true engineer and a good person. And finally, I cannot possibly say enough about my much better half, Eric. He created for me a home, away from my birthplace, complete with a loving family. We made it through this adventure together, through unending homework, looming deadlines, and sleepless nights at the lab. I would not be here today, or the person I am today, without him.

Contents

Part I Beyond Linear Consensus

Part II Network Measures and Adaptive Topologies

Part III Cartesian Product Networks

Appendix

Abstract

The control and design of multiple autonomous agents for distributed tasks has become a thriving area of aerospace research. An autonomous multi-agent networked system, or just *autonomous network*, is composed of many autonomous vehicles that interact and make local decisions to achieve a common global goal without a centralized body. Advantages of the distributed approach are robustness to individual vehicle failures, improved scalability, and reduced computational load when modeling multi-vehicle systems. A natural extension is to consider autonomous networks that can be influenced or controlled by external agents. We call these systems semi-autonomous networked dynamics systems, or just *semi-autonomous networks*, and they are the focus of this work.

In this dissertation, components of a semi-autonomous network are abstracted away to reason about the effective control of such systems. Specifically, the *agent dynamics* unique to the vehicles of the network is separated from the interaction network dynamics, or simply *network dynamics*, between agents. It is on the network dynamics that we explore effective protocols, design, and modeling for the control of semi-autonomous networks.

Consider a case study of our work: a human operator (controller) directing a cooperative unmanned aerial vehicle (UAV) swarm. The human–UAV and intra-swarm interactions form a *UAV interaction network* and consist of, for example, inter-UAV wireless communications or relative sensor measurements. Factors such as UAV swarm size and dispersion make it infeasible for human operators to interface directly with every UAV. An approach for creating effective interfaces to the controller is to have only a subset of the UAVs directly controlled. The remaining UAVs are then allowed to be indirectly controlled via the UAV interaction network, depicted in Fig. 1. This setup results in a highly coupled high-order nonlinear control model that proves to be difficult for human operators to manage and to reason about in general. The approach we advocate to simplify and create an effective human–swarm interaction is via a two-component hierarchical model consisting of the UAV dynamics component building on the UAV interaction network dynamics component.

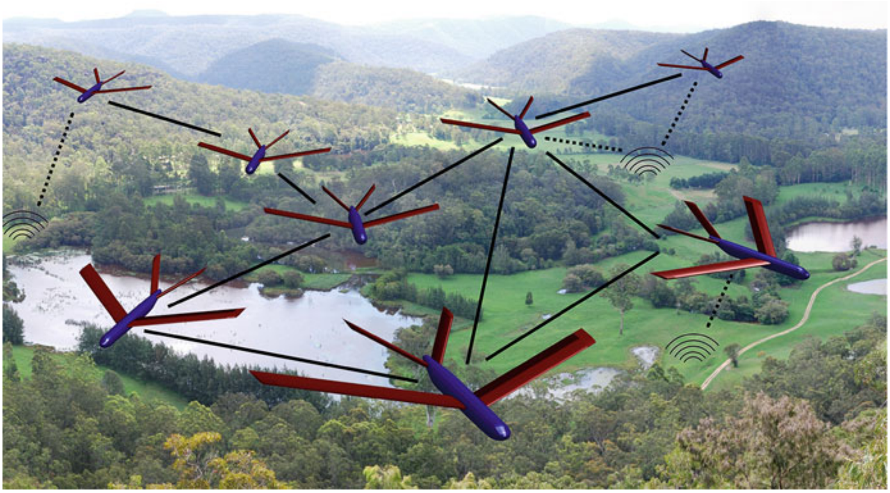


Fig. 1 A UAV swarming concept where only a subset of the UAVs are directly controlled by human operators. These operators influence the remaining UAVs through an interaction network between the vehicles

The UAV interaction network dynamics can be improved from three directions. Firstly, the interaction network dynamics can be exploited through the adoption of alternative *protocols* governing how the UAVs use each other's information. Secondly, through the *design* or redesign of the interaction topology, the propagation of information can be biased towards certain UAVs in the network. Finally, novel network *modeling* approaches can reduce the complexity of a large-scale UAV swarm, improving one's ability to perform analysis and design.

These three directions, protocols, design, and modeling, provide one or more of the building blocks behind the four structured parts of the dissertation. These four parts are addressed separately, together with related publications.

Part I: Beyond Linear Consensus—Protocols

A common objective of autonomous networks is to reach agreement on one or more of their states. For example, agreement on agent position in a UAV swarm leads to rendezvous. If agreeing on a virtual position, formation flight can be acquired given offsets from the virtual position. Velocity agreement is another attractive property for formation flight. For distributed surveillance, bearing agreement is often desired.

If agreement is required over a network without all-to-all communications, a distributed approach is necessary. A protocol which is particularly adept at distributed agreement is discretized version of the heat diffusion dynamics, known as the *consensus protocol*, and runs within the network dynamics. Consensus is a

popular linear protocol for the network dynamics, and consequently it is a rich area of research. This part introduces a new protocol that can be considered "a cousin" of consensus, termed the *advection protocol*.

Advection is the process where a distribution is actively transported by a flow field. A simple example of advection is oil dropped into a body of water. If the water is still, the oil will tend to remain concentrated, but if the water is flowing, for example in a river, the flow will cause a change in concentration. By defining the interactions between agents by a flow field, the advection protocol can be formed.

The advection protocol is shown to share many features of consensus, facilitating the extension of many of the consensus results to advection. Specifically, we compare invariance properties, equilibrium, and flow and stochastic interpretations of the protocols. The key tool in this analysis is the use of graph theory to describe the network interaction topology which underlies the network dynamics. One property distinct from the consensus protocol of particular significance is the conservation of state within the protocol. We illustrate the utility of this property in such applications as sensor coverage and load balancing.

Part I also explores the generalization of the consensus protocol to a family of nonlinear protocols. As coupling nonlinear systems typically generate high-order nonlinear dynamics which are difficult to analyze, nonlinear network dynamics is a largely unexplored research area. The Laplacian matrix, the system matrix at the heart of consensus, lives in the family of M-matrices. We define a class of nonlinear network protocols generalized using M-matrix theory.

We characterize the equilibria of these models as well as their invariant properties. Exploiting these properties, we present and analyze a nonlinear sensor surveillance task, a social networking opinion dynamics, and a neural networking task.

Publications per Chapter

Chapter 1. Advection on Graphs

- Airlie Chapman, Eric Schoof and Mehran Mesbahi (2012) Advection on Networks with an Application to Decentralized Load Balancing, 2680–2681. In *Proc. IEEE Conference on Intelligent Robots and Systems*.
- Airlie Chapman and Mehran Mesbahi (2011) Advection on Graphs, 1461–1466. In *Proc. 50th IEEE Conference on Decision and Control*.

Chapter 2. Beyond Linear Protocols

- Airlie Chapman and Mehran Mesbahi (2012) Stability Analysis of Nonlinear Networks via M-matrix Theory: Beyond Linear Consensus, 6626–6631. In *Proc. American Control Conference*.

Part II: Network Measures and Adaptive Topology—Design

One of the appeals of the consensus protocol is its ability to operate distributively and autonomously over simple trusting agents. This has the added benefit that controller agents, perceived as native agents, can seamlessly attach to the network and steer it in particular directions. When the controller agents are unfriendly, however, this presents an unfavorable perturbation of the dynamics.

This part investigates network measures to quantify the effectiveness of the controller's interfaces with respect to the network dynamics. The performance is measured in terms of the subsequent variation in the mean and variance of the network's agents, termed the *mean tracking measure* (average quadratic error) and *variance damping measure* (open loop $\mathcal{H}_2$ norm). Both measures provide a means of relating topological features of the interaction network structure which promote or deter controllability of the swarm, providing a method to reason about the effective design of network topologies.

Control theory presents many mechanisms to reject or dampen disturbances or promote the propagation of a control signal in a dynamic system. In general, they fall in two categories: active control such as dynamic feedback, and passive control such as structural damping. Typically in networked dynamics systems, the network structure is considered a passive element. If feedback is unavailable or global network knowledge is insufficient to apply feedback, we propose the novel idea of considering the network topology itself as an active system via dynamic selection of interconnection (edge) links and weights.

In situations where the interaction network topology is wireless, it becomes feasible to dynamically redesign the network. We developed distributed protocols that can be employed to dynamically rewire the interaction network topology improving the effectiveness of the controllers' interface with the network. These protocols each *locally* rewire the network topology, using edge trades between neighboring agents, to favorably increase or decrease each measure. The protocol's effectiveness is qualified using the game theoretic construct of the *price of anarchy* by considering the protocol as a non-cooperative game.

Applying online learning techniques, an algorithm was proposed for the agent to locally learn the effect of its interconnection on the various performance measures. Paired with an online optimization method, the edges of the network can be reweighted to reduce the propagation of mis-information from the environment without any global information. This presents a first foray into distributed online learning to optimize the topology in network dynamics.

UAV swarms affected by wind gusts present an attractive and novel application of these rewiring protocols. The wind gust can be considered as a security threat (malicious controller) to the stability of the UAV swarm while humans (friendly controllers) attempt to stabilize the swarm. We have demonstrated that the communication rewiring protocol that allows UAVs to exchange communication links with each other dramatically dampens the wind gust perturbations and amplifies the effectiveness of the human controllers.

Publications per Chapter

Chapter 3. Measures and Rewiring

- Airlie Chapman and Mehran Mesbahi (2013) Semi-Autonomous Consensus: Network Measures and Adaptive Trees, 19–31. In *IEEE Transactions on Automatic Control* 58 (1).
- Airlie Chapman and Mehran Mesbahi (2012) System Theoretic Aspects of Influenced Consensus: Single Input Case, 1505–1511. In *IEEE Transactions on Automatic Control* 57 (6).
- Airlie Chapman and Mehran Mesbahi (2010) Semi-Autonomous Networks: Network Resilience and Adaptive Trees, 7473–7478. In *Proc. 49th IEEE Conference on Decision and Control*.
- Airlie Chapman, Eric Schoof and Mehran Mesbahi (2010) Semi-Autonomous Networks: Theory and Decentralized Protocols, 1958–1963. In *Proc. IEEE International Conference on Robotics and Automation*.
- Airlie Chapman, Marzieh Nabi-Abdolyousefi and Mehran Mesbahi (2009) Identification and Infiltration in Consensus-type Networks, 84–89. In *1st IFAC Workshop on Estimation and Control of Networked Systems*.

Chapter 4. Distributed Online Topology Design for Distributed Rejection

- Airlie Chapman, Eric Schoof and Mehran Mesbahi (2013) Distributed Online Topology Design for Disturbance Rejection, 817–821. In *Proc. 52nd IEEE Conference on Decision and Control*.

Chapter 5. Network Topology Design for UAV Swarming with Wind Gusts

- Airlie Chapman and Mehran Mesbahi (2015) UAV Swarms: Models and Effective Interfaces. In *Handbook of Unmanned Aerial Vehicles*, Springer.
- Airlie Chapman, Ran Dai and Mehran Mesbahi (2011) Network Topology Design for UAV Flocking with Wind Gusts. In *Proc. AIAA Guidance, Navigation and Control Conference*.
- Airlie Chapman and Mehran Mesbahi (2011) UAV Flocking with Wind Gusts: Adaptive Topology and Model Reduction, 1045–1050. In *Proc. American Control Conference*.

Part III: Cartesian Product Networks—Modeling

Scalability is a common challenge when modeling large networks. There are many graph-based tools that can be exploited to factorize the network interaction topology, but the network dynamics can rarely be decomposed in the same manner. This part investigates the Cartesian product of a graph and demonstrates it to be a special operator that decomposes both the network topology and network dynamics. Moreover, this task can be performed efficiently.

Cartesian products form layered structures, often termed *networks-of-networks*. These types of systems are induced by, or eventuate through, many processes. Reasoning about the control of Cartesian networks can be applied to a broad range of topics. Man-made structures, for instance, are commonly fabricated with such a layered form, with applications including fault detection in infrastructure networks, piezoelectric sensor and actuator placement in smart structures, and efficient control of quantum computing networks. Layered structures are often a by-product of the problem formulation itself such as the uniform discretization of PDEs, and as such our theory has application to the control and monitoring of fluid and heat flow. Cartesian products also appear as a result of the analysis of the network, for example through the classification of entities in a network. This occurs in the analysis and influence of opinion dynamics on social networks with agents representing people with common interest groups.

The Cartesian decomposition of these networks is applied to present large semi-autonomous networks as smaller semi-autonomous factor networks. New tools for network trajectory factorization, controllability, and observability follow from the Cartesian product factorization. These tools provide a new computationally efficient method to analyze and design for large-scale networks.

Publications per Chapter

Chapter 6. Cartesian Products of Z-matrix Networks: Factorization and Interval Analysis

- Airlie Chapman, Marzieh Nabi-Abdolyousefi and Mehran Mesbahi (2014) Controllability and observability of networks-of-networks via Cartesian products, 2668–2679. In *IEEE Transactions on Automatic Control* 59 (10).

Chapter 7. Controllability and Observability of Cartesian Product Networks

- Airlie Chapman, Marzieh Nabi-Abdolyousefi and Mehran Mesbahi (2012) Controllability and Observability of Cartesian Product Networks, 80–85. In *Proc. 51st IEEE Conference on Decision and Control.*
- Airlie Chapman and Mehran Mesbahi (2012) Cartesian products of Z-Matrix networks: Factorization and interval analysis. In *Proc. 20th International Symposium on Mathematical Theory of Networks and Systems.*

Part IV: Structural Controllability—Modeling and Design

The concept of controllability is a cornerstone to much of control theory and captures a controller's ability to guide a system towards a desired configuration. *Strong structural controllability* is a generalized form of controllability and can be applied to establish guaranteed conditions for the controllability of network dynamics. These conditions are based solely on the network topology, i.e., who in the network is connected to whom, and is irrespective of the detailed characteristics of these couplings.

Research into structural controllability has a rich history, and by its very nature exposes the role of structure in network control. Structural controllability is based solely on the zero structure of the underlying dynamics, for network dynamics this equates to a relationship on the direct coupling between agents. This is irrespective of the magnitude of these couplings.

This part develops an $\mathcal{O}(n^2)$ algorithm to validate if a set of inputs leads to a strongly structurally controllable network. The research further establishes that finding a strongly controllable minimum cardinality controller set is NP-complete and develops an algorithm that efficiently finds a not necessarily minimal set. This work involves establishing necessary and sufficient conditions for strong structural controllability over different network structures.

These structural controllability results are exploited to reason about the security of networked systems. Vulnerable nodes and critical edges of a network are identified to aid in the design of secure network topologies.

Publications per Chapter

Chapter 8. Strong Structural Controllability of Networked Dynamics

- Airlie Chapman and Mehran Mesbahi (2013) Strong Structural Controllability of Networked Dynamics, 6141–6146. In *Proc. American Control Conference*.

Chapter 9. Security and Infiltration of Networks: A Structural Controllability and Observability Perspective

- Airlie Chapman and Mehran Mesbahi (2013) Security and Infiltration of Networks: A Structural Controllability and Observability Perspective, 143–160. In *Control of Cyber-Physical Systems: Lecture Notes on Control and Information Sciences* 449, Springer.

Final Remarks

Chapter 10 concludes the dissertation with some remarks and directions for future work.

Nomenclature

Linear Algebra

$\lfloor c \rfloor$	largest integer lower bound of scalar c
$[M]_{ij}$	entry of matrix M on its ith row and jth column
$[v]_i$	ith element of vector v
$\approx$	approximately equal to
$\lVert x \rVert_2$	2-norm of vector x; $\lVert x \rVert = (x^T x)^{1/2}$ unless indicated otherwise; also denoted as $\lVert x \rVert$
$\lVert x \rVert_\infty$	∞-norm of vector x; $\lVert x \rVert = \max_i(\lvert x_i \rvert)$ unless indicated otherwise
$\mathbb{E}(x)$	expected value of the random variable x
$\mathrm{sgn}(\cdot)$	signum function
$\lvert \cdot \rvert$	cardinality of a set; also denotes the absolute value of a scalar
$\lambda_i(M)$	ith eigenvalue of M; M is symmetric and its eigenvalue are ordered from least to greatest value
$\lVert M \rVert_2$	2-norm of matrix M
$\mathbf{rank}(M)$	rank of matrix M
$\mathbf{tr}(M)$	trace of M
e^M	matrix exponential of M
$M > 0$	matrix M is positive
$M > N$	matrix $M - N$ is positive
$M \geq N$	matrix $M - N$ is nonnegative
$M \oplus N$	Kronecker sum of M and N
$M \otimes N$	Kronecker product of M and N
$M(\alpha \mid \beta)$	submatrix of matrix $M \in \mathbb{R}^{m \times n}$ formed by removing rows with indices $\alpha \subseteq \{1, \ldots, m\}$ and columns with indices $\beta \subseteq \{1, \ldots, n\}$
$M \geq 0$	matrix M is nonnegative
$M \succ 0$	matrix M is positive definite
$M \succeq 0$	matrix M is positive semidefinite
M^{-1}	inverse of matrix M
M^T	transpose of matrix M
$\mathbf{0}_n$	vector of all zeros of length n; also denoted as $\mathbf{0}$
$\mathbf{1}_n$	vector of all ones of length n; also denoted as $\mathbf{1}$

$\mathbf{1}^\perp$	subspace orthogonal to span $\{\mathbf{1}\}$
e_i	column vector with 1 in the ith row and 0 otherwise
$I_{n\times n}$	$n \times n$ identity matrix; also denoted as I_n and I

Graphs

$\mathcal{G}$	weighted undirected graph or undirected graph; also referred to as a graph
$\mathcal{C}_n$	n node cycle graph, also denoted as $\mathcal{C}$
$\mathcal{D}$	weighted digraph
$\mathcal{G}+e$	graph $\mathcal{G}$ with edge e added; also denoted as $\mathcal{G}\bigcup e$
$\mathcal{G}-e$	graph $\mathcal{G}$ with edge e removed; also denoted as $\mathcal{G}\setminus e$
$\mathcal{H}$	bipartite graph
$\mathcal{K}_n$	n node complete graph, also denoted as $\mathcal{K}$
$\mathcal{P}_n$	n node path graph, also denoted as $\mathcal{P}$
$\mathcal{R}$	external agent graph
$\mathcal{S}_n$	n node star graph, also denoted as $\mathcal{S}$
$\mathcal{T}_n$	n node tree graph, also denoted as $\mathcal{T}$
$\mathcal{D}_1\square\mathcal{D}_2$	Cartesian product of $\mathcal{D}_1$ and $\mathcal{D}_2$
$\mathcal{D}^n$	$\mathcal{D}\square\mathcal{D}\square\ldots\square\mathcal{D}$ (n-times)
$\mathscr{A}(\mathcal{D})$	adjacency matrix of $\mathcal{D}$
$\mathscr{A}(\mathcal{G})$	adjacency matrix of $\mathcal{G}$
$\mathscr{D}(\mathcal{G})$	degree matrix of $\mathcal{G}$
$\mathscr{D}_{\text{in}}(\mathcal{D})$	in-degree matrix of $\mathcal{D}$
$\mathscr{D}_{\text{out}}(\mathcal{D})$	out-degree matrix of $\mathcal{D}$
$\mathscr{D}_s(\mathcal{D})$	self-loop matrix of $\mathcal{D}$
$\mathscr{L}(\mathcal{D})$	in-degree graph Laplacian or Laplacian matrix of $\mathcal{D}$; also denoted as $\mathscr{L}_{\text{in}}(\mathcal{D})$
$\mathscr{L}(\mathcal{G})$	Laplacian matrix of $\mathcal{G}$
$\mathscr{L}_{\text{out}}(\mathcal{D})$	out-degree graph Laplacian or Laplacian matrix of $\mathcal{D}$
$\mathscr{W}(\mathcal{G})$	weight matrix of $\mathcal{G}$
$\tilde{\delta}_i$	out-degree of node v_i
δ_i	in-degree of node v_i; also denotes the degree of node v_i for undirected graphs
δ_i^r	number of leaders adjacent to v_i
δ_i^v	number of followers adjacent to r_i
$\mathcal{I}(v_i)$	set of all neighbors of v_i closer to agents in R than v_i
$\mathcal{N}(v_i)$	set of nodes adjacent to i
$\mathcal{N}_{\text{out}}(v_i)$	out-degree neighborhood set of v_i
$\mathcal{N}_{\text{in}}(v_i)$	in-degree neighborhood set of v_i
$\sigma(\cdot)$	edge set index mapping function; $l=\sigma(ij)$ if edge l connects nodes i and j
$\Gamma(v_i)$	closest agent to v_i in $\mathcal{M}$
a_l	edge column of edge $l=\sigma(ij)$, also denoted as a_{ij}
$c(v_i,\mathcal{G})$	closeness centrality of node v_i in graph $\mathcal{G}$
$d(v_i,v_j)$	minimum path length between v_i and v_j

$E_{\text{eff}}(v_i)$	effective resistance at node v_i
v_i	vertex i; also used to denote the ith entry of vector v
$\lambda_i(\mathcal{G})$	shorthand for $\lambda_i(\mathscr{L}(\mathcal{G}))$
$\mathcal{E}_R$	external agent edge set
$\mathcal{M}$	main path agent set
$\mathcal{R}^i$	external agent graph with a single input agent attached at node v_i
$\mathcal{R}_c$	common external agent set
$\mathcal{R}_d$	distinct external agent set
$\text{Aut}(\mathcal{D})$	set of automorphisms of graph $\mathcal{D}$
$\det(\mathcal{D})$	determining number of digraph $\mathcal{D}$
$\text{stab}(S)$	stabilizing set of S
$\pi(\mathcal{E}_R)$	set of native agents that directly connect to external agents
σ	permutation set
E	edge set
R	external agent set
V	node set; also referred to as vertex set
V_s	set of nodes with self-loops

Other

$A(\mathcal{G}, \mathcal{R})$	influenced state matrix of graph $\mathcal{G}$ and external agent graph $\mathcal{R}$
$B(\mathcal{R})$	influenced input matrix of external agent graph $\mathcal{R}$
$B_n(S)$	input matrix of the input set S for a n node digraph; also referred to as $B(S)$
$C_n(S)$	output matrix of the output set S for a n node digraph; also referred to as $C(S)$
$J_\mu(\mathcal{G}, \mathcal{R})$	mean tracking measure of graph $\mathcal{G}$ and external agent graph $\mathcal{R}$
$J_\sigma(\mathcal{G}, \mathcal{R})$	variance damping measure of graph $\mathcal{G}$ and external agent graph $\mathcal{R}$
$P(\mathcal{G}, \mathcal{R})$	controllability gramian of the influenced dynamics with graph $\mathcal{G}$ and external agent graph $\mathcal{R}$; also denoted as $P(A(\mathcal{G}, \mathcal{R}), B(\mathcal{R}))$
$x_i(t)$	state of agent i at time t
$\boldsymbol{A}(\mathcal{D})$	pattern matrix of digraph $\mathcal{D}$
$\boldsymbol{A}_\oplus$	family of symmetry preserving, Cartesian invariant matrix representations
$\boldsymbol{A}_\times$	pattern matrix formed by placing stars along the diagonal of $\boldsymbol{A}$
$\boldsymbol{B}_n(S)$	pattern matrix of input matrix $B_n(S)$; also referred to as $\boldsymbol{B}(S)$
$\boldsymbol{C}_n(S)$	pattern matrix of output matrix $C_n(S)$; also referred to as $\boldsymbol{C}(S)$
$\circledast$	zero or star elements of a pattern matrix
$[\underline{M}, \overline{M}]$	interval matrix with lower bound $\underline{M}$ and upper bound $\overline{M}$
$\times$	star elements of a pattern matrix

Background

The first step to investigating autonomous networked dynamics systems is to characterize the underlying network topology and network dynamics. To this end, we introduce notation and a graph theoretic presentation of the network topology and a popular network dynamics protocol—the consensus protocol.

Notation

This section introduces notation used throughout this thesis.

For a column vector $v \in \mathbb{R}^p$, v_i or $[v]_i$ denotes the ith element. The ijth element of matrix M is $[M]_{ij}$. The term e_i denotes the column vector which contains all zero entries except $[e_i]_i = 1$. For matrices $M, N \in \mathbb{R}^{n\times n}$, $N \preceq M$ denotes that $M - N$ is positive semidefinite. We form the submatrix $A(\alpha|\beta)$ from $A \in \mathbb{R}^{m\times n}$, where $\alpha \subseteq \{1, \dots, m\}$ and $\beta \subseteq \{1, \dots, n\}$, by removing rows with indices α and columns with indices β.

The notation $\|\cdot\|$ and $\|\cdot\|_\infty$ denotes the 2-norm and infinity norm, respectively; $\mathbf{tr}(\cdot)$ denote the trace of a matrix; $|\cdot|$ denotes the cardinality of a set; $\mathbf{1}$ and $\mathbf{0}$ denotes the column vector of ones and zeros, respectively; I denotes the identity matrix.

A matrix M is nonnegative (positive), denoted $M \geq 0$ ($M > 0$) if all entries of M are nonnegative (positive). Further, $M \geq N$ ($M > N$) is equivalent to $M - N \geq 0$ ($M - N > 0$). We write $M \succ 0$ ($M \succeq 0$) if M is positive definite (semidefinite) matrix.

The Kronecker product of matrices A and B is denoted by $A \otimes B$, and the Kronecker sum of square matrices $C \in \mathbb{R}^{n\times n}$ and $D \in \mathbb{R}^{m\times m}$ defined and denoted as $C \oplus D := I_m \otimes C + D \otimes I_n$. The matrix exponential of a Kronecker sum has the attractive distributive property that $e^{C\oplus D} = e^C \otimes e^D$. The Hadamard product of two matrices is denoted by $\circ$ where $[M \circ N]_{ij} = [M]_{ij} \cdot [N]_{ij}$.

Network Topology

To represent the network topology, the communication points in the network are denoted as nodes, and edges represent communication links between points. For UAV swarms, the nodes represent UAVs and the edges correspond to inter-UAVs links, such as wireless communications or relative sensors. It is assumed that all communication links are bi-directional. These nodes and edges can be considered as forming an undirected graph.

Abstractly, a weighted undirected graph $\mathcal{G} = (V, E, W)$ is defined by a node set V with cardinality n, the number of nodes in the graph, and an edge set E and a weight set W. The edge set is comprised of pairs of nodes, where nodes v_i and v_j are adjacent if $\{v_i, v_j\} \in E \subseteq [V]^2$, the set of two-element subsets of V with each edge having weight $w_{ij} \in W$. If $w_{ij} = 1$ for all edges, then $\mathcal{G} = (V, E)$ is a unweighted undirected graph or simply an undirected graph. The ordering of the edge set is encoded through the index mapping $\sigma(\cdot)$ such that $l = \sigma(ij)$ if and only if edge l connects nodes i and j. If clear, vectors pertaining to the edges such as the edge weights will be denoted as w_{ij} and w_l, interchangeably.

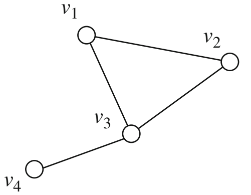
One special family of undirected graphs are *tree graphs*, denoted by the set $\mathcal{T}$, where all two node pairs are connected by exactly one simple path, i.e., a connected graph without cycles. A *spanning tree* of a graph is a tree subgraph that connects all vertices in the graph. Special undirected graphs are the n node complete graph $\mathcal{K}_n$ which has an edge between every node, the n node path graph $\mathcal{P}_n$ where $(i, j) \in E$ if and only if $|i - j| = 1$, the n node cycle graph $\mathcal{C}_n$ which is formed by adding edge $(1, n)$ to $\mathcal{P}_n$ and the star graph $\mathcal{S}_n$ where $(1, j) \in E$ if and only if $j = 2, \ldots, n$.

The *neighborhood set* $\mathcal{N}(v_i)$ is composed of the set of nodes adjacent to v_i. The scalar $d\left(v_i, v_j\right)$ is the minimum path length, induced by the graph $\mathcal{G}$, between nodes v_i and v_j. The degree δ_i of node v_i is the number of its adjacent nodes. The *degree matrix* $\mathscr{D}(\mathcal{G}) \in \mathbb{R}^{n \times n}$ is a diagonal matrix with δ_i at entry (i, i). The *weight matrix* $\mathscr{W}(\mathcal{G})$ is an $m \times m$ diagonal matrix with the weight of edge l at position $[\mathscr{W}(\mathcal{G})]_{ll}$.

There are many matrix representations of a graph; the two most popular are the adjacency matrix and graph Laplacian matrix that each fully characterizes the graph [1, 2]. The *adjacency matrix* is an $n \times n$ symmetric matrix with $[\mathscr{A}(\mathcal{G})]_{ij} = w_{ij}$ when $\{v_i, v_j\} \in E$ and $[\mathscr{A}(\mathcal{G})]_{ij} = 0$ otherwise. The *incidence matrix* $\mathscr{E}(\mathcal{G})$ is an $n \times m$ composed of edge columns a_l which encodes the edge $l = \sigma(ij)$ with $[a_l]_i = 1$ and $[a_l]_j = -1$, and zero otherwise.[1] The *graph Laplacian matrix* (or just Laplacian) is defined as $\mathscr{L}(\mathcal{G}) = \mathscr{E}(\mathcal{G})\mathscr{W}(\mathcal{G})\mathscr{E}(\mathcal{G})^T \in \mathbb{R}^{n \times n}$, or equivalently for unweighted graphs $\mathscr{L}(\mathcal{G}) = \mathscr{D}(\mathcal{G}) - \mathscr{A}(\mathcal{G})$. Figure 2 shows an example of unweighted undirected graph with its corresponding graph matrices.

The graph Laplacian matrix plays an important role in the dynamics of the network. An important feature of this matrix is that it is a (symmetric) positive semi-definite matrix. The spectrum is ordered as $0 = \lambda_1(\mathcal{G}) \leq \lambda_2(\mathcal{G}) \leq \cdots \leq \lambda_n(\mathcal{G})$,

[1] The assignment where $[a_l]_j = 1$ and $[a_l]_i = -1$ can also be applied.



$$\mathscr{A}(\mathcal{G}) = \begin{bmatrix} 0&1&1&0 \\ 1&0&1&0 \\ 1&1&0&1 \\ 0&0&1&0 \end{bmatrix}, \mathscr{D}(\mathcal{G}) = \begin{bmatrix} 2&0&0&0 \\ 0&2&0&0 \\ 0&0&3&0 \\ 0&0&0&1 \end{bmatrix}, \mathscr{L}(\mathcal{G}) = \begin{bmatrix} 2&-1&-1&0 \\ -1&2&-1&0 \\ -1&-1&3&-1 \\ 0&0&-1&1 \end{bmatrix}$$

Fig. 2 A sample network graph $\mathcal{G} = (V, E)$ where $V = \{v_1, v_2, v_3, v_4\}$ and $E = \{\{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_3\}, \{v_3, v_4\}\}$. The adjacency matrix $\mathscr{A}(\mathcal{G})$, degree matrix $\mathscr{D}(\mathcal{G})$, and Laplacian matrix $\mathscr{L}(\mathcal{G})$ of the graph are provided

where, for brevity, $\lambda_i(\mathcal{G})$ is used instead of $\lambda_i(\mathscr{L}(\mathcal{G}))$. The spectrum of the graph Laplacian matrix displays many features of the network. For example, the number of zero eigenvalues is equal to the number of connected components. Features that couple the network topology and dynamics are explored in the following subsection.

A generalization of a graph is a weighted digraph. A weighted digraph $\mathcal{D} = (V, E, W)$ is characterized by a node set V with cardinality n, an edge set E comprised of ordered pairs of nodes with cardinality m, and a weight set W with cardinality m. The adjacency matrix is an $n \times n$ matrix with $[\mathscr{A}(\mathcal{D})]_{ij} = w_{ij} \in W$ when $(j, i) \in E$ and $[\mathscr{A}(\mathcal{D})]_{ij} = 0$ otherwise. The self-loop matrix $\mathscr{D}_s(\mathcal{D}) \in \mathbb{R}^{n \times n}$ is a diagonal matrix with w_{ii} at position (i, i) and the set of nodes with self-loops is V_s. The in-degree matrix $\mathscr{D}_{\text{in}}(\mathcal{D}) \in \mathbb{R}^{n \times n}$ is a diagonal matrix with the number of edges incident to node i, its in-degree $\delta_i = \sum_{(j,i) \in E} w_{ij}$, at position (i, i). The out-degree matrix $\mathscr{D}_{\text{out}}(\mathcal{D})$ is similarly defined, using the out-degree $\tilde{\delta}_i = \sum_{(i,k) \in E} w_{ki}$. The in-degree graph Laplacian (or Laplacian) matrix $\mathscr{L}_{\text{in}}(\mathcal{D})$ (or $\mathscr{L}(\mathcal{D})$) is defined as $[\mathscr{L}(\mathcal{D})]_{ij} = -[\mathscr{A}(\mathcal{D})]_{ij}$ for $i \neq j$ and $[\mathscr{L}(\mathcal{D})]_{ii} = [\mathscr{D}_{\text{in}}(\mathcal{D})]_{ii}$. The out-degree graph Laplacian $\mathscr{L}_{\text{out}}(\mathcal{D})$ is similarly defined. The in-degree neighborhood set $\mathcal{N}_{\text{in}}(v_i)$ is composed of the set of nodes attached to in-degree edges of v_i. Similarly, $\mathcal{N}_{\text{out}}(v_i)$ is defined in terms of the out-degree edges of v_i.

A special family of digraphs is the strongly connected graphs, where a digraph is strongly connected if between every pair of distinct nodes there exists a directed path of edges. A graph is called balanced if $\mathscr{D}_{\text{in}}(\mathcal{D}) = \mathscr{D}_{\text{out}}(\mathcal{D})$. The node set $V_1 \subseteq V$ is a rooted node set for $V_2 \subseteq V$ if every node in V_2 can be reached from at least one node in V_1 along a directed path. Further, if V_1 is (not) a rooted node set for V, then $\mathcal{D}$ is input *(in)accessible* from V_1. A rooted directed tree graphs is a subgraph where $m = n - 1$, and there exists a node i such that $\{i\}$ is input accessible.

Consensus Dynamics

A common objective of autonomous networks is to reach agreement on one or more of their states. For example, agreement on UAV position in a UAV swarm achieves rendezvous. If agreeing on a virtual position, formation flight can be acquired with known position offsets from the virtual position. Velocity agreement is another attractive property for formation flight. For distributed surveillance, bearing agreement is often desired.

If agreement is required over a network without all-to-all communications, a distributed approach is necessary. A protocol which is particularly adept at distributed agreement is the consensus protocol, which is detailed below, and runs within the network dynamics.

Consider $x_i(t) \in \mathbb{R}$ to be the ith node's state at time t on which agreement is required for all nodes. The continuous-time consensus dynamics is defined over the undirected graph $\mathcal{G} = (V, E, W)$ as

$$\dot{x}_i(t) = \sum_{\{v_i, v_j\} \in E} w_{ij} \left(x_j(t) - x_i(t) \right). \tag{1}$$

Thus, to update the ith node's state, only the relative state of node i's neighbor's state is required. In a compact form with $x(t) \in \mathbb{R}^n$, the collective dynamics is represented as

$$\dot{x}(t) = -\mathscr{L}(\mathcal{G})x(t). \tag{2}$$

From the definition of the graph Laplacian all rows of $\mathscr{L}(\mathcal{G})$ sum to zero and $\lambda_1(\mathcal{G}) = 0$ with the corresponding eigenvector $\mathbf{1}_n$. Subsequently, for a connected graph $\mathcal{G}$ the network dynamics will converge to an agreement on the state, i.e., $x_1(t) = x_2(t) = \cdots = x_n(t) = \alpha$, for some constant α, for all initial conditions [3]. Further, the slowest convergence of the dynamics is determined by $\lambda_2(\mathcal{G})$ which is a measure of graph connectivity.

For digraph $\mathcal{D} = (V, E, W)$, the consensus dynamics takes the form

$$\dot{x}_i(t) = \sum_{(v_j, v_i) \in E} w_{ij} \left(x_j(t) - x_i(t) \right),$$

and collectively

$$\dot{x}(t) = -\mathscr{L}_{\text{in}}(\mathcal{D})x(t).$$

Features of consensus over digraphs are discussed in detail in Chap. 1.

References

1. B. Bollobás, *Modern Graph Theory*. New York: Springer, 1998.
2. R. Diestel, *Graph Theory*. Berlin: Springer, 2000.
3. R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

Part I
Beyond Linear Consensus

Chapter 1
Advection on Graphs

Abstract This chapter examines the dynamics of a networked, multi-agent system operating with an advection-based coordination algorithm. Flow advection is a close relative of diffusion whose discretized version forms the basis of the popular consensus dynamics. We endeavor to demonstrate in this chapter that discretizing the continuous advection equation also forms an attractive set of system dynamics for coordinated control. The key advantage of advection-based algorithms over directed consensus is that the sum of the states is always conserved. This chapter includes a formulation of the advection dynamics on directed graphs and a presentation of some of its characteristics, which are compared to the consensus dynamics. We also provide examples of the versatility of the advection dynamics: a formation control and sensor coverage example.

1.1 Introduction

Advection is the process where a distribution is actively transported by a flow field. A simple example of advection is oil dropped into a body of water. If the water is still, the oil will tend to remain concentrated, but if the water is flowing, for example in a river, the flow will cause a change in concentration. This chapter introduces a discrete form of the advection process for application in networked, multi-agent systems. Here a flow field defines interactions between agents, and inter-agent dynamics are based on the advection process.

Advection shares many similarities to diffusion and may be interpreted as diffusion in a flow field. A discretized form of diffusion is the framework for consensus problems. Consensus provides an effective model for distributed information-sharing and control of networked, multi-agent systems in settings such as multi-vehicle control, formation control, swarming, and distributed estimation; see, for example, [1–4]. An appeal of the consensus framework is that locally-based interaction dynamics can produce global network characteristics. Further, the performance characteristics are coupled to the underlying network structure. Advection dynamics are similarly coupled to the network, and in certain classes of networks, share identical dynamics with consensus.

At the heart of consensus problems is the diffusion model [5]. For undirected consensus, where the underlying agent interactions are Euclidean-based, the governing

A. Chapman, *Semi-Autonomous Networks*, Springer Theses,
DOI 10.1007/978-3-319-15010-9_1

dynamics are precisely the discrete version of diffusion. The two core properties and attractions of discrete diffusion are (a) the interpretation of a directed edge existing from agent i to j is that agent i can "influence" agent j and (b) the state sum is always conserved. For directed graphs (digraphs), where every edge from i to j does not necessarily have a corresponding edge from j to i, generally one of these properties has to be sacrificed. Traditionally, graph literature [1, 6] preferentially chooses the "influence" property and the sum conservation property only holds true for balanced digraphs, which corresponds to the "perfect" discretization of diffusion with both conditions. In this chapter, we will instead preferentially choose the second conservation property and sacrifice at times the "influence" property and formulate our model accordingly. The sum conservation property has the effect of inducing a flow through the directed edges in the digraph and in turn corresponds to the discrete version of the advection model. Advection has been used to model the spread of diseases [6], population migration [7], and supply and demand in economic systems [8].

The organization of the chapter is as follows. We begin by defining advection dynamics and characterize its state matrix, dynamics and equilibrium, with a particular focus on the underlying digraph structure. We then compare and contrast the advection features with traditional consensus dynamics and illustrate the strong link between them. Finally, we examine two advection problems.

The advection equation, also known as the transport equation, involves a scalar concentration u of a material affected by a flow field $\vec{v}$ and is conventionally [5] given by

$$\frac{\partial u}{\partial t} = -\nabla \cdot \left(\vec{v}u\right).$$

Here ∇ is the divergence operator defined on a continuously differentiable vector field $F = \sum_{i=1}^{k} U_i a_i$ with basis vectors a_i and coordinate frame $\{x_1, x_2, \ldots, x_k\}$ in $\mathbb{R}^k$, and defined as $\nabla \cdot F = \sum_{i=1}^{k} \frac{\partial U_i}{\partial x_i}$. The flux of the advection process is subsequently $F = \vec{v}u$.

In a discrete calculus analogue of the advection equation, we first define an interaction digraph (directed and weighted) over nodes based on the flow $\vec{v}$. The flow vector $\vec{v}$ dictates the interactions between nodes by defining directed edges and edge weights. We then adopt a discretized view of the flux $\vec{v}u$ through an edge $i \to j$ as consisting of the flow w_{ji} prescribed by $\vec{v}$ at the edge modified by the concentration x_i prescribed by u at node i. The flow along edge $i \to j$ is consequently $w_{ij}x_i(t)$. The concentration at node i at time t is denoted $x_i(t)$. The rate of change of the concentration of node i is then the flow into the node minus the flow out of the node, i.e.,

$$\dot{x}_i(t) = -\sum_{\{\forall j \mid i \to k\}} w_{ki} x_i(t) + \sum_{\{\forall j \mid j \to i\}} w_{ij} x_j(t). \tag{1.1}$$

This problem is well suited to a graph theoretic analysis.

We can now rewrite our dynamics (1.1) using these digraph concepts with the flow $\vec{v}$ generating the digraph $\mathcal{D}\left(\vec{v}\right) = (V, E\left(\vec{v}\right), W\left(\vec{v}\right))$ as

$$\dot{x}_i(t) = -\tilde{\delta}_i x_i(t) + \sum_{(j,i)\in E} w_{ij} x_j(t). \tag{1.2}$$

For brevity, we will denote the digraph as $\mathcal{D} = (V, E, W)$. The advection dynamics can therefore be written as

$$\dot{x}(t) = -\mathscr{L}_{\text{out}}(\mathcal{D})x(t). \tag{1.3}$$

We now proceed to examine system characteristics of the advection dynamics and compare them with consensus dynamics. Specifically, we compare invariance properties, equilibrium, and flow and stochastic interpretations of the dynamics where $\mathcal{D}$ contains a rooted branching. A road map of these features are depicted in Fig. 1.1.

1.2 Advection Properties

We will now proceed to examine some of the characteristics of advection and compare them with the more familiar consensus dynamics. As described in the Introduction chapter, the node dynamics for consensus can be written as

$$\begin{aligned}\dot{x}_i(t) &= \sum_{(j,i)\in E} w_{ij}\left(x_j(t) - x_i(t)\right) \\ &= -\delta_i x_i(t) + \sum_{(j,i)\in E} w_{ij} x_j(t). \end{aligned} \tag{1.4}$$

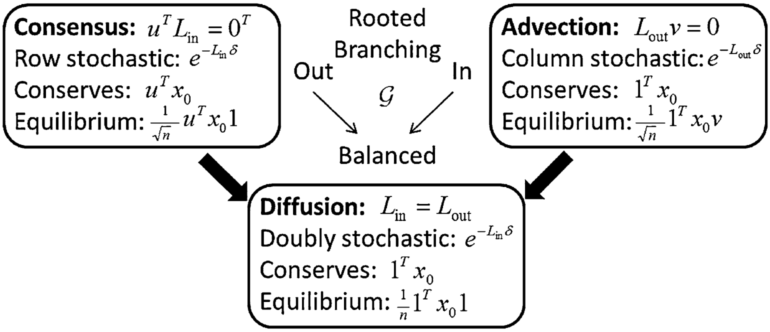

Consensus: $u^T L_{\text{in}} = 0^T$
Row stochastic: $e^{-L_{\text{in}}\delta}$
Conserves: $u^T x_0$
Equilibrium: $\frac{1}{\sqrt{n}} u^T x_0 1$
Rooted Branching
Out
In
$\mathcal{G}$
Balanced
Advection: $L_{\text{out}} v = 0$
Column stochastic: $e^{-L_{\text{out}}\delta}$
Conserves: $1^T x_0$
Equilibrium: $\frac{1}{\sqrt{n}} 1^T x_0 v$
Diffusion: $L_{\text{in}} = L_{\text{out}}$
Doubly stochastic: $e^{-L_{\text{in}}\delta}$
Conserves: $1^T x_0$
Equilibrium: $\frac{1}{n} 1^T x_0 1$

Fig. 1.1 Relationship between rooted out-branching consensus, rooted in-branching advection and rooted in/out-branching balanced consensus (in the discretization sense—"perfect" diffusion—exhibiting properties of both consensus and advection)

in compact form, the consensus dynamics are

$$\dot{x}(t) = -\mathscr{L}_{\text{in}}(\mathcal{D})x(t). \tag{1.5}$$

The first significant difference between the advection and consensus dynamics is the sum conservation property of advection, stated in the following proposition, which is generally not a property of directed consensus.

Proposition 1.1. *The advection dynamics (1.3) are (state) sum conservative, i.e.,* $\sum_{i=1}^{n} x_i(t) = \sum_{i=1}^{n} x_i(0)$ *for all time* t.

Proof. Directly from (1.2),

$$\begin{aligned}\sum_{i=1}^{n} \dot{x}_i(t) &= \sum_{i=1}^{n} \left(- \sum_{(i,k)\in E} w_{ki} x_i(t) + \sum_{(j,i)\in E} w_{ij} x_j(t) \right) \\ &= \sum_{(j,i)\in E} \left(-w_{ij} x_i(t) + w_{ij} x_i(t) \right) = 0.\end{aligned}$$

Therefore, $\sum_{i=1}^{n} x(t)$ is always conserved. □

An alternate interpretation of the sum conservation property is that the mean of the states is always constant. Consensus also conserves a weighted sum of the initial states. This feature will be stated later in Proposition 1.10.

Another difference between these dynamics is that for consensus the interaction mechanics between neighboring agents where the dynamics of agent i can only "influence" agent j directly if there is an edge (i, j) in the digraph $\mathcal{D}$. This is generally not the case for advection. The advantage of this "influence" property means that the consensus dynamics are only based on relative states between neighboring agents as represented in (1.4); an absolute reference frame is not required. Consequently, the consensus dynamics can be driven by sensors that can only measure relative states, e.g., if x_i is the position of agent i, distance sensors mounted on agent i can measure the relative position to agent j as $x_j - x_i$. For advection, unless $\delta_i = \tilde{\delta}_i$, the dynamics of agent i can not be represented as purely a sum of relative states and so agent i must have knowledge of its state in a global frame.

We hope to demonstrate that for some applications the cost of maintaining an absolute frame is minimal and worth the benefits of conservation exhibited by the advection dynamics. The first noteworthy application is for information networks where an agent's information state is communicated between agents, e.g., a wireless sensor network where x_i is the ith sensor's state and agent j transmits its state to agent i if $(i, j) \in E$. Provided agent i knows its weighted "influence" on its neighbors $\tilde{\delta}_i$, no further information is required for the advection dynamics. The advection dynamics also has an extension to heterogeneous networks with more complex agents, able to sense their own state, selecting arbitrary out-degree edge weights, and simpler agents without global state and/or $\tilde{\delta}_i$ knowledge, selecting weights such that $\delta_i = \tilde{\delta}_i$. It will be shown in the following proposition that if the digraph is balanced, then advection dynamics is the same as consensus dynamics. For unbalanced digraphs, there can be as many as $n - 2$ agents in the network with advection dynamics identical to consensus dynamics requiring only relative state measurements, with only the remaining two agents with non-identical advection dynamics requiring knowledge of their state in a global frame.

Proposition 1.2. *The advection dynamics (1.3) and consensus dynamics (1.5) are identical for all initial $x(0)$ if and only if the underlying digraph $\mathcal{D}$ is balanced.*

Proof. The dynamics are identical for all initial $x(0)$ if and only if $\mathscr{L}_{\text{in}}(\mathcal{D}) = \mathscr{L}_{\text{out}}(\mathcal{D})$, if and only if $\mathscr{D}_{\text{in}}(\mathcal{D}) - \mathscr{A}(\mathcal{D}) = \mathscr{D}_{\text{out}}(\mathcal{G}) - \mathscr{A}(\mathcal{D})$, if and only if $\mathscr{D}_{\text{in}}(\mathcal{D}) = \mathscr{D}_{\text{out}}(\mathcal{D})$, if and only if $\delta_i = \tilde{\delta}_i$ for all $i, j \in V$, i.e., $\mathcal{D}$ is balanced. □

An interpretation of advection over balanced digraphs is that the flow divergence at each node is always zero which reduces the dynamics to diffusion. A consequence of Proposition 1.2 is that one can consider balanced digraphs as the "perfect" discretization of the diffusion dynamics with both the sum conservation and the "influence" property.

1.2.1 Out-Degree Laplacian

Many of the properties of consensus dynamics (1.5) are well understood and it is useful to relate them to advection dynamics (1.3) using their respective state matrices $\mathscr{L}_{\text{in}}(\mathcal{G})$ and $\mathscr{L}_{\text{out}}(\mathcal{G})$. We proceed to accomplish this by using the notion of a reverse digraph $\mathcal{D}^r = (V, E^r, W^r)$ formed by reversing the edges of a digraph $\mathcal{D} = (V, E, W)$, i.e., $(i, j) \in E$ with $w_{ji} \in W \iff (j, i) \in E^r$ with $w_{ij} \in W^r$. We then have the following proposition:

Proposition 1.3. *For a digraph $\mathcal{D}$ and corresponding reverse digraph $\mathcal{D}^r$, $\mathscr{L}_{\text{out}}(\mathcal{D}) = \mathscr{L}_{\text{in}}(\mathcal{D}^r)^T$.*

Proof. Directly from the definition of the reverse digraph, we note that the out-degree of node i in $\mathcal{D}$ is the in-degree of node i in $\mathcal{D}^r$, consequently $\mathscr{D}_{\text{in}}(\mathcal{D}) = \mathscr{D}_{\text{out}}(\mathcal{D}^r)$. Further, $[\mathscr{A}(\mathcal{D})]_{ij} = w_{ij} = [\mathscr{A}(\mathcal{D}^r)]_{ji}$, so $\mathscr{A}(\mathcal{D}) = \mathscr{A}(\mathcal{D}^r)^T$. Therefore, noting that $\mathscr{D}_{\text{in}}(\mathcal{D})$ is a symmetric matrix, $\mathscr{L}_{\text{out}}(\mathcal{D}) = \mathscr{D}_{\text{out}}(\mathcal{D}) - \mathscr{A}(\mathcal{D}) = \mathscr{D}_{\text{in}}(\mathcal{D}^r) - \mathscr{A}(\mathcal{D}^r)^T = (\mathscr{D}_{\text{in}}(\mathcal{D}^r) - \mathscr{A}(\mathcal{D}^r))^T = \mathscr{L}_{\text{in}}(\mathcal{D}^r)^T$. □

Using Proposition 1.3, we adapt existing properties of $\mathscr{L}_{\text{in}}(\mathcal{D})$ to properties of $\mathscr{L}_{\text{in}}(\mathcal{D})$. Let us define a classifications of digraphs which we use in subsequent propositions.

Definition 1.4. A digraph $\mathcal{D}$ is a *rooted branching* if

(1) it does not contain a directed cycle and

(2) it has a node v_r (root) such that

(a) for every other node $v \in \mathcal{D}$ there is a directed path from v_r to v, in which case it is called a *rooted out-branching* or
(b) for every other node $v \in \mathcal{D}$ there is a directed path from v to v_r, in which case it is a *rooted in-branching*.

We now state some known properties of $\mathscr{L}_{\text{in}}(\mathcal{D})$ and equivalent properties of $\mathscr{L}_{\text{out}}(\mathcal{D})$ for the cases where $\mathcal{D}$ contains a rooted out-branching, rooted in branching, and/or when $\mathcal{D}$ is balanced. Let $\mathscr{L}_{\text{in}}(\mathcal{G}) = P_{\text{in}} J_{\text{in}}(\Lambda_{\text{in}}) P_{\text{in}}^{-1}$ be the Jordan decomposition of $\mathscr{L}_{\text{in}}(\mathcal{D})$, P_{in} a nonsingular matrix with normalized columns, and Λ_{in} the eigenvalues of $\mathscr{L}_{\text{in}}(\mathcal{D})$. Similarly, $\mathscr{L}_{\text{out}}(\mathcal{D}) = P_{\text{out}} J_{\text{out}}(\Lambda_{\text{out}}) P_{\text{out}}^{-1}$.

Proposition 1.5 ([6, 9]). *The matrix $\mathscr{L}_{\text{in}}(\mathcal{D})$ exhibits the following properties relating to $\mathcal{D}$:*

(a) if $\mathcal{D}$ contains a rooted out-branching then

(i) $\textbf{rank}\,\mathscr{L}_{\text{in}}(\mathcal{D}) = n - 1$ *with* $\mathscr{L}_{\text{in}}(\mathcal{D})\mathbf{1} = \mathbf{0}$ *and*

(ii) $J_{\text{in}}(\Lambda) = \begin{bmatrix} 0 & \cdots & & 0 \\ \vdots & J_{\text{in}}(\lambda_2) & & \vdots \\ & & \ddots & 0 \\ 0 & \cdots & 0 & J_{\text{in}}(\lambda_m) \end{bmatrix}$, *where* $Re(\lambda_i) > 0$, *and* $J_{\text{in}}(\lambda_i)$ *is the Jordan block associated with eigenvalues* λ_i, $i = 2, \ldots, m$ *and* $\lambda_1 = 0$.

(b) if $\mathcal{G}$ is balanced then

(i) $\mathbf{1}^T \mathscr{L}_{\text{in}}(\mathcal{D}) = \mathbf{0}^T$ *and*
(ii) $\mathscr{L}_{\text{in}}(\mathcal{D}) + \mathscr{L}_{\text{in}}(\mathcal{D})^T$ *is positive semidefinite.*

Corollary 1.6. *The matrix $\mathscr{L}_{\text{out}}(\mathcal{D})$ exhibits the following properties relating to $\mathcal{D}$:*

(a) if $\mathcal{D}$ contains a rooted in-branching then

(i) $\textbf{rank}\,\mathscr{L}_{\text{out}}(\mathcal{D}) = n - 1$ *with* $\mathbf{1}^T \mathscr{L}_{\text{out}}(\mathcal{D}) = \mathbf{0}^T$ *and*

(ii) $J_{\text{out}}(\Lambda) = \begin{bmatrix} 0 & \cdots & & 0 \\ \vdots & J_{\text{out}}(\lambda_2) & & \vdots \\ & & \ddots & 0 \\ 0 & \cdots & 0 & J_{\text{out}}(\lambda_m) \end{bmatrix}$, *where* $Re(\lambda_i) > 0$, *and* $J_{\text{out}}(\lambda_i)$ *is the Jordan block associated with eigenvalues* λ_i, $i = 2, \ldots, m$ *and* $\lambda_1 = 0$.

(b) if $\mathcal{D}$ is balanced then

(i) $\mathscr{L}_{\text{out}}(\mathcal{D})\mathbf{1} = \mathbf{0}$ *and*
(ii) $\mathscr{L}_{\text{out}}(\mathcal{D}) + \mathscr{L}_{\text{out}}(\mathcal{D})^T$ *is positive semidefinite.*

Proof. If $\mathcal{D}$ has a rooted out-branching, then $\mathcal{D}^r$ has a rooted in-branching. Applying Proposition 1.3, $\textbf{rank}\,\mathscr{L}_{\text{in}}(\mathcal{D}) = \textbf{rank}\,\mathscr{L}_{\text{in}}(\mathcal{D})^T = \textbf{rank}\,\mathscr{L}_{\text{out}}(\mathcal{D}^r)$ and

$\mathscr{L}_{\text{in}}(\mathcal{D})\mathbf{1} = \mathscr{L}_{\text{out}}(\mathcal{D}^r)^T\,\mathbf{1} = (\mathbf{1}^T\mathscr{L}_{\text{out}}(\mathcal{D}^r))^T$. Using Proposition 1.5(a.i), then (a.i) follows. Now, $\mathscr{L}_{\text{in}}(\mathcal{D})^T = (P_{\text{in}} J_{\text{in}}(\Lambda)\, P_{\text{in}}^{-1})^T = (P_{\text{in}}^{-1})^T J_{\text{in}}(\Lambda)^T\, P_{\text{in}}^T$. Applying Proposition 1.3, $\mathscr{L}_{\text{out}}(\mathcal{D}^r) = P_{\text{out}} J_{\text{out}}(\Lambda)\, P_{\text{out}}^{-1}$ has the properties $P_{\text{out}} = (P_{\text{in}}^{-1})^T$ and $J_{\text{in}}(\lambda_2) = J_{\text{out}}(\lambda_2)^T$ and so $Re(\lambda_i)$ of $J_{\text{in}}(\lambda_i)$ and $J_{\text{out}}(\lambda_i)$ for all $i = 1,\ldots,m$ are equal. Using Proposition 1.5(a.ii), (a.ii) follows. If $\mathcal{D}$ is balanced, from Proposition 1.2 $\mathscr{L}_{\text{out}}(\mathcal{D}) = \mathscr{L}_{\text{in}}(\mathcal{D})$, thus using Proposition 1.5(b.i) and (b.ii), (b.i) and (b.ii) follows. □

1.2.2 Dynamics and Agreement

We now examine the dynamics of (1.3). The equilibrium $\tilde{x}_i$ for all $i \in V$ satisfies

$$\tilde{\delta}_i \tilde{x}_i = \sum_{(j,i)\in E} w_{ij}\tilde{x}_j. \tag{1.6}$$

The flow interpretation of this equilibrium is that the divergence of flow at every node goes to zero.

One significant feature of this equilibrium can be found when we define $w_{ij} = \frac{1}{|\mathcal{N}_{\text{out}}(v_i)|}$ for each $i \in V$. Therefore, $\tilde{\delta}_i = 1$ and the equilibrium condition is $x_i = \sum_{(j,i)\in E} \frac{1}{|\mathcal{N}_{\text{out}}(v_j)|} x_j$ for all $i \in V$. This equilibrium is exactly the rank metric used in the PageRank algorithm underlying the Google search engine [10]. The premise of the ranking is that a node i should be high ranked if (a) the rank metric of nodes linking to node i are high and (b) should be low if node i has few incoming edges or if the nodes linking to node i have a low rank metric.

The dynamics of consensus and advection can be represented in closed form as $x(t) = e^{-\mathscr{L}_{\text{in}}(\mathcal{D})t}x(0)$ and $x(t) = e^{-\mathscr{L}_{\text{out}}(\mathcal{D})t}x(0)$, respectively. Many features of these dynamics are a direct consequence of the characteristics of the matrices $e^{-\mathscr{L}_{\text{in}}(\mathcal{D})t}$ and $e^{-\mathscr{L}_{\text{out}}(\mathcal{D})t}$. These features are later exploited in the following proposition to reason about invariance and convergence properties of these dynamics.

Proposition 1.7 ([6]). *For $\psi_{\text{in}}(\mathcal{G}) = e^{-\mathscr{L}_{\text{in}}(\mathcal{D})\epsilon}$, where $\epsilon > 0$,*

(i) $\psi_{\text{in}}(\mathcal{D}) \geq 0$, $\psi_{\text{in}}(\mathcal{D})$ is a right stochastic matrix, i.e., $[\psi_{\text{in}}(\mathcal{D})]_{ij} \geq 0$ and $\psi_{\text{in}}(\mathcal{D})\,\mathbf{1} = \mathbf{1}$, and
(ii) $[\psi_{\text{in}}(\mathcal{D})]_{ij} > 0 \iff i = j$ or there is a directed path from j to i in $\mathcal{D}$.

Corollary 1.8. *For $\psi_{\text{out}}(\mathcal{D}) = e^{-\mathscr{L}_{\text{out}}(\mathcal{D})\delta}$, where $\epsilon > 0$,*

(i) $\psi_{\text{out}}(\mathcal{D}) \geq 0$, $\psi_{\text{out}}(\mathcal{G})$ is a left stochastic matrix, i.e., $[\psi_{\text{out}}(\mathcal{G})]_{ij} \geq 0$ and $\mathbf{1}^T\psi_{\text{out}}(\mathcal{D}) = \mathbf{1}^T$, and
(ii) $[\psi_{\text{out}}(\mathcal{D})]_{ij} > 0 \iff i = j$ or there is a directed path from i to j in $\mathcal{D}$.

Proof. Applying Proposition 1.3, $\psi_{\text{in}}(\mathcal{D}^r) = e^{-\mathscr{L}_{\text{in}}(\mathcal{D}^r)\epsilon} = e^{-\mathscr{L}_{\text{out}}(\mathcal{D})^T\epsilon} = \left(e^{-\mathscr{L}_{\text{out}}(\mathcal{D})\epsilon}\right)^T = \psi^T_{\text{out}}(\mathcal{D})$. Using Proposition 1.7, the Corollary follows. □

The familiar property that a balanced digraph $\mathcal{D}$ exhibits a doubly stochastic ψ_{in} is a consequence of Proposition 1.7, the related Corollary 1.8, and for balanced $\mathcal{D}$, $\mathscr{L}_{\text{in}}(\mathcal{D}) = \mathscr{L}_{\text{out}}(\mathcal{D})$ (Proposition 1.2). A result of this property is that, for consensus dynamics over a balanced digraph, the state of the nodes, at any instant in time, is a convex combination of the values of all the nodes at a previous instant in time.

Proposition 1.9. *The advection dynamics are positively invariant over $x_i \geq 0$ for all $i \in V$, i.e., if $x_i(0) \geq 0$ for all $i \in V$ then $x_i(t) \geq 0$ for all $i \in N$ for all $t > 0$.*

Proof. From Corollary 1.8, $e^{-\mathscr{L}_{\text{out}}(\mathcal{D})t}$ is a nonnegative matrix so $x(t) = e^{-\mathscr{L}_{\text{out}}(\mathcal{D})t}x(0)$ is nonnegative for $x_i(0) \geq 0$ for all $i \in N$ and all $t > 0$. □

The following Proposition 1.10 for the consensus dynamics characterizes the equilibrium where $\mathcal{D}$ contains a rooted out-branching. Equivalently, Proposition 1.11 for the advection dynamics characterizes the equilibrium where $\mathcal{D}$ contains a rooted in-branching.

Proposition 1.10 ([6]). *For a digraph $\mathcal{D}$ containing a rooted out-branching, the consensus dynamics (1.5), initialized from $x(0) = x_0$, satisfies*

$$\lim_{t\to\infty} x(t) = \frac{1}{\sqrt{n}}\left(u^T x_0\right)\mathbf{1},$$

where $u = \bar{u}/\|\bar{u}\|$ and $\bar{u}^T\mathscr{L}_{in}(\mathcal{D}) = 0$. Further, the quantity $u^T x_0$ is conserved and $u_i > 0$ for all $i \in V$ if and only if $\mathcal{D}$ is strongly connected.

Proposition 1.11. *For a digraph $\mathcal{D}$ containing a rooted in-branching, the advection dynamics (1.3), initialized from $x(0) = x_0$, satisfies*

$$\lim_{t\to\infty} x(t) = \frac{1}{\sqrt{n}}\left(\mathbf{1}^T x_0\right)v,$$

where $v = \bar{v}/\|\bar{v}\|$ and $\mathscr{L}_{\text{out}}(\mathcal{D})\bar{v} = 0$. Further, $v_i > 0$ for all $i \in V$ if and only if $\mathcal{D}$ is strongly connected.

Proof. Noting that $x(t) = e^{-\mathscr{L}_{\text{out}}(\mathcal{D})t}x_0$ and $\lim_{t\to\infty} e^{-\mathscr{L}_{\text{out}}(\mathcal{D})t} = vw^T$, from the rank condition and Jordan decomposition in Corollary 1.6, where the first column of P_{out} and first row of P^{-1}_{out} are v and w^T respectively (i.e., the normalized right and left eigenvectors associated with the zero eigenvalue). Choosing $w = \frac{1}{\sqrt{n}}\mathbf{1}$ from Corollary 1.6 one has

$$\lim_{t\to\infty} x(t) = \left(\frac{1}{\sqrt{n}}v\mathbf{1}^T\right)x_0 = \frac{1}{\sqrt{n}}\left(\mathbf{1}^T x_0\right)v.$$

Since $\mathcal{D}$ is strongly connected, for every node pair i and j there exists a directed path in $\mathcal{D}$, and so for every node pair j and i there exists a reverse directed path. Consequently, $\mathcal{D}^r$ is also strongly connected. From Proposition 1.10, the left null space u of $\mathscr{L}_{\text{in}}(\mathcal{D}^r)$ has all positive elements, therefore, from Proposition 1.3, since $\mathscr{L}_{\text{out}}(\mathcal{D}) = \mathscr{L}_{\text{in}}(\mathcal{D}^r)^T$ the right null space v of $\mathscr{L}_{\text{out}}(\mathcal{D})$ has all positive elements. □

The following proposition relating to the equilibrium state of the consensus dynamics over a balanced, rooted in-branching $\mathcal{D}$ is a consequence of Propositions 1.10 and 1.11, the fact that the consensus and advection dynamics are identical (Proposition 1.2), and that if a digraph is balanced and has a rooted in-branching then it also has a rooted out-branching.

Proposition 1.12 ([6]). *The consensus dynamics (and hence advection dynamics) over a digraph $\mathcal{D}$ reaches $x(t) = \frac{1}{n}\mathbf{1}\mathbf{1}^T x_0$ for every initial condition if and only if $\mathcal{D}$ is balanced and contains a rooted in-branching (and hence a rooted out-branching).*

The results of this section are summarized in Fig. 1.1 which is structured to highlight the differences between the two sets of dynamics over rooted in/out-branching digraphs and the culmination of these features in balanced digraphs.

1.3 Examples

Next we showcase two applications of advection dynamics to networked, multi-agent systems. We focus on two examples that of formation control and a sensor coverage problem.

1.3.1 Example 1: Formation Control

The formation control problem where multi-agent teams form geometric patterns is a popular application for consensus [6]. With consensus, the formation is acquired by reaching a consensus on an origin in space and then each agent is coded with a reference position relative to this origin. We now proceed to perform the same task using advection where the geometry of the configuration is controlled by the weights of the edges.

We consider a swarm of n vehicles moving along the x-axis with velocity $\nu \in \mathbb{R}$. The objective is for the swarm to move in formation with a predefined shape. We assume each vehicle is aware of its global position and that each vehicle is able to measure the relative position to one other predesignated vehicle, thus acquiring its global position.

Let $\bar{x}(t) \in \mathbb{R}^n$ and $\bar{y}(t) \in \mathbb{R}^n$ be the position of vehicles at time t along the x and y axes, respectively. The origin of the coordinate system is selected such that $\mathbf{1}^T \bar{x}_i(0) \neq 0$ and $\mathbf{1}^T \bar{y}_i(0) \neq 0$. The advection-based dynamics of the vehicles are

$$\begin{aligned}\dot{x}(t) &= -\mathscr{L}_{\text{out}}(\mathcal{D}_x)x(t)\\ \dot{y}(t) &= -\mathscr{L}_{\text{out}}(\mathcal{D}_y)y(t),\end{aligned} \tag{1.7}$$

where $x(t) := \bar{x}(t) - vt\mathbf{1}$ and $y(t) := \bar{y}(t)$. The digraphs $\mathcal{D}_x = (V, E, W_x)$ and $\mathcal{D}_y = \left(V, E, W_y\right)$ are directed cycle digraphs[1] with positive weights $\left\{w^x_{i,j}\right\} \in W_x$ and $\left\{w^y_{i,j}\right\} \in W_y$, respectively. The edge set E corresponds to the inter-agent sensing where if vehicle i can measure the position of vehicle j then there is an edge $\{i, j\} \in E$. From (1.6), for the advection dynamics with underlying digraph $\mathcal{D}^x$ the equilibrium $(x, y) = (\tilde{x}, \tilde{y})$ is $\tilde{x}_i = \alpha_x/w^x_{i,i+1}$ for all $i = 1, \ldots, n-1$ and $\tilde{x}_n = \alpha_x/w^x_{n,1}$ where $\alpha_x = \mathbf{1}^T x_0/\sum_{\{j,i\}\in E}\left(1/w^x_{i,j}\right)$. As all weights are positive and $x_i(0) > 0$ then $\alpha_x > 0$. Similarly, for equilibrium $\tilde{y}$, constant $\alpha_y > 0$ and corresponding to digraph $\mathcal{D}^y$. Therefore, the weight selection completely dictates the *shape* of the equilibrium with the scaling of the shape is dictated by α_x and α_y. From Proposition 1.1, the sum of states is always constant and so the centroid of the formation is $\left(\frac{1}{n}\mathbf{1}^T x(0) + vt, \frac{1}{n}\mathbf{1}^T y(0)\right)$.

We apply this advection formation technique to 6 vehicles moving along the x axis with velocity $v = 2\,\text{m/s}$ and a required constellation with shape defined in an arbitrary reference frame as $x_s = (1, 2, 1, 3, 2, 1)^T$ and $y_s = (1, 2, 0, 0, -2, -1)^T/\sqrt{2}$; see Fig. 1.2 for shape. Let

$$\left[\, w^x_{12}\; w^x_{23}\; w^x_{34}\; w^x_{45}\; w^x_{56}\; w^x_{61} \,\right] = \left[\, 1\; \tfrac{1}{2}\; 1\; \tfrac{1}{3}\; \tfrac{1}{2}\; 1 \,\right]$$

and

$$\left[\, w^y_{12}\; w^y_{23}\; w^y_{34}\; w^y_{45}\; w^y_{56}\; w^y_{61} \,\right] = \sqrt{2}\left[\, 1\; 1/2\; 0\; 0\; -1/2\; -1 \,\right] + 2,$$

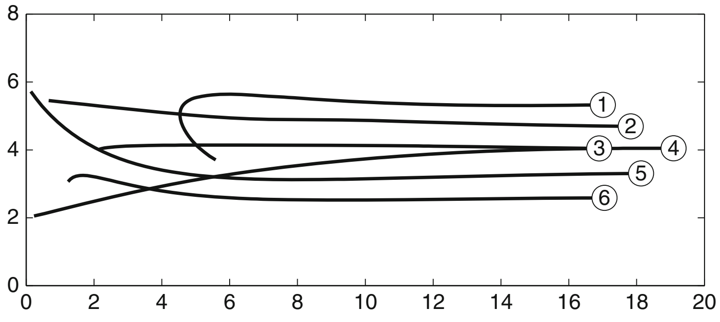


Fig. 1.2 Advection formation control with initial random condition and position after 8 s marked with circles

[1] A directed cycle digraph $\mathcal{D} = (V, E, W)$ is an n node digraph with $\{i+1, i\} \in E$ and edge weight $w_{i,i+1} \in W$ for $i = 1, \ldots, n-1$ as well as $\{1, n\} \in E$ with edge weight $w_{n,1} \in W$.

where this weight selection was chosen for ease of demonstration and is not the only selection that will satisfy a certain formation geometry. The vehicle trajectory initialized randomly in the first quadrant is displayed in Fig. 1.2 with the desired final formation achieved.

1.3.2 Example 2: Sensor Network

We consider a sensor surveillance task, operating in a land corridor of length d_w, where the sensors are directional with a narrow cone of observation. A set of n sensors are randomly placed pointing west in the corridor, which is oriented east to west. The objective is to acquire good coverage of the corridor while satisfying a total sensor power constraint by trading power between neighboring sensors. Let $x_i(t) \geq 0$ be the fraction of total network power and $z_i(t) \geq 0$ be the coverage of sensor i at time t. We utilize a coverage model of the form $z_i(t) = \beta x_i(t)$ (single-ray range sensor or wedge-shaped sensor with fixed arc-length) for some $\beta > 0$[2]. We subsequently require that the total power of the sensor network is constant, i.e., $\sum x_i(t) = 1$ for all t, while maintaining good coverage. We will now discuss our measure of "good coverage," which aims to minimize gaps between sensor observation areas along the east–west axis.

Assuming the minimal power is available to guarantee no gaps between sensors areas, the corresponding optimal power for each sensor i would be $x_i^* = \frac{\zeta_i}{d_w}$, where ζ_i is the distance to the closest sensor to the east of sensor i. Consider a local area around a point $\boldsymbol{p} \in \mathbb{R}^2$ in the corridor, with a set u^+ of δ_i^+ sensors west of $\boldsymbol{p}$, and a set u^- of δ_i^- sensors east of $\boldsymbol{p}$. Our local coverage cost function is $c_i(\boldsymbol{p}) = \left|\sum_{k\in u+} z_k - \sum_{k\in u-} z_k\right|$. The cost function penalizes nonuniform coverage east and west of point $\boldsymbol{p}$. For an infinite corridor, $x^* = \operatorname{argmin}_x \int_0^{d_w} c_i(\rho, 0)\, d\rho$. If the coverage for all $k \in u^-$ is approximately equal to z^-, $c_i = \left|\sum_{k\in u+} z_k - \delta_i^- z^-\right|$. Ideally then,

$$z^- = \frac{1}{\delta_i^-} \sum_{k\in u+} z_k, \tag{1.8}$$

and so if we were to place a sensor i at location $\boldsymbol{p}$ then we assume for "good coverage" $z_i \approx z^-$. An interpretation of this equilibrium is that coverage of an area west of the sensor i which is $\sum_{k\in u+} \tilde{z}_k$ should be maintained in an equal area east of sensor i where there are already located δ_i^- sensors, so sensor i is responsible for approximately $\frac{1}{\delta_i^-}$ of the area east of $\boldsymbol{p}$, i.e., $\frac{1}{\delta_i^-} \sum_{k\in u+} \tilde{z}_k$.

Let the position of sensor i be $(p_i, q_i) \in \mathbb{R}^2$ (where the x axis points east and the y axis north). We assume, since the sensors are in a land corridor, that

[2] This work has also been extended to $z_i(t) = \beta x_i(t)^p$ for $p > 0$.

$\max_{i\in N}(q_i) - \min_{i\in N}(q_i)$ is small, and subsequently, if the communication range on sensors is $d_c > 0$, $d_c \in \mathbb{R}$, that sensor i is in communication range of sensor j if $\left|p_i - p_j\right| \leq d_c$. Additionally we assume that if sensors are sufficiently close to the east or west end of the corridor, communications can be relayed between them via some infrastructure at the ends of the corridor. A communication digraph $\mathcal{D} = (V, E, W)$ is defined such that if sensor j is within communication range of sensor i and $p_j > p_i$ then $(i, j) \in E$ with the exception of the sensors at the ends of the corridor where if $p_j - p_i \geq d_w - d_c$ then $(j, i) \in E$. The digraph is unweighted with $w_{ij} = 1$ for all $(j, i) \in E$.

Assuming d_c is large enough to form a rooted in-branching, and $\tilde{\delta}_i > 0$ for all $i \in V$, then $\mathcal{D}$ will be strongly connected. Applying the advection dynamics (1.3), our equilibrium power $\tilde{x}$ will satisfy

$$\tilde{x}_i = \frac{1}{\tilde{\delta}_i} \sum_{(k,i)\in E} \tilde{x}_k,$$

and $\tilde{x}_i > 0$ for all $i \in V$ if $\sum x_i(0) = \mathbf{1}^T x(0) > 0$ (Proposition 1.11). The corresponding equilibrium coverage $\tilde{z}$ is $\tilde{z}_i = \frac{1}{\tilde{\delta}_i} \sum_{(k,i)\in E} \tilde{z}_k$ for all $i \in V$. As previously discussed, this is our condition (1.8) for "good coverage" whereby the coverage of the area d_c west of the sensor maintained d_c east of sensor i. An interesting aspect of this formulation is that the unweighted network topology is being exploited to infer inter-sensor distance information and hence coverage density characteristics.

We assume that all sensors are initialized with a feasible power, i.e., $\sum_{i=1}^{n} x_i(0) = 1$ and $x_i(0) \geq 0$ for all $i \in V$. From Proposition 1.1, the total power of the sensor network will be conserved, i.e., $\sum_{i=1}^{n} x_i(t) = 1$ for all time t. As advection is positively invariant over the nonnegative $x_i(t)$, from Proposition 1.9, the power will always be nonnegative, i.e., $x_i(t) \geq 0$, for all $i \in V$.

We apply this approach to a $d_w = 40\,\text{m}$ long land corridor containing 40 randomly placed sensors. The initial power fraction was assigned randomly and $d_c = 1.75\,\text{m}$ dictates the topology of the flow digraph. The final equilibrium power $\tilde{x}(t)$ overlayed on the digraph $\mathcal{D}$ is displayed in Fig. 1.3. Figure 1.4 depicts the observation cones for (a) the optimal power usage from all sensors, (b) the equilibrium power usage obtained using an arc-length fixed to that of the corridor's width, and (c) a uniform power usage for all sensors. We find that the minimum power requirement by the advection power distribution is within 1.25 times of the optimal power. This is compared to a uniform sensor power which required 2.5 times the optimal power.

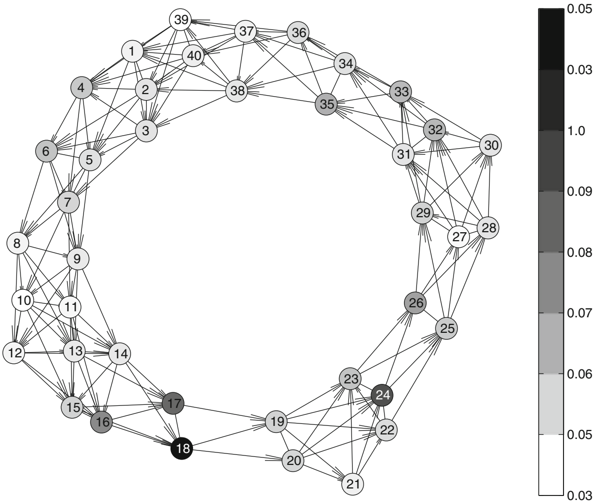


Fig. 1.3 Sensor digraph with gray scale gradations corresponding to equilibrium power $x(t)$. Nodes are numbered from west to east

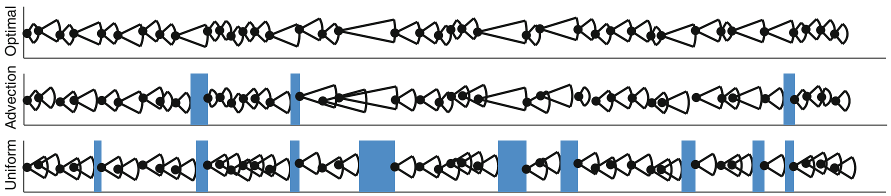


Fig. 1.4 Optimal, advection dynamics and uniform sensor coverage of the land corridor. The shaded bands indicate areas not covered by the sensors

1.4 Remarks

This chapter presents an advection-based approach to multi-agent cooperative control. We compare the advection dynamics to the popular consensus dynamics but also make comments on novel properties that are only held by advection. One property of particular significance is the conservation of the sum of the states. We demonstrate the utility of this property in such applications as sensor coverage where power can be traded through the network to optimize coverage. Because of the parallels with consensus dynamics, there is a large area of future advection research involving the application of advection to problems traditionally solved by consensus. One application of particular interest is the introduction of control nodes, which do not conform to advection, into an advection-based network.

References

1. R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.
2. H. G. Tanner, G. J. Pappas, and V. Kumar, "Leader-to-formation stability," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 443–455, 2004.
3. A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.
4. Y. Hatano and M. Mesbahi, "Agreement over random networks," *IEEE Transactions on Automatic Control*, vol. 50, no. 11, pp. 1867–1872, 2005.
5. L. Grady and J. Polimeni, *Discrete Calculus*. New York: Springer, 2010.
6. M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton: Princeton University Press, 2010.
7. R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York: Cambridge University Press, 1990.
8. A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*. Academic Press, 1979.
9. F. Bullo, J. Cortes, and S. Martinez, *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*. Princeton: Princeton University Press, 2009.
10. S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISBN Systems*, vol. 30, pp. 107–117, 1998.

Chapter 2
Beyond Linear Protocols

Abstract This chapter examines the set of equilibria and asymptotic stability of a large class of dynamical networks with non-identical nonlinear node dynamics. The interconnection dynamics are defined by M-matrices. An example of such a class of systems include nonlinear consensus protocols as well as other distributed protocols of interest in cooperative control and distributed decision-making. We discuss the model's relationship to the network topology, investigate the properties of its equilibria, and provide conditions for convergence to the set of equilibria. We also provide examples of the versatility of this model in the context of a sensor coverage problem. The model is extended to incorporate additional nonlinearities; an application for this latter model is also provided in the realm of neural networks.

2.1 Introduction

Stability is an important requirement for networked systems, as it is for dynamic systems in general. Examining stability for nonlinear non-networked systems is generally a challenging task with Lyapunov stability theory used as a common tool. A further challenge is that the connection of stable, isolated dynamics does not guarantee network stability with node interactions able to introduce instabilities. The consequence of these challenges is that proving stability of networked systems often involves searching for Lyapunov candidate by trial-and-error. In this chapter, we provide a class of nonlinear networks with interconnection dynamics defined by M-matrices that exhibit asymptotic stability.

M-matrices are used to model synchronization in networks [1], population migration [2], Markov processes, and supply and demand in economic systems [3]. M-matrices also appear in the discretization of differential operators. An example is the discretization of diffusion [1] and advection [4] dynamics which generate the in-degree and out-degree Laplacian matrix, respectively, and both of which are M-matrices. The in-degree Laplacian matrix forms the basis of consensus models which are effective for distributed information-sharing and control of networked, multi-agent systems in settings such as multi-vehicle control, formation control, swarming, and distributed estimation; see, for example, [5, 6].

The novelty of our work in the field of nonlinear networks lies in its generality. Nonlinear dynamics over networks has been a largely unexplored research

A. Chapman, *Semi-Autonomous Networks*, Springer Theses,
DOI 10.1007/978-3-319-15010-9_2

area; [7] provides a good summary of the field. Many nonlinear network results are quite conservative such as assuming uniform node dynamics, undirected network structure, and restrictive classes of nonlinear dynamics. Our results are over non-identical nodes, strongly connected directed network topologies, and have mild assumptions on the nonlinear dynamics. We believe that these generalizations are of special interest to the area of nonlinear consensus models. Not surprisingly, there is a great focus in nonlinear network research on globally asymptotically stable equilibrium. However, in many applications, asymptotic convergence to an equilibria is of interest, such as Laplacian dynamics [5]. It is with this in mind that we examine this more general type of convergence.

Araki and Kondo [8] decomposed networks into subnetworks obtaining Lyapunov functions for these subsystems, and under DC gain conditions, showed asymptotic stability of the nonlinear system. In [7], the authors provide sufficient conditions for asymptotic convergence over a large class of nonlinear systems but the application of the chapter's results does require the construction of Lyapunov functions. Xiang and Chen [9] provide a passivity measure indicating the amount of effort required to stabilize a node. The measure is used to form a network stability condition that guarantees asymptotic stability.

Recently, there has been interest in nonlinear consensus, a subclass of nonlinear dynamics over networks. Cortes [10] proposed nonlinear distributed algorithms over smooth functions that achieve consensus, with many results assuming balanced digraphs. In [11], the authors examined convergence to equilibria, referred to as semistability, for nonlinear consensus over balanced digraphs. Convergence of nonlinear consensus algorithms to a single point was examined in [12] using a contracting property. Yu et al. [13] examined linear consensus with an additional nonlinear term solely dependent on the agent's state and identical over all agents. We examine a similar model, in the extension component of this chapter, but with nonlinear consensus and the addition of non-identical nonlinear terms.

The organization of the chapter is as follows. We begin by defining the class of nonlinear network models and related background. We provide digraph related features of the network dynamics. The left and right null-spaces of M-matrices are used to characterize the equilibria of the model as well as an invariance property of the system. An asymptotic convergence result is then presented and justified by exploiting properties of M-matrices. Two examples of the network model are investigated: a social networking example based on opinion dynamics inspired by a similar example in [1] with nonlinear non-differentiable dynamics, followed by a sensor network problem based on the model in [4]. Finally, the model is extended to incorporate individual node dynamics independent of the network structure. An illuminating neural network example [8] is provided which demonstrates the utility of this extended model.

2.2 Model

We consider a multi-agent network of n coupled nodes with each nodes' state $x_i(t) \in \mathbb{R}$ at time t. The system is described by the differential equations

$$\dot{x}_i(t) = -a_{ii} f_i(x_i(t)) + \sum_{j \neq i} a_{ij} f_j\left(x_j(t)\right), \; i = 1, \ldots, n,$$

where $f_i : \mathbb{R} \to \mathbb{R}$ is a scalar-valued function representing the ith node dynamics, and $a_{ij} \in \mathbb{R}$ is nonnegative for all $i, j \in \{1, \ldots, n\}$. In a more compact form, the dynamics can be written as

$$\dot{x} = -Af(x), \tag{2.1}$$

where $x = [x_1, \ldots, x_n]^T \in \mathbb{R}^n$, $f(x) = [f_1(x_1), \ldots, f_n(x_n)]^T \in \mathbb{R}^n$ and the matrix A is defined as

$$A = \begin{bmatrix} a_{11} & -a_{12} & \cdots & -a_{1n} \\ -a_{21} & a_{22} & -a_{23} & \vdots \\ \vdots & & \ddots & -a_{n-1,n} \\ -a_{n1} & \cdots & -a_{n,n-1} & a_{nn} \end{bmatrix}.$$

In this way, the A matrix codifies the nodes' interconnections. We define the class of matrices of this form as $Z_n = \left\{A = \left(a_{ij}\right) \in \mathbb{R}^{n \times n} : a_{ii} \geq 0, a_{ij} \leq 0, i \neq j\right\}$. We make the assumption that $f(x)$ is continuous and $\int_0^{x_i} f_i(y)\,dy$ is radially unbounded, i.e., ($\int_0^{x_i} f_i(y)\,dy \to \infty$ as $|x_i| \to \infty$), for $i = 1, \ldots, n$. The class of functions that satisfies this requirement we denote by F_0. Further, we assume that A is a member of the subclass of matrices known as irreducible M-matrices denoted by M_0. These terms are defined as follows:

Definition 2.1 ([3]). A matrix $A \in Z_n$ is called a nonsingular (singular) M-matrix if the real part of every eigenvalue of A is positive (nonnegative).

Definition 2.2 ([14]). A matrix $A \in \mathbb{R}^{n \times n}$ is reducible if and only if, for some permutation matrix P, the matrix $P^T AP$ is block upper triangular. A matrix that is not reducible is irreducible.

M-matrices appear in a myriad of problems including biological, physical, and social sciences; for example, see [15, 16]. One of the famous matrices is the in-degree Laplacian matrix used in consensus (or synchronization) dynamics. An attraction of the Laplacian-based dynamics is the structure of the networks can be coded into the in-degree Laplacian matrix, which will be defined presently, and consequently graph-theoretic results can be applied to the dynamic models. As for the in-degree Laplacian matrix, the underlying coupling network between agents in the network is encoded in the M-matrix A.

The network topology of model (2.1) can be formed using a digraph realization of A. The corresponding digraph realization of matrix A is a digraph $\mathcal{D}$ where $\{i, j\} \in E$ if $a_{ji} > 0$ and $i \neq j$ with corresponding weight $a_{ji} = w_{ji} \in W$. Because of the link between the off-diagonals of matrix A and its digraph realization $\mathcal{D}$, we will often denote the matrix A as $A(\mathcal{D})$ to emphasize the underlying digraph.

As the diagonal of the M-matrix is ignored in the digraph realization there is more than one M-matrix for any given digraph. Consequently, when provided with the digraph $\mathcal{G}$ and the diagonal of $A(\mathcal{D})$ the M-matrix $A(\mathcal{D})$ is fully defined. The in-degree Laplacian matrix $\mathscr{L}_{\text{in}}(\mathcal{D})$ and out-degree Laplacian matrix $\mathscr{L}_{\text{out}}(\mathcal{D})$ have the same digraph realization $\mathcal{D}$ but differ in their diagonals defined with $[\mathscr{L}_{\text{in}}(\mathcal{D})]_{ii} = \sum_{j \neq i} a_{ij}$ and $[\mathscr{L}_{\text{out}}(\mathcal{D})]_{ii} = \sum_{j \neq i} a_{ji}$ for all $i = 1, \ldots, n$. These M-matrices exhibit the special conditions $\mathscr{L}_{\text{in}}(\mathcal{D})\,\mathbf{1} = \mathbf{0}$ and $\mathbf{1}^T \mathscr{L}_{\text{out}}(\mathcal{D}) = \mathbf{0}^T$.

The related digraph $\mathcal{D}$ of an M-matrix $A(\mathcal{D})$ can be used to extract matrix properties of $A(\mathcal{D})$, even though there is not a one-to-one correspondence between $A(\mathcal{D})$ and $\mathcal{D}$. If the digraph $\mathcal{D}$ is undirected, then $A(\mathcal{D})$ is symmetric. The irreducibility of a matrix A can be established using the digraph realization of A. This result is summarized in the following proposition.

Proposition 2.3 ([16]). *A matrix $A(\mathcal{D})$ is irreducible if and only if $\mathcal{D}$, the digraph realization of A, is strongly connected. A digraph is strongly connected if between every pair of distinct nodes there exists a directed path.*

We give a brief note that the property $A + A^T \succcurlyeq 0$, which is useful for proving asymptotic convergence using Lyapunov theory, is in general not exhibited by the family of irreducible M-matrices. One of the attractions of irreducible M-matrices is that they fit into a special class of diagonally semi-stable matrices that will be exploited in the core result of this chapter (Lemma 2.11). This property is formally stated in the following.

Proposition 2.4 ([3]). *If A is a irreducible M-matrix, then there exists a positive diagonal matrix $D \in \mathbb{R}^{n \times n}$ such that*

$$DA + A^T D \succcurlyeq 0.$$

Further, if A is nonsingular $DA + A^T D \succ 0$.

We will not include a repetition of the proof by [3] in this chapter but we will re-derive this proposition for the special case where the M-matrix is the in-degree Laplacian matrix. We will deviate from the traditional derivation and use a graph-based proof to illustrate the utility of the digraph realization of M-matrices.

Proposition 2.5. *For a strongly connected digraph $\mathcal{D}$ and $v^T \mathscr{L}_{in}(\mathcal{D}) = 0$, $D\mathscr{L}_{in}(\mathcal{D}) = \mathscr{L}_{in}(\tilde{\mathcal{D}})$ for some balanced digraph $\tilde{\mathcal{D}}$ where D is a positive diagonal matrix with $[D]_{ii} = v_i$ for $i = 1, \ldots, n$ and $D\mathscr{L}_{in}(\mathcal{D}) + \mathscr{L}_{in}(\mathcal{D})^T D \succeq 0$.*

Proof. It has previously been shown in [17] that the eigenvector $v = [v_1, v_2, \ldots, v_n]^T$ can be found by examining the digraph $\mathcal{D} = (V, E, W)$ specifically

$$v_i = \sum_{T \in \mathcal{T}_i} \prod_{e_{kj} \in T} w_{jk},\ i = 1, \ldots, n$$

where $\mathcal{T}_i$ is the family of rooted directed spanning trees over $\mathcal{D}$ with node i as the root, and e_{ij} is an edge in a given tree $\mathcal{T}$ with weight w_{ji}. As $\mathcal{D}$ is strongly connected, $\mathcal{T}_i$ is nonempty and so $v_i > 0$, for all i.

Consider a new digraph formed from the old with $\tilde{\mathcal{D}} = \left(V, E, \tilde{W}\right)$ where $\tilde{W} = \left\{\tilde{w}_{jk}\right\}$ and $\tilde{w}_{jk} = v_j w_{jk}$ then $\mathscr{L}_{\text{in}}\left(\tilde{\mathcal{D}}\right) = D\mathscr{L}_{\text{in}}\left(\mathcal{D}\right)$. Further, for the left eigenvector $\tilde{v}$ corresponding to the zero eigenvalue has

$$\begin{aligned}
\tilde{v}_i &= \sum_{T \in \mathcal{T}_i} \prod_{e_{kj} \in T} \tilde{w}_{jk} \\
&= \sum_{T \in \mathcal{T}_i} \prod_{e_{kj} \in T} v_j w_{jk} \\
&= \left(\sum_{T \in \mathcal{T}_i} \prod_{e_{kj} \in T} w_{jk} \right) \prod_{j \neq i} v_j \\
&= v_i \prod_{j \neq i} v_j \\
&= \prod_{i=1}^{n} v_j .
\end{aligned}$$

Here, $\tilde{v}_i = \tilde{v}_j$ for all $i, j \in \{1, \ldots, n\}$, thus $\tilde{v} = \left(\prod_{i=1}^{n} v_j\right) \mathbf{1}$, or normalized $\tilde{v} = \frac{1}{n}\mathbf{1}$. Therefore, as the left eigenvector corresponding to the zero eigenvalue is $\mathbf{1}$ and similarly the right eigenvector is $\mathbf{1}$ (as $\mathscr{L}_{\text{in}}\left(\mathcal{D}\right)$ is an in-degree Laplacian), then $\tilde{\mathcal{D}}$ is a balanced digraph [1]. A property of balanced digraphs is that $\mathscr{L}_{\text{in}}(\tilde{\mathcal{D}}) + \mathscr{L}_{\text{in}}(\tilde{\mathcal{D}})^T \succeq 0$, hence $D\mathscr{L}_{\text{in}}(\mathcal{D}) + \mathscr{L}_{\text{in}}(\mathcal{D})^T D \succeq 0$. □

A corollary of Proposition 2.5 which is implicit within the Proposition's proof is as follows:

Corollary 2.6. *The edges of a strongly connected digraph can always be re-weighted to achieve a balanced digraph.*

The right and left null-spaces of an irreducible M-matrix play an important role in the dynamics of model (2.1) in particular in relation to the model's equilibria. The following proposition will be exploited shortly:

Proposition 2.7 ([3]). *For $A \in M_0$, A is invertible or, the rank of A is $n-1$, in which case every element in the right and left eigenvector corresponding to the zero eigenvalue are nonzero, and share the same sign.*

We will now investigate the equilibria and an invariant property of model (2.1), present the main convergence proof of the chapter, and provide examples.

2.3 Equilibria and Convergence

Model (2.1), under the assumptions $f(\cdot) \in F_0$ and $A \in M_0$, has in general many equilibria which we define by the set

$$\mathcal{A} = \{x \in \mathbb{R}^n | Af(x) = 0\}.$$

Consequently, the set $\mathcal{A}$ is completely defined by the properties of A and $f(\cdot)$. If $\mathcal{A}$ is composed of isolated equilibria and $x(t) \to \mathcal{A}$, then $x(t)$ will converge to some $x_e \in \mathcal{A}$.

From Proposition 2.7, A is nonsingular, or singular with rank $n-1$, we will proceed to investigate $\mathcal{A}$ for each of these cases.

For the case where A is nonsingular then this set becomes $\mathcal{A} = \{x \in \mathbb{R}^n | f(x) = 0\}$. As $f(x)$ is continuous, this implies that an equilibrium $x_e \in \mathcal{A}$ is isolated if and only if, for some ball around $[x_e]_i$, $f_i(x_i) = 0$ implies that $x_i = [x_e]_i$. If $f(x)$ is differentiable at x_e, then if $\frac{d}{dx_i} f_i([x_e]_i) \neq 0$ for $i = 1, \ldots, n$, there exists the aforementioned ball around x_i for $i = 1, \ldots, n,$. Therefore the equilibrium is isolated.

When A is singular there exists a right and left eigenvector corresponding to zero, v and w, respectively, such that $w^T A = \mathbf{0}^T$ and $Av = \mathbf{0}$. Using this fact, we can define a conserved quantity for the model as follows:

Proposition 2.8. *Let w be the left eigenvector of singular $A \in M_0$ corresponding to the zero eigenvalue. Then the quantity $w^T x(t)$ remain invariant under model (2.1).*

Proof. Since $w^T A = 0$, one has

$$\frac{d}{dt}\left\{w^T x(t)\right\} = -w^T A f(x) = 0.$$

□

For special matrices like the out-degree Laplacian matrix $\mathscr{L}_{\text{out}}(\mathcal{D})$, this is a familiar property with $\mathbf{1}^T \mathscr{L}_{\text{out}}(\mathcal{D}) = 0$, the conserved quantity is $\sum_{i=1}^n x_i(t)$.

The eigenvector v provides an alternate definition of the set of equilibria, specifically

$$\mathcal{A} = \{x \in \mathbb{R}^n | f(x) = \beta v, \beta \in \mathbb{R}\}. \tag{2.2}$$

Proposition 2.9. *The equilibrium $x_e \in \mathcal{A}$, corresponding to a singular A, is isolated if $f(x)$ is differentiable at x_e, $\frac{d}{dx_i} f_i([x_e]_i) \neq 0$ for all $i = 1, \ldots, n$, and*

$$\sum_{i=1}^{n} \frac{v_i w_i}{\frac{d}{dx_i} f_i([x_e]_i)} \neq 0,$$

where v and w are the right and left eigenvectors of A corresponding to the zero eigenvalue.

Proof. Consider the function

$$g(x, \beta) = [g_1(x_1, \beta), \ldots, g_n(x_n, \beta), g_{n+1}(x)]^T,$$

$$g_i(x_i, \beta) = f_i(x_i) - v_i \beta,\ i = 1, \ldots, n$$

and

$$g_{n+1}(x) = w^T(x - x_0),$$

where $\beta \in \mathbb{R}$, and x_0 is the initial condition. If x_e is an equilibrium, then for some β_e, $g(x_e, \beta_e) = 0$.

As $f(x)$ is differentiable at x_e then $g(x_e, \beta_e)$ is differentiable at (x_e, β_e). Calculating the Jacobian of $g(\cdot)$ about (x_e, β_e),

$$\nabla g(x, \beta) = \begin{bmatrix} \frac{d}{dx_1} g_1(x_1, \beta) & 0 & 0 & \frac{d}{d\beta} g_1(x_1, \beta) \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & \frac{d}{dx_n} g_n(x_n, \beta) & \frac{d}{d\beta} g_n(x_n, \beta) \\ \frac{d}{dx_1} g_{n+1}(x) & \cdots & \frac{d}{dx_n} g_{n+1}(x) & 0 \end{bmatrix}$$
$$= \begin{bmatrix} \frac{d}{dx_1} f_1(x_1) & 0 & 0 & v_1 \\ 0 & \ddots & 0 & \vdots \\ 0 & 0 & \frac{d}{dx_n} f_n(x_n) & v_n \\ w_1 & \cdots & w_n & 0 \end{bmatrix}.$$

The determinant of the Jacobian is

$$\det(\nabla g(x, \beta)) = -\sum_{i=1}^{n} v_i w_i \left(\prod_{j \neq i} \frac{d}{dx_j} f_j(x_j) \right).$$

From the condition $\frac{d}{dx_i} f_i\left([x_e]_i\right) \neq 0$ for all $i = 1, \ldots, n$,

$$\det\left(\nabla g\left(x_e, \beta_e\right)\right) = -\left(\prod_{i=1}^{n} \frac{d}{dx_i} f_i\left([x_e]_i\right)\right) \sum_{i=1}^{n} \frac{v_i w_i}{\frac{d}{dx_i} f_i\left([x_e]_i\right)}.$$

Consequently, if $\sum_{i=1}^{n} v_i w_i / \frac{d}{dx_i} f_i([x_e]_i) \neq 0$, then $\det\left(\nabla g\left(x_e, \beta_e\right)\right) \neq 0$. By the Inverse Mapping Theorem [18], when $\det\left(\nabla g\left(x_e, \beta_e\right)\right) \neq 0$ then there exists a ball around (x_e, β_e) such that $g\left(x, \beta\right)$ is one-to-one. In other words, there exists a region about (x_e, β_e) such that there are no other points satisfying $g\left(x, \beta\right) = 0$, i.e., no other equilibrium points.

Hence, x_e is an isolated equilibrium. □

From Proposition 2.9 and the fact that $v_i w_i$ is nonzero and has the same sign for all $i = 1, \ldots, n$ (Proposition 2.7), we have the following:

Corollary 2.10. *The equilibrium $x_e \in \mathcal{A}$, corresponding to a singular A, is isolated if $f\left(x\right)$ is differentiable at x_e and $\frac{d}{dx_i} f_i\left([x_e]_i\right) > 0$ for all $i = 1, \ldots, n$.*

We now provide the main result of the chapter which describes the asymptotic convergence of the model.

Lemma 2.11. *Under the assumption that $f_i(x) \in F_0$ and $A \in M_0$, model (2.1) has $x(t) \to \mathcal{A}$ for all initial conditions. If $\mathcal{A}$ is composed of isolated equilibria, then $x(t) \to x_e$ for some $x_e \in \mathcal{A}$.*

Proof. As $A \in M_0$ then from Proposition 2.4 there exists a positive diagonal matrix D with $DA + A^T D \succeq 0$. Consider the Lyapunov function $V(x) : \mathbb{R}^n \to \mathbb{R}$ defined by

$$V(x) = \sum_{i=1}^{n} [D]_{ii} \int_0^{x_i} f_i\left(y\right) dy,$$

which is continuous, differentiable, and radially unbounded.

Taking the derivative of $V(x)$,

$$\begin{aligned}
\dot{V}(x) &= \sum_{i=1}^{n} [D]_{ii}\, f_i(x_i)\dot{x}_i \\
&= f(x)^T D\dot{x} \\
&= -f(x)^T DAf(x) \\
&= -\frac{1}{2} f(x)^T DAf(x) - \frac{1}{2} f(x)^T A^T Df(x) \\
&= -\frac{1}{2} f(x)^T \left(DA + A^T D\right) f(x).
\end{aligned}$$

Now, from the property $DA + A^T D \succeq 0$, $\dot{V}(x) \leq 0$ where $\dot{V}(x) = 0$ for the set $\mathcal{A}$. Since $V(x)$ is radially unbounded and $\dot{V}(x) \leq 0$ for all x, the set $\Omega = (x \in \mathbb{R}^n | V(x) \leq c)$ is a compact, positively invariant set. From LaSalle's theorem [19], we conclude that every trajectory starting in Ω asymptotically converges to the set $\mathcal{A}$ as $t \to \infty$. Moreover since $V(x)$ is radially unbounded, the conclusion is global, as for every $x(0)$ there exists a c such that $x(0) \in \Omega$. Further, if all equilibria are isolated, then the trajectory must approach one of these equilibria. □

Next we showcase two examples of networked, multi-agent systems which fit under the umbrella of model (2.1). We focus on two examples: an opinion consensus problem and a sensor coverage problem.

2.3.1 Example 1: Opinion Consensus Dynamics

Consider a dynamical social model over a network, similar to the model in [1]. The ith member's private belief of an issue at time t is represented by $x_i(t) \in \mathbb{R}$ with $x_i(t) \geq 0$ ($x_i(t) < 0$) corresponding to a positive (negative) opinion on the issue. The conviction of the members' opinion is represented by the deviation of his/her social belief from zero, i.e.,$|x_i(t)|$. The member will cast a public binary vote $f_i(x_i) \in \{-1, 1\}$ on a ballot based on how supportive the ballot is of their private belief and a bias term represented through the finite variable $\epsilon_i \in \mathbb{R}$. We define this function as

$$f_i(x_i) = \begin{cases} 1 & \text{if } x_i(t) \geq -\epsilon_i \\ -1 & \text{otherwise.} \end{cases}$$

Member beliefs are influenced by observing the votes of other members. The n members of the group are represented by nodes in an interaction digraph $\mathcal{D} = (V, E, W)$ where an edge $\{j, i\} \in E$ with weighting $w_{ij} \in W$ corresponds to the impact of member j's vote on member i. The interaction dynamics are defined by

$$\dot{x}_i = \sum_{\{j,i\} \in E} w_{ij} \left(f_j(x_j) - f_i(x_i) \right), \; i = 1, \ldots, n,$$

or in matrix form,

$$\dot{x} = -\mathscr{L}_{\text{in}}(\mathcal{D}) f(x) \tag{2.3}$$

where $f(x) = [f_1(x_1), \ldots, f_n(x_n)]^T$. As $f(x)$ is not continuous Lemma 2.11 can not directly be used. We consequently consider an approximation of $f_i(x_i)$ defined as

$$h_i(x_i) = \begin{cases} 1 & \text{if } x_i \geq -\epsilon_i \\ x_i/\gamma + \epsilon_i/\gamma + 1 & \text{if } -\epsilon_i - \gamma/2 < x_i < -\epsilon_i \\ -1 & \text{otherwise.} \end{cases}$$

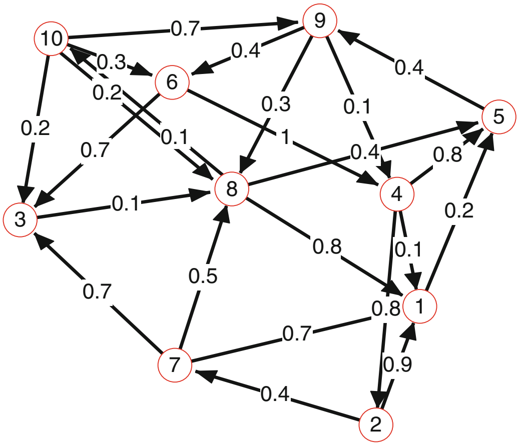


Fig. 2.1 Member interaction digraph with an *arrow* from i to j representing an edge (i, j) and corresponding edge weights w_{ji} marked on each edge

We observe that as $\gamma \to 0$ then $h_i(x_i) \to f_i(x_i)$. Further, $h(x) = [h_1(x_1), \ldots, h_n(x_n)]^T$ is continuous and

$$\int_0^{x_i} h_i(y)\,dy = \begin{cases} x_i & \text{if } x_i \geq -\epsilon_i \\ x_i^2/2\gamma + \epsilon_i x_i/\gamma + x_i & \text{if } -\epsilon_i - \gamma/2 < x_i < -\epsilon_i \\ -x_i & \text{otherwise,} \end{cases}$$

so $h(\cdot) \in F_0$ as $\int_0^{x_i} h_i(y)\,dy$ is radially unbounded. As $\gamma \to 0$, the equilibria set is $\mathcal{A} = \{x \in \mathbb{R}^n | f(x) = \{-\mathbf{1}, \mathbf{1}\}\} = \{x \geq -\epsilon\} \bigcup \{x < -\epsilon\}$, where $\epsilon = \{\epsilon_1, \ldots, \epsilon_n\}^T$. The equilibria $\mathcal{A}$ corresponds to all members displaying the same voting preference. Therefore, from Lemma 2.11, if $\mathcal{D}$ is strongly connected then, as $\gamma \to 0$, the social model (2.3) has $x(t) \to \mathcal{A}$ for all initial conditions. In other words, the members will eventually reach a consensus on public voting preference.

We consider a ten member, strongly connected interaction digraph $\mathcal{D}$, depicted in Fig. 2.1, and apply model (2.3) with randomly chosen variables $x(0) \in (-\mathbf{1}, \mathbf{1})$, $\epsilon \in (-\mathbf{1}, \mathbf{1})$, and $w_{ij} \in (0, 1]$ for all $\{j, i\} \in E$. The resulting $x - \epsilon$ and $f(x)$ trajectories are displayed in Fig. 2.2. We observe initially the vote was split with six members for and four members against the ballot. Eventually all members converge to a consensus against the ballot.

2.3.2 *Example 2: Sensor Network*

We revisit the sensor surveillance task of Sect. 1.3.2 and consider a new coverage model of the form $z_i(t) = \beta\sqrt{x_i(t)}$ (wedge-shaped sensor) for some $\beta > 0$. Using

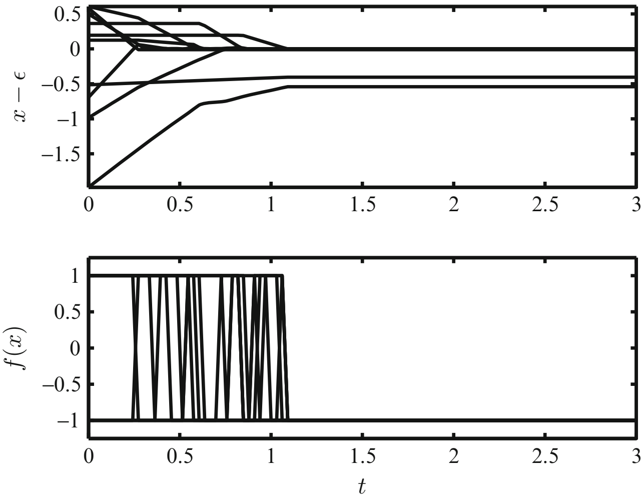


Fig. 2.2 Private belief without bias dynamics $x - \epsilon$ and public ballot votes $f(x)$ over time

the same measure of "good coverage", which aims to minimize gaps between sensor observation areas along the east–west axis and state the derived requirement as

$$\tilde{z}_i = \frac{1}{\delta_i} \sum_{(k,i)\in E} \tilde{z}_k, \text{ for } i = 1, \ldots, n, \tag{2.4}$$

where $\delta_i > 0$ in the number of in-degree neighbors of i and E is the edges of some digraph $\mathcal{G}$. Using the digraph $\mathcal{G}$ defined in Sect. 1.3.2, the selected dynamic model is the out-degree Laplacian dynamics (or advection dynamics)

$$\dot{x} = -\mathscr{L}_{\text{out}}(\mathcal{D})\, f(x), \tag{2.5}$$

where $f(x) = [f_1(x_1), \ldots, f_n(x_n)]^T$, $f_i(x_i) = \sqrt{x_i}$ for $x_i \geq 0$ and $f_i(x_i) = 0$ otherwise. The choice of this dynamics model is justified by the dynamics' equilibrium satisfying

$$\sqrt{x_i} = \frac{1}{\delta_i} \sum_{(k,i)\in E} \sqrt{x_k}, \text{ for } i = 1, \ldots, n, \tag{2.6}$$

which is equivalent to (2.4). From Proposition 2.7 and the fact $\mathbf{1}^T L_{\text{out}}(\mathcal{G}) = \mathbf{0}^T$, then $\sum x_i(0) = \mathbf{1}^T x(0)$. There is exactly one equilibrium satisfying condition (2.6) when $\mathbf{1}^T x(0) > 0$, specifically x_e where $[x_e]_i = \frac{1}{n}\mathbf{1}^T x(0) v_i^2$ and v is the normalized right eigenvector of $\mathscr{L}_{\text{out}}(\mathcal{D})$ corresponding to the zero eigenvalue. As $f(x) \in F_0$ and $\mathscr{L}_{\text{out}}(\mathcal{D}) \in M_0$ then by Lemma 2.11, the dynamics (2.5) will converge to x_e.

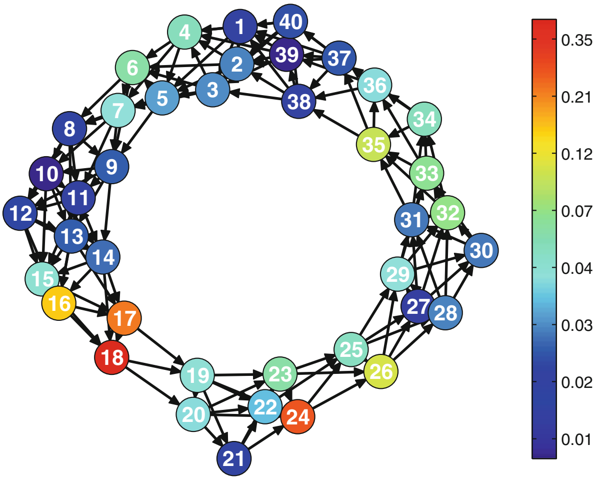


Fig. 2.3 Sensor digraph with color gradations corresponding to the equilibrium power x_e. Nodes are numbered from west to east

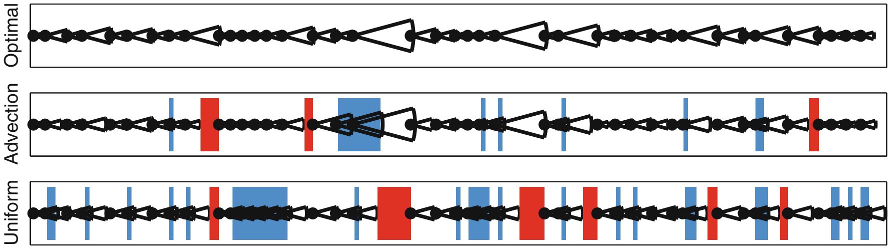


Fig. 2.4 Optimal, advection dynamics and uniform sensor coverage of the land corridor. The dark (*red*) shaded bands indicate areas not covered by any sensors, and light (*blue*) shaded bands indicate areas redundantly covered by multiple sensors

The same initial conditions were applied as in Sect. 1.3.2. The final equilibrium power x_e overlaid on the digraph $\mathcal{G}$ is displayed in Fig. 2.3. Figure 2.4 depicts the observation cones and uncovered and redundantly covered sensor areas for (a) the optimal power usage from all sensors (providing sensor coverage sufficient to cover the east–west axis without redundant coverage), (b) the equilibrium power usage obtained using the advection protocol, and (c) a uniform power usage for all sensors. We find that the minimum power requirement by the advection equilibrium power to cover the corridor is within 1.25 times of the optimal power.

We now extend model (2.1) to incorporate additional terms pertaining to the individual node's dynamics separate from the network dynamics.

2.4 Extension

Consider a modification of model (2.1) via the addition of a nonlinear term to each agent that is only dependent on that agent's state. This term represents the node's dynamics independent of network interactions. The model then becomes

$$\dot{x} = -g(x) - Af(x), \tag{2.7}$$

where $g(x) = [g_1(x_1), \dots, g_n(x_n)]^T \in \mathbb{R}^n$ and $g_i : \mathbb{R} \to \mathbb{R}$ is a scalar-valued function. We assume that $f_i(x_i)g_i(x_i) \geq 0$. The class of functions that satisfy this requirement we denote by $g(\cdot) \in G_0(f(\cdot))$. The equilibria set for model (2.1) is

$$\mathcal{B} = \{x \in \mathbb{R}^n | g(x) + Af(x) = 0\}.$$

We now provide an equivalent asymptotic convergence property to Lemma 2.11.

Lemma 2.12. *Under the assumption that $f(x) \in F_0$, $g(x) \in G_0(f(\cdot))$ and $A \in M_0$, model (2.1) has $x(t) \to \mathcal{B}$ for all initial conditions. If $\mathcal{B}$ is composed of isolated equilibria, then $x(t) \to x_e$ for some $x_e \in \mathcal{B}$.*

Proof. Using the same Lyapunov function $V(x)$ as in the proof of Lemma 2.11 and taking the derivative of $V(x)$ we have

$$\begin{aligned}
\dot{V}(x) &= \sum_{i=1}^{n} [D]_{ii} f_i(x_i)\dot{x}_i \\
&= f(x)^T D\dot{x} \\
&= -f(x)^T D\,(Af(x) + g(x)) \\
&= -f(x)^T DAf(x) - f(x)^T Dg(x) \\
&= -\frac{1}{2} f(x)^T \left(DA + A^T D\right) f(x) - \sum_{i=1}^{n} [D]_{ii} f_i(x_i)g_i(x_i)
\end{aligned}$$

Now, from the property $DA + A^T D \succeq 0$ and $f_i(x_i)g_i(x_i) \geq 0$ which implies $\sum_{i=1}^{n} [D]_{ii} f_i(x_i)g_i(x_i) \geq 0$, $\dot{V}(x) \leq 0$ where $\dot{V}(x) = 0$ for the set $\mathcal{B}$. As in the proof for Lemma 2.11, the set $\Omega = (x \in \mathbb{R}^n | V(x) \leq c)$ is compact and positively invariant. From LaSalle's theorem [19], we conclude that every trajectory starting in Ω asymptotically converges to the set $\mathcal{B}$ as $t \to \infty$. Moreover since $V(x)$ is radially unbounded, the conclusion is global, as for every $x(0)$ there exists a c such that $x(0) \in \Omega$. Further, if all equilibria are isolated, then the trajectory must approaches one of these equilibria. □

The additive neural network is an example of the model relevant to Lemma 2.12 and will be described in the following section.

2.4.1 Example 3: Neural Network

We consider a neural network described by the model

$$\dot{x} = -Bx - Af(x), \tag{2.8}$$

where $B \in \mathbb{R}^{n \times n}$ is a positive diagonal matrix, $A \in M_0$, $f(x)$ is a differential function with $f_i(0) = 0$, $0 < \frac{d}{dx_i} f_i(x_i) \leq \beta$. Consequently, as $f(x)$ is an increasing function $f(x) \in F_0$. If we let $g(x) = Bx$ where $g_i(x_i) = [B]_{ii} x_i$, then as $x_i \geq 0$ $(x_i \leq 0)$ implies $f_i(x_i) \geq 0$ and $g_i(x_i) \geq 0$ $(f_i(x_i) \leq 0$ and $g_i(x_i) \geq 0)$, then $g_i(x_i) f(x_i) \geq 0$. Hence, $g(x) \in G_0(f(\cdot))$. Model (2.8) corresponds to an additive neural network [20]. Here, $x_i(t)$ is the state of a neuron, $-[B]_{ii} x_i$ is a dampening term, $f_i(x_i)$ is its output state which effects the dynamics of other neurons in the network via the weighting matrix A. There is a great interest establishing the global asymptotic stability of such systems with application to optimization and cognitive problems [20–22].

It was shown in [20] that $x = \mathbf{0}$ is a unique equilibrium. As model (2.8) satisfies the assumptions of Lemma 2.12, then $x = \mathbf{0}$ is globally asymptotically stable.

2.5 Remarks

This chapter presents an analysis of a class of nonlinear dynamic networks involving M-matrices, of which nonlinear consensus is a member. Properties of the model's M-matrix were related to its underlying network interaction topology. We explored the equilibria of the network and derived sufficient conditions for an isolated equilibrium. Asymptotic stability of the model with non-identical nodes and strongly directed network topology was proven under continuity and boundedness assumption on the nonlinear dynamics. The model was also extended with an additive nonlinear term and a similar convergence condition was provided. Three applications that are examples of M-matrix models were presented. Future work of particular interest involves the introduction of control terms into the dynamics and the subsequent examination of the model's stability.

References

1. M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton: Princeton University Press, 2010.
2. R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York: Cambridge University Press, 1990.
3. A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*. Academic Press, 1979.

4. A. Chapman and M. Mesbahi, "Advection on graphs," in *Proc. 50th IEEE Conference on Decision and Control*, 2011, pp. 1461–1466.
5. R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.
6. H. G. Tanner, G. J. Pappas, and V. Kumar, "Leader-to-formation stability," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 443–455, 2004.
7. T. Liu, D. Hill, and J. Zhao, "Asymptotic stability of dynamical networks," in *30th Chinese Control Conference (CCC)*, vol. 0, 2011, pp. 928–933.
8. M. Araki and B. Kondo, "Stability and transient behavior of composite nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 17, no. 4, pp. 537–541, 1972.
9. J. Xiang and G. Chen, "On the V-stability of complex dynamical networks," *Automatica*, vol. 43, no. 6, pp. 1049–1057, Jun. 2007.
10. J. Cortés, "Distributed algorithms for reaching consensus on general functions," *Automatica*, vol. 44, no. 3, pp. 726–737, Mar. 2008.
11. Q. Hui, W. M. Haddad, and S. P. Bhat, "Finite-Time Semistability and Consensus for Nonlinear Dynamical Networks," *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1887–1900, Sep. 2008.
12. A. Ajorlou and A. Momeni, "Convergence analysis for a class of nonlinear consensus algorithms," in *Proc. American Control Conference*, 2010, pp. 6318–6323.
13. W. Yu and G. Chen, "Consensus in Directed Networks of Agents with Nonlinear Dynamics," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1436–1441, 2011.
14. R. Varga, *Matrix Iterative Analysis*. Upper Saddle River: Prentice Hall, 1962.
15. R. J. Plemmons, *A survey of the theory and applications of M-Matrices*. Knoxville, Tenn. University of Tennessee, Computer Science Dept., 1976.
16. A. Berman, R. Varga, and R. Ward, "Matrices with nonpositive off-diagonal entries," *Linear Algebra and its Applications*, vol. 21, no. 3, pp. 233–244, 1978.
17. H. Guo, M. Y. Li, and Z. Shuai, "A graph-theoretic approach to the method of global Lyapunov functions," *Proceedings of the American Mathematical Society*, vol. 136, no. 08, pp. 2793–2802, Mar. 2008.
18. T. M. Apostol, *Mathematical Analysis*. Reading: Addison-Wesley, 1974.
19. H. K. Khalil, *Nonlinear Systems*. Upper Saddle River: Prentice Hall, 1996.
20. M. Forti, A. Liberatore, S. Manetti, and M. Marini, "On absolute stability of neural networks," in *1994 IEEE International Symposium on Circuits and Systems*, vol. 6, 1994, pp. 241–244.
21. S. Arik and V. Tavsanoglu, "A sufficient condition for absolute stability of a larger class of dynamical neural networks," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 5, pp. 758–760, 2000.
22. H. Qiao, J. Peng, and Z. Xu, "A reference model approach to stability analysis of neural networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 33, no. 6, pp. 925–36, Jan. 2003.

Part II
Network Measures and Adaptive Topologies

Chapter 3
Measures and Rewiring

Abstract Examining the effectiveness of control in networked systems is a thriving research area. If the semi-autonomous agents' interaction dynamics are consensus-based, we dub this subclass as *semi-autonomous consensus*, which is the focus of the chapter. Within such a subclass, we consider the dynamics of networked agents in the context of performance (friendly influence) and security (unfriendly influence). Our approach to appraise a semi-autonomous consensus network is to expose the network to fundamental test signals, namely white noise and an impulse, and use the resultant system response to quantify network performance and security. Traditionally, input–output properties are varied by altering the dynamics of the network agents. We instead adopt *topological* methods for this task, designing five protocols for tree graphs that rewire the network topology, leaving the network agents' dynamics untouched. In pursuit of this objective, four adaptive protocols are introduced to either increase or decrease the mean tracking and variance damping measures, respectively. Finally, a proposed fifth hybrid protocol is shown to have a guaranteed performance for *both* measures using a game-theoretic formalism.

3.1 Introduction

Consensus-based systems provide effective means of distributed information-sharing and control for networked, multi-agent systems in settings such as multi-vehicle control, formation control, swarming, and distributed estimation; see, for example, [2–6]. One of the appeals of consensus algorithms is their ability to operate *distributively* and *autonomously* over simple *trusting* agents. This has the added benefit that external (control) agents, perceived as *native* agents, can seamlessly attach to the network and steer it in particular directions. These additional agents, ignoring consensus rules, will *influence* the system dynamics compared to the *unforced* networked system resulting in scenarios such as leader–follower [3, 6] and drift correction [8]. The detriment is that this same approach can be adopted by malicious infiltrating agents. We refer to consensus-based systems, with friendly and/or unfriendly attached nodes, as *semi-autonomous consensus* networks. Although the convergence properties of consensus algorithms have been

A. Chapman, *Semi-Autonomous Networks*, Springer Theses,
DOI 10.1007/978-3-319-15010-9_3

extensively studied, examining the network input–output properties in a controlled setting, and their interpretation, is in its infancy—studied in such recent works as [9, 10, 29].

For a semi-autonomous consensus network exposed to either (or both) friendly and unfriendly agents, it is necessary to reason about either (or both) *performance* and *security*. *Performance* (friendly external agents) in the traditional undirected consensus is a well-researched problem in consensus-type algorithms with a general favoritism for the second smallest eigenvalue of the graph Laplacian as a metric to quantify the convergence rate [4, 11], though interest has been shown with other network measures as well, for example the largest eigenvalue of the graph Laplacian [12]. These metrics prove less attractive in a semi-autonomous consensus setting where convergence rates can vary dramatically based on where in the network external agents attach. An alternative is to examine worst, best, or average case convergence of the directed network formed by treating external agents as native agents [6, 30]. Network design to improve some of these measure is explored in [12–14].

In regard to *security* (unfriendly external agents), most modern day semi-autonomous networks rely on access security to the network which is unsuited to a trusting semi-autonomous consensus setting. An alternative to generate a secure network is intrusion detection[1] coupled with either inter-agent security through each agent's dynamics or intra-agent security via the network topology. The former includes implementation of disturbance rejection or agent disabling techniques, e.g., noise canceling systems and power grid "brown outs". The latter involves global or local network rewiring, e.g., TCP network re-routing. This adaptive topology approach for security as well as performance is the main focus of the present work.

Network performance and security via adaptive topology (intra-agent security) is a largely unexplored area within a semi-autonomous consensus setting. In addition, this is the only security response available in the case where the agent dynamics and interaction protocol are assumed to be fixed or expensive to alter. Examples of such systems include networks with hardwired dynamics and interactions, for example due to safety and the required performance guarantees, and systems with physically and biologically motivated dynamics and interactions, e.g., diffusion across self-assembly units and bio-inspired networks.

Furthermore, in the case where the intent of external agents may only be known probabilistically, network rewiring presents a security option that is less dramatic than altering agent dynamics and interactions. Characteristics such as the consensus value of a diffusion network are maintained under network rewiring which is not generally the case when the agents dynamics and interactions are altered due to a security and performance criteria—an attractive property in the event that one misdiagnoses the intent of the external agent.

[1] Techniques for intrusion or fault detection on consensus-type networks include those based on reachability analysis [15], and the more popular unknown-input observers [10, 16, 18].

The main difficulty for analyzing networks where both friendly and unfriendly agents can attach is that features that are conducive to security are not generally favorable for performance. Our work provides metrics for both performance and security and discusses associated topological features that can be used to design performant and secure networks. We also propose protocols that rewire the network topology in order to exploit these topological features.

In this chapter we examine the performance and security of a network in response to an external agent injecting a *test signal*, namely a white Gaussian signal or an impulse, into the network. The performance and security of the network is measured in terms of the subsequent mean and variance of the agents' state; we refer to these metrics as the *mean tracking* measure and *variance damping* measure. Both measures are used to propose five decentralized protocols for tree graphs that adaptively aim to improve or degrade either the performance or the security of the network by undertaking local edge swaps. Figure 3.1 illustrates these concepts where the graph topology is used to vary the output characteristics of the network.

Two motivating applications for the present work and the utility of the metrics as they relate to each are:

- State estimation—where x_i is the estimate of the node i's state, through consensus, can be globally estimated via relative node state exchanges between neighboring agents, e.g., drift correction and time synchronization [17, 19]. Consider external agents that do not accept information exchanges from other agents and deliver instead a Gaussian white noise with unit intensity. The external agents' disregard of the consensus dynamics may be due to superior sensing

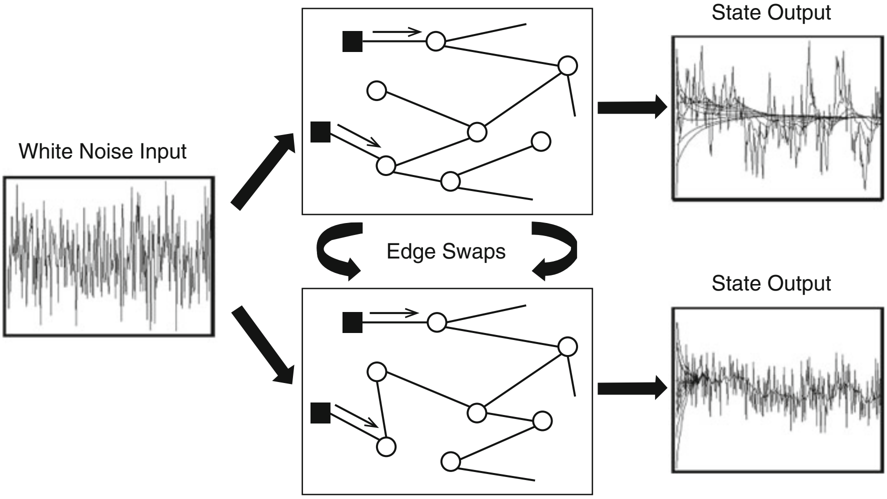


Fig. 3.1 Illustration of the problem setup: External agents (*squares*) inject white noise into the network manipulating the state output of the native agents (*circles*). A protocol performs edge swaps to alter the state output, specifically the network's mean tracking and variance damping measure

compared to native agents, malfunctions, or malicious intent. Viewing the difference between a states' value and the external agents' as the state error, the mean tracking measure is the expected quadratic performance of this state error. The variance damping measure, on the other hand, is the expected nodes' error variance. The intent of an external agent can only be known probabilistically as such security in the system is left to the less intrusive adaptive topological methods leaving the agent and interaction dynamics unchanged.

- Flocking—where $x_i \in \mathbb{R}^m$ (e.g., $m = 2, 3$) is the velocity of agent i, $\dot{x}_i = u_i$, and u_i is dependent on the relative velocities of neighboring agents, e.g., UAV flocking and fish swarming [6, 11]. A node can then be considered as an external agent guiding the flock by ignoring consensus with either friendly or malicious intent. The ease with which the flock tracks this agent while the agent holds its velocity constant can be gauged using the mean tracking measure. If the agent undertakes a sudden impulse-like maneuver the damping of its state's propagation through the network can be quantified by the variance damping measure. The underlying interaction dynamics are fixed due to the nature of the onboard relative sensors. The agent dynamics are fixed to guarantee predesignated performance characteristics such as control rate and interagent distance bounds. Subsequently, performance and security methods can rely only on adaptive topology methods.

To clarify the contributions of this chapter, it is worthwhile to compare our results and approach with similar works in literature. Designing topologies to optimize for certain metrics has previously been addressed; by Ghosh et al. [13] for maximizing the second smallest eigenvalue of the graph Laplacian, by Zelazo and Mesbahi [12] for optimizing the network $\mathcal{H}_2$ performance, and by Wan et al. [14] for maximizing the largest eigenvalue of the graph Laplacian, each using optimization techniques over *weighted* graphs. Our problem of edge swaps, considered in this chapter in an *optimization setting* would require NP-hard mixed-integer programming. We have thus opted for a game-theoretic formalization to quantify network performance and security. The protocol's effectiveness is qualified using the suboptimality properties of the Nash equilibria by modeling the external-native agent dynamics as a non-cooperate game [8, 20]. The simplest form of adaptive network security is the removal of those nodes in the network connected to infiltrators [15, 21]. Using percolation theory, Callaway et al. [22] illustrated this to be a potentially disruptive remedy as it can cause the network to become disconnected even for highly dense graphs which subsequently presents an attack vector that an infiltrator could exploit, e.g., by falsely tagging trustworthy agents as untrustworthy. Tyson et al. [23] have discussed intuitive methods of network reconfiguration to improve resilience, specifically using thresholding methods to decide when to alter the topology. Security techniques that involve adapting the agent dynamics to compensate for k infiltrators have been addressed for $2k + 1$ connected graphs for Byzantine faults and $k + 1$ connected graphs for general faults [18, 24].

The main contributions of this chapter are threefold. First, a pair of network measures are proposed. The mean tracking measure is the average quadratic performance measure of the error in response to the test signal and is linked to the network structure via an electrical network analogy. The variance damping measure, on the other hand, is the expected mean square error of the states, which can be calculated using the controllability gramian and can also be related to the network structure using an electrical analogy. Second, four protocols are developed to optimize the network topology with respect to the proposed measures. These protocols each locally rewires the network topology, using edge swaps between neighboring nodes, to favorably increase or decrease each measure respectively, but not concurrently, thus improving global performance with respect to friendly or unfriendly attached agents. Finally, we formulate the metrics in terms of the effective resistance of an electrical network and, in so doing, illustrate the coupling between mean tracking and variance damping measures. This has motivated the development of a hybrid protocol using a game-theoretic formalism that provides a guaranteed balance between input rejection with respect to the mean and variance measures, particularly useful in security scenarios. All protocols perform edge swaps (rewiring) which can be executed in parallel, asynchronously, and require only local agent information of the network structure. The protocols are applied to two motivating applications, namely time synchronization and UAV flocking.

The chapter is organized as follows. Section 3.2 contains the problem formulation and relevant background. The mean tracking measure is examined in Sect. 3.3 and its relationship to the effective resistance is established and subsequently used to design two protocols for increasing and decreasing the mean tracking measure. A similar treatment of the variance damping measure is presented in Sect. 3.4. Section 3.5 presents a more versatile protocol that provides guarantees on both measures analyzed using game-theoretic techniques. We conclude the chapter with a few remarks in Sect. 3.6.

3.2 Leader–Follower Consensus Dynamics

We next introduce a model of multi-input *influenced* consensus over an undirected graph $\mathcal{G} = (V, E)$ associated with a pair $\mathcal{R} = (R, \mathcal{E}_R)$, where R is the cardinality r external agent set and $\mathcal{E}_R \subseteq R \times V$ is the set of edges used by the external agents to inject signals into the network. It is assumed that each external agent $r_j \in R$ is attached to exactly one node $v_i \in V$ along one of the r edges $\{r_j, v_i\} \in \mathcal{E}_R$ and subsequently delivers a signal $u_j(t) \in \mathbb{R}$. Figure 3.2 provides a graphical representation of this notation and setup.

The resulting influenced system, popularly known as *leader–follower consensus* dynamics [6], now assumes the form,

$$\dot{x}_i(t) = \sum_{\{v_i,v_j\}\in E} \big(x_j(t) - x_i(t)\big) + \sum_{\{v_i,r_j\}\in \mathcal{E}_R} \big(u_j(t) - x_i(t)\big) \tag{3.1}$$

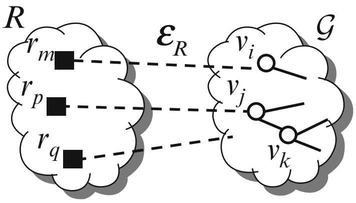


Fig. 3.2 Example of influence notation

with the full dynamics

$$\dot{x}(t) = A(\mathcal{G}, \mathcal{R})x(t) + B(\mathcal{R})u(t), \tag{3.2}$$

where $B(\mathcal{R}) \in \mathbb{R}^{n\times r}$ with $[B(\mathcal{R})]_{ij} = 1$ when $\{r_j, v_i\} \in \mathcal{E}_R$ and $[B(\mathcal{R})]_{ij} = 0$ otherwise, and

$$A(\mathcal{G}, \mathcal{R}) = -(\mathscr{L}(\mathcal{G}) + M(\mathcal{R})) \in \mathbb{R}^{n\times n}, \tag{3.3}$$

where $M(\mathcal{R}) := B(\mathcal{R})B(\mathcal{R})^T \in \mathbb{R}^{n\times n}$. We also introduce a special type of single-agent control as $\mathcal{R}^i$ where $R = \{r_1\}$ and $\mathcal{E}_R = \{r_1, v_i\}$. Further the set of agents v_i such that $\{r_j, v_i\} \in \mathcal{E}_R$ for some r_j will be denoted by $\pi(\mathcal{E}_R)$; this is the set of native agents that directly connect to external agents.

We recognize $A(\mathcal{G}, \mathcal{R})$ in (3.3) as the Dirichlet matrix, or grounded Laplacian [19, 25]. The spectrum of $A(\mathcal{G}, \mathcal{R})$ relates closely to the spectrum of $\mathscr{L}(\mathcal{G})$. In this way, the structure of the underlying graph is related to the dynamics of model (3.2).

The following result highlights this connection.

Proposition 3.1. *The eigenvalues of the matrix $-A(\mathcal{G}, \mathcal{R})$ in (3.3) satisfy the following inequalities:*

(a) $\lambda_j(\mathcal{G}) \leq \lambda_j(-A(\mathcal{G}, \mathcal{R}))$,
(b) $\lambda_j(-A(\mathcal{G}, \mathcal{R})) \leq \lambda_j(\mathcal{G}) + 1$, *and*
(c) for $j > 1$, $\lambda_{j-1}(-A(\mathcal{G}, \mathcal{R})) \leq \lambda_j(\mathcal{G})$.

Proof. The matrix $-A(\mathcal{G}, \mathcal{R}^i)$ is the sum of two positive semidefinite matrices $\mathscr{L}(\mathcal{G})$ and $M(\mathcal{R})$. As such, the matrix $-A(\mathcal{G}, \mathcal{R})$ is positive semidefinite. By the eigenvalue interlacing theorem [7, Corollary 4.3.3 and Theorem 4.3.6], bounds (a) and (c) follow. Moreover, Weyl's Theorem [7, Theorem 4.3.1] implies that

$$\lambda_j(-A(\mathcal{G}, \mathcal{R}^i)) = \lambda_j(\mathscr{L}(\mathcal{G}) + M(\mathcal{R})) \leq \lambda_j(\mathcal{G}) + \lambda_n(M(\mathcal{R})) = \lambda_j(\mathcal{G}) + 1.$$

□

An auxiliary observation on the Dirichlet matrix, to be used subsequently, is the following.

Proposition 3.2 ([30]). *The matrix $A(\mathcal{G}, \mathcal{R})$ of model (3.2) is negative definite (and so invertible) if the original graph is connected.*

Proof. If the graph $\mathcal{G}$ is connected, then the span of the null space of $\mathscr{L}(\mathcal{G})$ is $\mathbf{1}$ and $\mathbf{1}$ is not in the null space of $M(\mathcal{R})$. Thus $x^T\mathscr{L}(\mathcal{G})x > 0$ for $x \neq \alpha\mathbf{1}$ for $\alpha \in \mathbb{R}$ and $\mathbf{1}^T M(\mathcal{G})\mathbf{1} > 0$. Moreover, for $x \neq \mathbf{0}$,

$$x^T\left(-A(\mathcal{G},\mathcal{R})\right)x = x^T\left(\mathscr{L}(\mathcal{G}) + M(\mathcal{R})\right)x^T = x^T\mathscr{L}(\mathcal{G})x + x^T M(\mathcal{R})x > 0,$$

and the statement of the proposition follows. □

Remark 3.3. A popular model for network intrusion or faults is to consider native agents as those ignoring consensus instead of attaching external agents [9, 10, 18]. This model can be adapted by considering the subgraph of behaving agents as $\mathcal{G}$ and each edge between the misbehaving agents and $\mathcal{G}$ as $\left\{r_j, v_i\right\} \in \mathcal{E}_R$ corresponding, in our model, to a behaving agent v_i at one end and an external agent $r_j \in R$ at the other. The presented analysis is therefore applicable to both models. As this chapter has a particular focus on tree graphs it is worthwhile mentioning that if the original graph is a tree then $\mathcal{G}$ in (3.2) will be the union of tree graphs. The converse is not true; the original graph need not be a tree if $\mathcal{G}$ in (3.2) is a tree.

We approach the network performance and security problem from two fronts; first via the cost to the network for its agents to track a constant signal—dubbed *mean tracking* measure (discussed in Sect. 3.3), and second as the cost to the network to dampen a noisy external agent's signal—dubbed *variance damping* measure (discussed in Sect. 3.4). The following two sections will focus on these measures.

3.3 Mean Tracking Measure

The mean tracking measure is a metric for the effectiveness of a network, via its topology, to track a grounded shared constant external agents' signal $u_c \in \mathbb{R}$. This metric is equally applicable to the network's performance in regard to tracking the mean u_c of external agents' signal, e.g., Gaussian noise about u_c, hence the metric's name *mean* tracking. We derive the mean tracking measure as the cost incurred by external agents to steer the mean state of the entire network to u_c, over an infinite horizon. In order to quantify the performance and security of the network to resist the influence of external agents injecting random signals, the following two observations are in order: (1) the dynamics of the mean of agents' state is captured by model (3.2) where $[u]_i$ is replaced by the mean of external agents' signal u_c, (2) when the underlying graph is connected, all agents' state converge in the mean to u_c. The last statement is a direct consequence of Proposition 3.2. More specifically, noting that $u(t) \equiv \mathbf{1}_n u_c$ and $A(\mathcal{G},\mathcal{R})^{-1}B\mathbf{1}_r = -\mathbf{1}_n$, the quadratic performance cost of the mean, with coordinate change $\tilde{x}(t) = \mathbb{E}\left(x(t)\right) - u_c\mathbf{1}_n$, where $[\mathbb{E}(x(t))]_i$ is the expected value of the variable $x_i(t)$ at time t, can be derived as[2]

[2] The scaling by 2 is cosmetic.

$$
\begin{aligned}
2\int_0^{t_f} \tilde{x}(t)^T\tilde{x}(t)dt &= \int_0^{t_f} \tilde{x}(t)^T x(t) + x(t)^T\tilde{x}(t) - \tilde{x}(t)^T \mathbf{1}_n u_c - u_c \mathbf{1}_n^T \tilde{x}(t)dt \\
&= \int_0^{t_f} \tilde{x}(t)^T x(t) + x(t)^T\tilde{x}(t) + \tilde{x}(t)^T A^{-1}B\mathbf{1}_r u_c + u_c\mathbf{1}_r^T B^T A^{-1}\tilde{x}(t)dt \\
&= \int_0^{t_f} \left(Ax(t) + B\mathbf{1}_r u_c\right)^T A^{-1}\tilde{x}(t) + \tilde{x}(t)^T A^{-1}\left(Ax(t) + B\mathbf{1}_r u_c\right) dt \\
&= \int_0^{t_f} \dot{x}(t)^T A^{-1}\tilde{x}(t) + \tilde{x}(t)^T A^{-1}\dot{x}(t)dt \\
&= \int_0^{t_f} \frac{d}{dt}\tilde{x}(t)^T A^{-1}\tilde{x}(t)dt \\
&= \tilde{x}(t_f)^T A^{-1}\tilde{x}(t_f) - \tilde{x}(0)^T A^{-1}\tilde{x}(0).
\end{aligned}
$$

In order to parametrize the performance and security of the network for a specific set $\mathcal{R}$, let us define the accumulative state mean over the length of time the input is applied t_f as

$$
\begin{aligned}
J_\mu\left(\mathcal{G}, \mathcal{R}, t_f\right) &= \mathbb{E}_{\|\tilde{x}(0)\|=1}\left(2\int_0^{t_f} \tilde{x}(t)^T\tilde{x}(t)dt\right) \\
&= \mathbb{E}_{\|\tilde{x}(0)\|=1}\left(\tilde{x}(t_f)^T A^{-1}\tilde{x}(t_f) - \tilde{x}(0)^T A^{-1}\tilde{x}(0)\right) \\
&= \mathbb{E}_{\|\tilde{x}(0)\|=1}\mathbf{tr}\left(\tilde{x}(0)\tilde{x}(0)^T\left(\left(e^{At_f}\right)^T A^{-1}e^{At_f} - A^{-1}\right)\right) \\
&= \mathbb{E}_{\|\tilde{x}(0)\|^2=n}\mathbf{tr}\left(\frac{1}{\sqrt{n}}\tilde{x}(0)\frac{1}{\sqrt{n}}\tilde{x}(0)^T(e^{At_f}A^{-1}e^{At_f} - A^{-1})\right) \\
&= \frac{1}{n}\mathbf{tr}\left(\left(\mathbb{E}_{\|\tilde{x}(0)\|^2=n}\tilde{x}(0)\tilde{x}(0)^T\right)\left(e^{At_f}A^{-1}e^{At_f} - A^{-1}\right)\right) \\
&= \frac{1}{n}\mathbf{tr}\left(I\left(e^{2At_f} - I\right)A^{-1}\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\lambda_i\left(-A\right)}\left(1 - e^{-2\lambda_i(-A)t_f}\right),
\end{aligned}
$$

where $\mathbb{E}_{\|\tilde{x}(0)\|=1}(\cdot)$ denotes the expected value over all initial conditions satisfying $\|\tilde{x}(0)\| = 1$.

It is assumed that the native agents in the network do not know the value of t_f; as such in the remaining parts of our chapter we assume that t_f is large, justifying the use of $J_\mu\left(\mathcal{G}, \mathcal{R}, \infty\right)$ as the mean tracking measure. For brevity, we denote $J_\mu\left(\mathcal{G}, \mathcal{R}, \infty\right)$ as $J_\mu\left(\mathcal{G}, \mathcal{R}\right)$. Metrics similar to $J_\mu\left(\mathcal{G}, \mathcal{R}, \infty\right)$, based on the minimum and maximum value of $2\int_0^\infty \tilde{x}(t)^T\tilde{x}(t)dt$ over initial conditions lying on the unit ball. Bounds pertaining for a single input case have been relegated to the Appendix.

We now formally define our $J_\mu\left(\mathcal{G}, \mathcal{R}\right)$ metric.

Definition 3.4. The *mean tracking* measure of a network is the average quadratic performance cost incurred by external agents to steer the mean state of the entire network to their respective mean value, over an infinite horizon, and is equal to

$$J_\mu\left(\mathcal{G},\mathcal{R}\right) = \frac{1}{n}\mathbf{tr}\left(-A(\mathcal{G},\mathcal{R})^{-1}\right). \tag{3.4}$$

Remark 3.5. We briefly note the connection between the more familiar best case convergence rate of the grounded dynamics, the average convergence rate of the grounded dynamics and the mean tracking measure, i.e., the minimum nonzero eigenvalue of the Laplacian, the average of the eigenvalues of the Laplacian and the measure $J_\mu\left(\mathcal{G},\mathcal{R}\right)$ (3.4). Consider fusing all external agents to form a node v_{n+1} and adding it to our graph $\mathcal{G}$, connecting v_{n+1} to the network through "directed" edges from v_{n+1} to each node in $\pi\left(\mathcal{E}_R\right)$; we call this new "directed" graph $\tilde{\mathcal{G}}$.[3] Then $\mathcal{G}$ and $\tilde{\mathcal{G}}$ have the property $\lambda_2(\tilde{\mathcal{G}}) \le J_\mu\left(\mathcal{G},\mathcal{R}\right)^{-1} \le \frac{1}{n}\sum_{i=1}^{n+1}\lambda_i(\tilde{\mathcal{G}})$.[4]

The following section will provide more insights into the mean tracking measure.

3.3.1 Analysis of Mean Tracking Measure

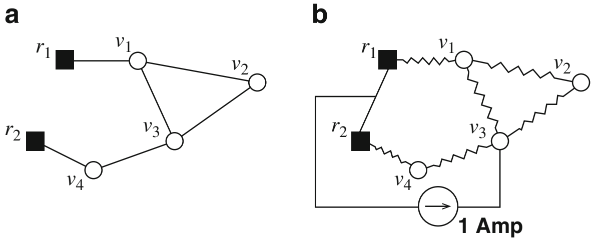
It has previously been established that the diagonal of the matrix $-A(\mathcal{G},\mathcal{R})^{-1}$, where $A(\mathcal{G},\mathcal{R})$ is the Dirichlet matrix in (3.2), has a resistive electrical network interpretation [19]. In this setup, the agents V and R, defined in Sect. 2.2, represent respectively, connection points between resistors corresponding to the edges E and $\mathcal{E}_R$. In addition, all connection points corresponding to the set R are electrically shorted. The effective resistance between two connection points in an electrical network is defined as the voltage drop between the two points, when a 1 Amp current source is connected across the two points. Then, the i-th diagonal element of $-A(\mathcal{G},\mathcal{R})^{-1}$ is the effective resistance $E_{\text{eff}}(v_i)$ between the common shorted external agents R and v_i. An example of the equivalent electrical network is displayed in Fig. 3.3. The implication is that

$$J_\mu\left(\mathcal{G},\mathcal{R}\right) = \frac{1}{n}\sum_{i=1}^{n} E_{\text{eff}}\left(v_i\right). \tag{3.5}$$

Tree graphs are often adopted for agent-to-agent communication topologies as they minimize edge (communication) costs while maintaining connectivity. Using (3.5), we introduce some properties of $J_\mu\left(\mathcal{G},\mathcal{R}\right)$ (3.4) specific to trees.

[3] For a survey of directed graphs, we refer the reader to [6].

[4] All eigenvalues of $\mathscr{L}(\tilde{\mathcal{G}})$ are real, $\lambda_1(\tilde{\mathcal{G}}) = 0$ and $\lambda_{i+1}(\tilde{\mathcal{G}}) = \lambda_i(A(\mathcal{G},\mathcal{R}))$ for $i = 1,\ldots,n$.



$$A(\mathcal{G},\mathcal{R}) = \begin{bmatrix} -3 & 1 & 1 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & 1 & -3 & 1 \\ 0 & 0 & 1 & -2 \end{bmatrix}, \; B(\mathcal{R}) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

Fig. 3.3 (**a**) Network graph with external (control) agents r_1 and r_2 attached to agents v_1 and v_4 respectively, leading to an altered Laplacian $A(\mathcal{G},\mathcal{R})$ and input matrix $B(\mathcal{R})$ of model (3.2). (**b**) Equivalent electrical network. The potential difference $V_{v_3} - V_{\mathcal{R}}$ is the effective resistance between v_3 and common resistor node $\{r_1, r_2\}$

Let us first define the special set of agents that lie on any of the shortest paths between agents in $\mathcal{R}$ as the *main path* agents, designated by the set $\mathcal{M}$. This is a unique set for a given pair $(\mathcal{G},\mathcal{R})$. Moreover for all $v_i \notin \mathcal{M}$, there exists a unique $v_j \in \mathcal{M}$ that has a shorter minimum path to v_i than any other agent in $\mathcal{M}$; we define this agent as $\Gamma(v_i)$, i.e., $\Gamma\ (v_i)$ is the closest agent to v_i that is a member of the main path. Therefore for tree graphs we can state the following.

Lemma 3.6 (Mean Tracking Measure for Trees). *For the n-agent connected tree $\mathcal{T}$, the mean tracking measure is*

$$J_\mu(\mathcal{T},\mathcal{R}) = \frac{1}{n}\left(\sum_{v_i \in \mathcal{M}} E_{\text{eff}}\,(v_i) + \sum_{v_i \notin \mathcal{M}} \left[E_{\text{eff}}\,(\Gamma\,(v_i)) + d\,(v_i, \Gamma\,(v_i))\right]\right).$$

Proof. If $v_i \notin \mathcal{M}$, then the equivalent electrical network involving v_i can be simplified into a resistor representing $E_{\text{eff}}\,(\Gamma\,(v_i))$ ohms in series with $d\,(v_i, \Gamma\,(v_i)) \times 1$ ohm resistors. The result then follows from (3.5). □

There is an intuitive link between the *centrality* of an agent in a network and its influence on the network's dynamics. This correlation becomes apparent for tree graphs in the following.

Corollary 3.7 (Single-External Mean Tracking Measure). *For the n-agent connected tree $\mathcal{T}$ the mean tracking measure of the network to a single external agent attached to any agent $v_i \in V$ is*

$$J_\mu(\mathcal{T},\mathcal{R}^i) = \frac{1}{n}\left(\sum_{j=1}^{n} d\,\left(v_i, v_j\right) + n\right).$$

Proof. The proof follows from Lemma 3.6 with $\{v_i\} = \mathcal{M}$ and $E_{\text{eff}}(v_i) = 1$. □

Corollary 3.7 has a few immediate ramifications. Consider the single external agent as a native node v_{n+1} of the graph $\mathcal{T}$ forming the new graph $\tilde{\mathcal{T}}$ with $n+1$ nodes. The mean tracking measure $J_\mu\left(\mathcal{T}, \mathcal{R}^i\right)$ is then equal to the closeness centrality measure[5] of node v_{n+1}, i.e.,

$$J_\mu\left(\mathcal{T}, \mathcal{R}^i\right) = \frac{1}{n}\sum_{j=1}^{n} d\left(v_{n+1}, v_j\right) = c\left(v_{n+1}, \tilde{\mathcal{T}}\right).$$

Further as

$$J_\mu(\mathcal{T}, \mathcal{R}^i) = \frac{n-1}{n} c\left(v_i, \mathcal{T}\right) + 1,$$

the *most influential node* to attach in a tree graph under the measure J_μ (3.4) is the one with the largest closeness centrality measure.

Corollary 3.8 (Single-External Mean Tracking Measure Bounds). *For the n-agent connected tree $\mathcal{T}$ the mean tracking measure of the network to a single external agent attached to any agent $v_i \in V$ is bounded as* $2 - \frac{1}{n} \leq J_\mu(\mathcal{T}, \mathcal{R}^i) \leq \frac{1}{2}(n+1)$.

Proof. Over all trees, the central node of the star graph has the smallest accumulative distance of $n-1$ to all other nodes and an end node of the path graph has the largest accumulative distance of $\sum_{i=1}^{n-1} i$ to all other nodes. The statement of the corollary follows from these two observations. □

Proposition 3.9 (Multi-External Mean Tracking Measure Bounds). *For the n-agent connected tree $\mathcal{T}$ the mean tracking measure of r external agents attached to any set of agents in V is bounded above by a tree graph with all main path nodes satisfying $v_i \in \pi\left(\mathcal{E}_R\right)$, in which case,*

$$J_\mu(\mathcal{T}, \mathcal{R}) \leq \frac{1}{2n}\left((n-r)^2 + 3\left(n-r\right) + r + \frac{2}{r+1}\right).$$

Proof. From the effective resistance interpretation of $J_\mu(\mathcal{T}, \mathcal{R})$ in (3.5), we note that adding resistors in series generates a higher resistance than adding them in parallel. Therefore, $\operatorname{argmax}_{(\mathcal{T},\mathcal{R})} J_\mu\left(\mathcal{T}, \mathcal{R}\right)$ is a tree $\mathcal{T}$ and the influence set $\mathcal{R}$ where $\mathcal{M} = \pi\left(\mathcal{E}_R\right)$, as adding an agent to the main path places resistors in parallel rather than the alternative which is in series. This family of graphs with $\mathcal{M} = \pi\left(\mathcal{E}_R\right)$, we define as the set $\mathcal{H}$. Furthermore from Lemma 3.6, the largest accumulative distance of these nodes v_i will correspond to a path connected to the highest effective

[5] Closeness centrality $c\left(v_i, \mathcal{G}\right)$ is the mean of the shortest path lengths between node v_i and other nodes in the graph $\mathcal{G}$.

resistance node of the main path subgraph. Now, the main path subgraph, of a tree in $\mathcal{H}$ with the largest effective resistance sum, is the star graph as the equivalent electrical network has the least number of parallel resistors. Applying resistor rules, we thus obtain

$$\max_{(\mathcal{T},\mathcal{R})\subseteq\mathcal{H}} \sum_{v_i\in\mathcal{M}} E_{\text{eff}}(v_i) = \frac{r^2+r+2}{2(r+1)}. \tag{3.6}$$

Similarly, the largest $E_{\text{eff}}\left(v_j\right)$ for any single node $v_j \in \mathcal{M}$ of a tree in $\mathcal{H}$, corresponds to the main path subgraph which is a path and specifically one end node v^* which leads to

$$\max_{(\mathcal{T},\mathcal{R})\subseteq\mathcal{H},v_j\in\mathcal{M}} E_{\text{eff}}\left(v_j\right) = \frac{F_{2r-1}}{F_{2r}} = \frac{\phi^{2r-1}-(-1/\phi)^{2r-1}}{\phi^{2r}-(-1/\phi)^{2r}} \leq 1,$$

where F_i is the i-th Fibonacci number and φ is the golden ratio.[6] The largest effective resistance sum over trees in $\mathcal{H}$ of a non-main path subgraph can now be formed from a path attached to node v^*, i.e.,

$$\begin{aligned}\max_{(\mathcal{T},\mathcal{R})\subseteq\mathcal{H}} \sum_{v_i\notin\mathcal{M}} E_{\text{eff}}(v_i) &= \sum_{i=1}^{n-r}\left[\max_{(\mathcal{T},\mathcal{R})\subseteq\mathcal{H},v_j\in\mathcal{M}} E_{\text{eff}}\left(v_j\right)+1\right] \\ &\leq \sum_{i=1}^{n-r}(1+i) = (n-r)^2+3(n-r). \end{aligned} \tag{3.7}$$

Using bounds (3.6) and (3.7) combined with (3.5) we have

$$\begin{aligned} J_\mu(\mathcal{T},\mathcal{R}) &\leq \frac{1}{n}\left(\max_{(\mathcal{T},\mathcal{R})\subseteq\mathcal{H}} \sum_{v_i\in\mathcal{M}} E_{\text{eff}}(v_i) + \max_{(\mathcal{T},\mathcal{R})\subseteq\mathcal{H}} \sum_{v_i\notin\mathcal{M}} E_{\text{eff}}(v_i)\right) \\ &= \frac{1}{2n}\left((n-r)^2+3(n-r)+r+\frac{2}{r+1}\right). \end{aligned}$$

□

[6] The derivation follows from that of the infinite connected resistor network and the recursive definition of the golden ratio $\phi = \frac{1}{2}\left(1+\sqrt{5}\right)$. One has $F_i = \frac{1}{\sqrt{5}}\left(\phi^i - (-1/\phi)^i\right)$.

3.3.2 *Adaptive Protocols to Improve or Degrade the Mean Tracking Measure for Trees*

We now propose a pair of protocols applicable to tree graphs that locally trade edges, e.g., communication links, between adjacent agents with the objective of deterring or encouraging the influence of external agents attached to the network, feeding in a *constant mean signal*. We consider a scenario where agents in $\pi(\mathcal{E}_R)$ broadcast acknowledgment signals informing the network that they are being favorably or unfavorably influenced. Consequently all agents within the graph are aware of the "local" directions of the external agents, and more specifically, their neighboring agents that are closer to these external agents. We denote by $\mathcal{I}(v_i)$ the set of all agents that are neighbors of v_i and lie on the shortest path between v_i and any $r_j \in R$. We emphasize that we assume that the external agents in set R are solely composed of friendly or unfriendly agents and agents are able to distinguish between the external agents' intent.[7]

The following lemma describes Protocol 1 which can be executed on any arbitrary agent v_i and requires the knowledge of $\mathcal{I}(v_i)$ and $\mathcal{N}(v_i)$; hence, the protocol operates on "local" information. In the following, we denote edge removal and addition by the set notation "$-/+$."

Lemma 3.10 (Edge Swap for Improved Mean Tracking Measure). *Under Protocol 1, $J_\mu(\mathcal{T}, \mathcal{R})$ (3.4) is strictly increasing.*

Proof. If $v_m \in \mathcal{M}$, then for all $v_l \in \mathcal{N}(v_m)$ we have $v_m \in \mathcal{I}(v_l)$. Therefore in the context of Protocol 1, $v_j, v_k \notin \mathcal{M}$. Then from Lemma 3.6, before the edge swap, we have $E_{\text{eff}}(v_j) = E_{\text{eff}}(v_k) = E_{\text{eff}}(v_i) + 1$, and after the edge swap $E_{\text{eff}}(v_j) = E_{\text{eff}}(v_i) + 2$. Any agent v_p such that v_j lies on the shortest path between v_p and any agent in $\mathcal{M}$ will increase its effective resistance by 1 after the edge swap. Since the effective resistance strictly increases or stays the same for all agents following the edge swap, $J_\mu(\mathcal{T}, \mathcal{R})$ increases. □

Some attractions of Protocol 1 is that it can be executed concurrently or in a random agent order, guarantees that $J_\mu(\mathcal{T}, \mathcal{R})$ increases, and maintains a connected tree at each iteration. This is attained without the knowledge of the global network topology.

When all agents adopt Protocol 1, all trees with a single external agent attached to the graph will eventually evolve to a graph with the greatest $J_\mu(\mathcal{T}, \mathcal{R}^1) = \frac{n+1}{2}$, namely a path graph with the external agent at one end. Trees with multiple external agents will acquire a path-like appearance with the main path unaffected by the protocol's edge swaps.

Protocol 1 was applied to a random tree graph on 40-agents with a single external agent connected to v_1. The path graph with the external agent attached to the end was achieved after 100 edge swaps. A sample of the intermediate graphs, the

[7] For unfavorable detection an algorithm such as those proposed in [10, 15, 16, 18] can be used.

Protocol 1 Increased mean tracking measure edge swap

```
foreach Agent $v_i$ do
    if $\exists v_j, v_k \in \mathcal{N}(v_i)$, $v_j \neq v_k$ and $v_j, v_k \notin \mathcal{I}(v_i)$ then
        $E \to E - \{v_i, v_j\} + \{v_j, v_k\}$
    end
end
```

Protocol 2 Decreased mean tracking measure edge swap

```
foreach Agent $v_i$ do
    if $v_k = \mathcal{I}(v_i)$, $\exists v_j \in \mathcal{N}(v_i)$ and $v_j \neq v_k$ then
        $E \to E - \{v_i, v_j\} + \{v_j, v_k\}$
    end
end
```

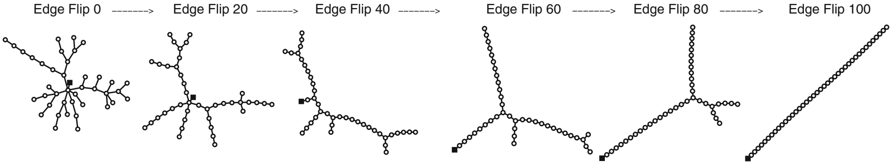


Fig. 3.4 Selected iterations of an adaptive tree graph running Protocol 1 with an external agent attached (*square*)

mean tracking measure over all iterations, and the evolution of the state mean are displayed in Figs. 3.4 and 3.5. The network measure $J_\mu\left(\mathcal{T}, \mathcal{R}^1\right)$ increased for each edge swap and no more edge swaps were possible when the tree became a path graph with $J_\mu\left(\mathcal{T}, \mathcal{R}^1\right) = 20.5$.

A complementary Protocol 2 that aims to decrease $J_\mu\left(\mathcal{T}, \mathcal{R}\right)$ can also be obtained from Lemma 3.10. Under this protocol the graph converges to a star-like graph, while preserving the structure of the main path. The protocol was run on a 40-agent random tree graph with three external agents. The original and final graphs, achieved after 21 edge swaps, are displayed in Fig. 3.6.

Remark 3.11. For agent v_i with access to the "local" information provided by $\mathcal{I}\left(v_i\right)$ and $\mathcal{N}(v_i)$, Lemma 3.10 describes the *only* edge swap protocol that guarantees $J_\mu\left(\mathcal{T}, \mathcal{R}\right)$ increases and a connected tree is maintained. Let us illustrate this by examining edge swap protocols *not* covered by Lemma 3.10; for these edge swap cases v_j and/or v_k can be main path agents, i.e., swaps involving $v_j \in \mathcal{I}\left(v_i\right)$ and/or $v_k \in \mathcal{I}\left(v_i\right)$. Consider the tree graph $\mathcal{T}_1 = (V, E_1)$ displayed in Fig. 3.7a. We note that

$$J_\mu(\mathcal{T}_1, \mathcal{R}^{1,5}) = \frac{1}{n}\left(\frac{35}{6} + \frac{5}{6}p + \frac{8}{6}q + \frac{p}{2}\left(p+1\right) + \frac{q}{2}\left(q+1\right)\right),$$

where p and q are the lengths of the paths connected to agent v_1 and v_2, respectively.

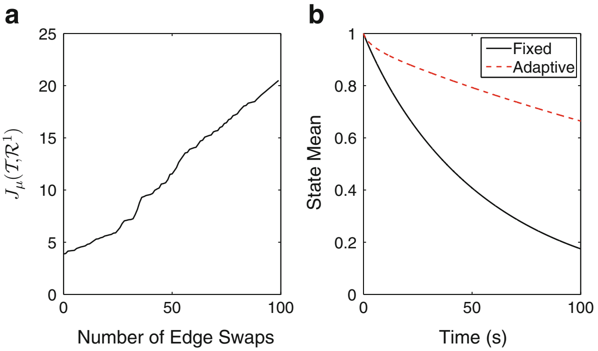
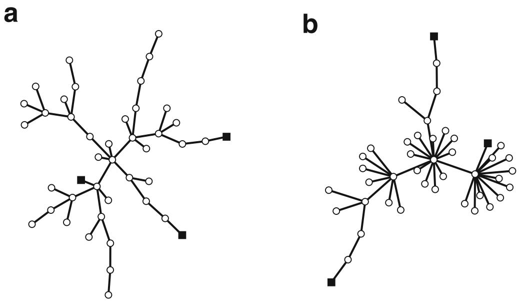


Fig. 3.5 (**a**) Mean tracking measure and (**b**) state mean for the fixed and adaptive tree graphs over time for the 40-agent random tree graph in Fig. 3.4 running Protocol 1

Fig. 3.6 (**a**) Original and (**b**) final tree graphs with three external agents attached (*squares*) after applying Protocol 2



Let us consider the potential edge swaps available to agent $v_2 \in \mathcal{M}$. Locally, agent v_2 is aware that $\mathcal{N}(v_2) = \{z_1, v_1, v_3\}$ and $\mathcal{I}(v_2) = \{v_1, v_3\}$. The potential edge swaps cases available are:

1. One neighbor on and one off the main path, e.g., swap $E_2 = E_1 - \{v_2, z_1\} + \{z_1, v_1\}$, forming $\mathcal{T}_2 = (V, E_2)$ and $E_3 = E_1 - \{v_2, z_1\} + \{z_1, v_3\}$ forming $\mathcal{T}_3 = (V, E_3)$.
2. Both neighbors on the main path, e.g., swap $E_4 = E_1 - \{v_2, v_3\} + \{v_3, v_1\}$ forming $\mathcal{T}_4 = (V, E_4)$.

Under Case 1, we have

$$J_\mu(\mathcal{T}_2, \mathcal{R}^{1,5}) = \frac{1}{n}\left(\frac{35}{6} + \frac{5}{6}p + \frac{9}{6}q + \frac{p}{2}(p+1) + \frac{q}{2}(q+1)\right)$$

$$J_\mu(\mathcal{T}_3, \mathcal{R}^{1,5}) = \frac{1}{n}\left(\frac{35}{6} + \frac{5}{6}p + \frac{5}{6}q + \frac{p}{2}(p+1) + \frac{q}{2}(q+1)\right)$$

$$J_\mu(\mathcal{T}_3, \mathcal{R}^{1,5}) < J_\mu(\mathcal{T}_1, \mathcal{R}^{1,5}) < J_\mu(\mathcal{T}_2, \mathcal{R}^{1,5}).$$

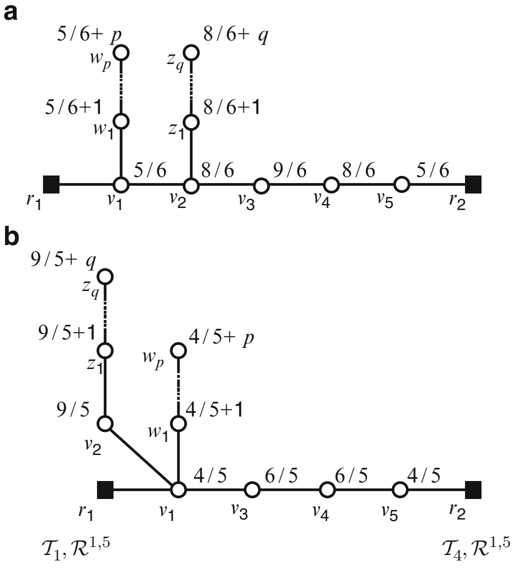


Fig. 3.7 Tree graphs, (**a**) $\mathcal{T}_1$ and (**b**) $\mathcal{T}_4$, with two attached external agents $\{r_1, r_2\}$. The effective resistance $E_{\text{eff}}(v_i)$ appears adjacent to each agent. The variables p and q are the lengths of the paths connected to agent v_1 and v_2, respectively

As under "local" information v_1 and v_3 are indiscernible, Case 1 does not guarantee that $J_\mu\,(\mathcal{T}, \mathcal{R})$ is increasing or decreasing.

In the meantime, under Case 2, we are led to graph $\mathcal{T}_4$ as displayed in Fig. 3.7b, with

$$J_\mu(\mathcal{T}_4, \mathcal{R}^{1,5}) = \frac{1}{n}\left(\frac{29}{5} + \frac{4}{5}p + \frac{9}{5}q + \frac{p}{2}\,(p+1) + \frac{q}{2}\,(q+1)\right)$$

$$J_\mu(\mathcal{T}_1, \mathcal{R}^{1,5}) - J_\mu(\mathcal{T}_4, \mathcal{R}^{1,5}) = \frac{1}{30n}\,(1 + p - 14q)\,.$$

Thus,

$$\begin{aligned} J_\mu(\mathcal{T}_1, \mathcal{R}^{1,5}) &> J_\mu(\mathcal{T}_4, \mathcal{R}^{1,5}) \quad \text{if } p > 14q - 1 \\ &\leq J_\mu(\mathcal{T}_4, \mathcal{R}^{1,5}) \quad \text{otherwise.} \end{aligned}$$

Under only "local" information, the relative magnitudes of p and q cannot be discerned so no monotonicity guarantees may be assumed.

A by-product of this remark is that a strictly increasing local-knowledge protocol cannot guarantee the tree graph with the largest $J_\mu\,(\mathcal{T}, \mathcal{R})$ (3.4) for $r > 1$ external agents.

3.3.3 *Example: Clock Synchronization*

Clock synchronization is often necessary in many distributed systems—improving the consistency of data and the correctness of algorithms. Precise time synchronization is needed for distributed applications such as sensor data fusion, scheduling, localization, coordinated actuation, and power-saving duty cycling. Motivated by the work of [8] we assembled the following experiment.

Consensus on clock time was run on 100 decentralized computer terminals (our agents) communicating over a tree network. Because time consensus can only correct for differential errors between terminals and not absolute errors without a reference, friendly external agents periodically connect to the network and deliver the constant correction for the absolute bias in the system. Upon connection, the friendly external agents initiate a friendly flag which is passed through the network, providing the local direction of the friendly agents and initiating Protocol 2. The network adapts under this protocol to promote convergence to the correct absolute clock time. On disconnection, the agents initiate a disconnect flag.

Similarly, we introduce a malicious external agent that attempts to drive the system to a false absolute time. Upon connection, the neighbors of the external agents send out a distress signal triggering the network to initiate Protocol 1 so as to deter the false convergence of the network. It is assumed that the friendly agents on discovery of a malicious external agent will clear the network of these foreign agents and trigger the termination of Protocol 1 before commencing delivery of the correction signal again. In other words, we assume friendly and malicious agents would not be concurrently connected to the network.

To examine the performance of the protocols, equal time was provided for both friendly and malicious external agents, specifically alternating 100 s intervals for 5,000 s. This switching interval is long enough for transients to settle and so is appropriate for the application of these protocols. The network was initialized as a random tree with all agents at the time offset of 0 s (the correct offset is -1 s). The set $\pi(\mathcal{E}_R)$ of agents connected to three external agents is randomly selected at each new 10 s interval. The friendly and unfriendly external agents deliver time offsets of -1 s and 1 s, respectively. The average of the constant values, i.e., 0 s, would be expected for the mean offset without the application of the protocols. In the meantime, the protocols are able to favor the friendly agent, bringing the average offset to -0.26 s. Clock offset means are displayed for the first 1,000 s for the fixed and adaptive trees in Fig. 3.8.

3.4 Variance Damping Measure

It can be the case that the mean is not of central interest and that adjustment of the variance of the states may be more desirable. Further, motivated by devious intrusion type techniques that may employ pulse-like control to avoid triangulation,

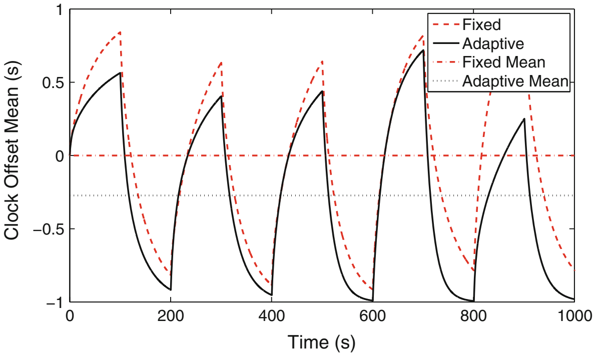


Fig. 3.8 Clock offset mean of the fixed and adaptive tree graphs (running Protocols 1 and 2) and the corresponding mean state

the energy of the states from a unit impulse input is another potentially desirable indicator for network performance and security. With this in mind, the controllability gramian, defined as $P =: \int_0^\infty e^{A\tau} BB^T e^{A^T\tau} d\tau$ for the system $\dot{x}(t) = Ax(t) + Bu(t)$, proves to be particularly suitable for such an analysis. We will focus on, $\mathbf{tr}(P)$ as

(a) the average variance of the agents' state in the network is

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left(z_i^2(t)\right) = \frac{1}{n}\mathbf{tr}\left(\mathbb{E}\left[z(t)z^T(t)\right]\right) = \frac{1}{n}\mathbf{tr}(P),$$

as $t \to \infty$, where $z(t) = x(t) - u(t)$ and u is a zero mean Gaussian with covariance I.

(b) the energy of the states at the output from a unit impulse input u when $x(0) = 0$ is

$$\int_0^\infty x(t)^T x(t)dt = \mathbf{tr}(P).$$

We note that P will be dependent on $\mathcal{G}$ and $\mathcal{R}$ and so henceforth is denoted by $P(\mathcal{G}, \mathcal{R})$. The *variance damping* measure is a metric quantifying the network's susceptibility to white noise from external agents.

Definition 3.12. The *variance damping* measure of a network is defined as[8]

$$J_\sigma(\mathcal{G}, \mathcal{R}) = \frac{2}{n}\mathbf{tr}(P(\mathcal{G}, \mathcal{R})). \tag{3.8}$$

[8] The scaling by $\frac{2}{n}$ is cosmetic.

The following section will provide more insights into the variance damping measure (3.8).

3.4.1 Analysis of Variance Damping Measure

Directly from the definition of the controllability gramian one has

$$\begin{aligned} J_\sigma(\mathcal{G},\mathcal{R}) &= \frac{2}{n}\mathbf{tr}(P(\mathcal{G},\mathcal{R})) \\ &= \frac{2}{n}\mathbf{tr}\left(\int_0^\infty e^{A(\mathcal{G},\mathcal{R})\tau} B(\mathcal{R}) B(\mathcal{R})^T e^{A(\mathcal{G},\mathcal{R})^T\tau} d\tau\right) \\ &= \frac{2}{n}\mathbf{tr}\left(M(\mathcal{R})\int_0^\infty e^{2A(\mathcal{G},\mathcal{R})\tau} d\tau\right) \\ &= -\frac{1}{n}\mathbf{tr}\left(M(\mathcal{R})\, A(\mathcal{G},\mathcal{R})^{-1}\right). \end{aligned} \tag{3.9}$$

Lemma 3.13 (General Variance Damping Measure). *For a connected graph $\mathcal{G}$, the variance damping measure is*

$$J_\sigma(\mathcal{G},\mathcal{R}) = \frac{1}{n}\sum_{v_i\in\pi(\mathcal{E}_R)} E_{\text{eff}}(v_i). \tag{3.10}$$

Proof. We note that $M(\mathcal{R})$ is a diagonal matrix with $[M(\mathcal{R})]_{ii} = 1$ if $v_i \in \pi(\mathcal{E}_R)$ and $[M(\mathcal{R})]_{ii} = 0$, otherwise. Therefore

$$\left[M(\mathcal{R})\, A(\mathcal{G},\mathcal{R})^{-1}\right]_{ii} = \begin{cases} \left[A(\mathcal{G},\mathcal{R})^{-1}\right]_{ii} & \text{if } v_i \in \pi(\mathcal{E}_R) \\ 0 & \text{otherwise.} \end{cases}$$

The statement of the lemma now follows. □

Corollary 3.14 (Single-External Variance Damping Measure). *For a connected graph and the influence model (3.2) with one external agent,*

$$J_\sigma\left(\mathcal{G},\mathcal{R}^i\right) = \frac{1}{n}.$$

Proof. The equivalent effective resistance between v_i and r_1 with $\{v_i\} = \pi(\mathcal{E}_R)$ is $E_{\text{eff}}(v_i) = 1$ as there is only one resistor link between v_i and r_1. The statement of the corollary now follows. □

Remark 3.15. The implication of Corollary 3.14 is that on average, a single-external agent attached to an n-agent connected graph has the same reduction in average variance to white noise and energy dissipation from an impulse input regardless of the structure of the network and where the external agent is attached.

Proposition 3.16 (Multiple-External Variance Damping Measure). *For connected graphs and the influence model (3.2) with r external agents, the variance damping measure is bounded below by a graph with all main path nodes satisfying $v_i \in \pi(\mathcal{E}_R)$ in which case*

$$J_\sigma(\mathcal{G}, \mathcal{R}) \geq \frac{r}{\sqrt{5}n}.$$

Proof. By Rayleigh's Monotonicity Principle[9] the minimum effective resistance will occur when the main path is only composed of the r agents $\pi(\mathcal{E}_R)$. Of these r agent graphs, the path graph with the most resistors in parallel will have the smallest effective resistance and therefore the smallest value of $J_\sigma(\mathcal{G}, \mathcal{R})$. The eigenvalues of the Laplacian of a r-node path graph are $\lambda_{r+1-i}(\mathcal{P}) = 2+2\cos\frac{\pi i}{r}$, for $i = 1, \ldots, r$ [26]. For $\tilde{\mathcal{R}}$ corresponding to an external agent attached to every agent in $\mathcal{P}$, from (3.3) and $M(\tilde{\mathcal{R}}) = I$ it follows that $\lambda_{r+1-i}(-A(\mathcal{P}, \tilde{\mathcal{R}})) = \lambda_{r+1-i}(\mathcal{P}) + 1$. Hence from (3.9), we conclude that

$$J_\sigma(\mathcal{G}, \mathcal{R}) \geq \frac{1}{n}\mathbf{tr}\left(-A(\mathcal{P}, \tilde{\mathcal{R}})^{-1}\right) = \frac{1}{n}\sum_{i=1}^{r}\frac{1}{3+2\cos\frac{\pi i}{r}} \geq \frac{r}{\sqrt{5}n}.$$

□

3.4.2 *Adaptive Protocols to Improve or Degrade the Variance Damping Measure for Trees*

We now propose another protocol for tree graphs with the objective of reducing the state variance due to external agents attached to the network and feeding in Gaussian white noise with covariance I, i.e., decreasing the variance damping measure (3.8). Again the protocol involves local edge trades executed concurrently and/or in a random agent order, guarantees that $\mathbf{tr}(P(\mathcal{T}, \mathcal{R}))$ decreases, and maintains a connected tree at each iteration. A complementary protocol to increase the variance damping measure is also proposed.

We note that for a connected tree graph $\mathcal{T}$, $J_\sigma(\mathcal{T}, \mathcal{R})$ is only dependent on $d(r_i, r_j)$ for all $\{r_i, r_j\}$ pairs in set R (as defined in Sect. 2.2), and so only dependent on the main path with agent set $\mathcal{M}$ (as defined in Sect. 3.3.1).

[9] Rayleigh's Monotonicity Law states that if the edge resistance in an electrical network is decreased, then the effective resistance between any two agents in the network can only decrease [1].

Protocol 3 Decreased variance damping measure edge swap

```
foreach Agent $v_i \notin \pi(\mathcal{E}_R)$ do
    if $\exists v_j, v_k \in \mathcal{N}(v_i)$, $v_j \neq v_k$ and $v_j, v_k = \mathcal{I}(v_i)$ then
        $E \to E - \{v_i, v_j\} + \{v_j, v_k\}$
    end
end
```

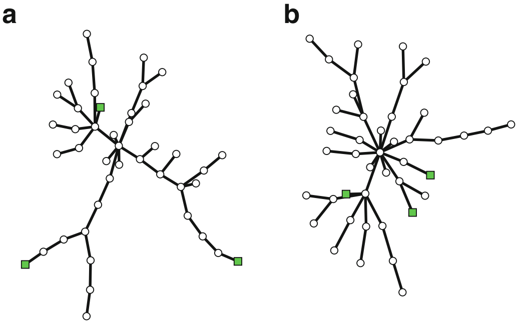


Fig. 3.9 (**a**) Original and (**b**) final tree graphs with three external agents attached (*squares*) after applying Protocol 3

Lemma 3.17 (Edge Swap for Decreased Variance Damping Measure). *Under Protocol 3, $J_\sigma(\mathcal{T}, \mathcal{R})$ (3.8) monotonically decreases.*

Proof. Firstly, when $|\mathcal{I}(v_i)| = 2$, $v_i \in \mathcal{M}$. As v_j and v_k are closer to an external agent than the main path agent v_i, one has $v_j, v_k \in \mathcal{M}$. The edge swap involves removing v_i from $\mathcal{M}$, so the effect is to reduce the resistance of an edge within the electrical network representing this subgraph. By Rayleigh's Monotonicity Law, the sum $\sum_{\{v_i, r_j\} \in \mathcal{E}_R} E_{\text{eff}}(v_i)$ will not increase and the lemma follows. □

Single-external agent trees will remain unaffected by Protocol 3. For double-external agent trees, the main path will degenerate to $\{v_i, v_j\} = \mathcal{M}$ where $\{\{v_i, r_1\}, \{v_j, r_2\}\} = \mathcal{E}_R$.

Protocol 3 was run on a 40-agent random tree with three external agents injecting zero mean white noise to the network. The original and final graphs, the variance damping measure and a sample output comparison between the fixed and adaptive networks (running Protocol 3) are displayed in Figs. 3.9 and 3.10.

A complementary energy amplification Protocol 4, that aims to increase $\mathbf{tr}(P(\mathcal{T}, \mathcal{R}))$, can also be obtained from Lemma 3.10. This protocol is suitable for impulse detection as larger $J_\sigma(\mathcal{T}, \mathcal{R})$ produces higher output energy $\int_0^\infty x(t)^T x(t) dt$.

Remark 3.18. For the case where $|\mathcal{I}(v_i)| > 2$, an edge swap has the effect of reducing v_i's degree and elongates the main path subgraph. Rayleigh's Monotonicity Law cannot be applied in this scenario as no "resistance" is being removed from the main path. Similar to Remark 3.11, these edge swaps do not guarantee $J_\sigma(\mathcal{T}, \mathcal{R})$ (3.8) is monotonically decreasing. Therefore, the proposed protocols are the best

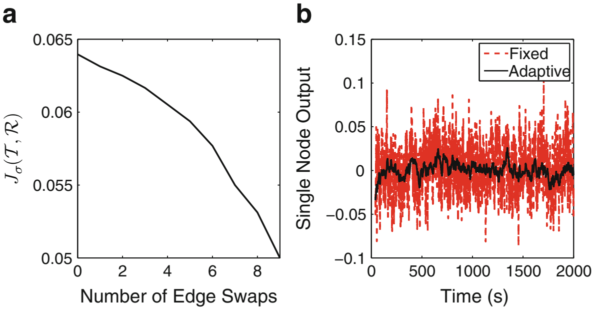


Fig. 3.10 (**a**) Variance damping measure and (**b**) one of the state node's output for the fixed and adaptive tree graphs over time for the 40-agent random tree graph in Fig. 3.9 exposed to three external agents running Protocol 3

Protocol 4 Increased variance damping measure edge swap

```
foreach Agent $v_i$ do
    if $|\mathcal{I}(v_i)| > 1$ and $\exists v_j, v_k \in \mathcal{N}(v_i)$, $v_j \in \mathcal{I}(v_i)$ and $v_k \notin \mathcal{I}(v_i)$ then
        $E \rightarrow E - \{v_i, v_j\} + \{v_j, v_k\}$
    end
end
```

"local" information edge swapping protocols and no guarantees can be made that the "local" information protocol will converge to the best "global" information edge swap solution.

3.4.3 Measure Comparison

We previously remarked that Protocols 1 and 2 do not alter the main path. Consequently, by Lemma 3.13, the quantity $J_\sigma(\mathcal{T},\mathcal{R})$ is conserved throughout these protocols so that, although the mean tracking measure is altered, the variance damping measure remains the same. The converse is not true as Protocols 3 and 4 involve manipulations of the main path and, as mentioned in Remark 3.11, this can arbitrarily vary $J_\mu(\mathcal{T},\mathcal{R})$ (3.4). Generally speaking as $J_\sigma(\mathcal{T},\mathcal{R})$ increases under Protocol 4 the graph elongates and so $J_\mu(\mathcal{T},\mathcal{R})$ tends to increase. Similarly, as $J_\sigma(\mathcal{T},\mathcal{R})$ decreases under Protocol 3 the graph compresses and so $J_\mu(\mathcal{T},\mathcal{R})$ tends to decrease. This trend is starkly apparent when the two metrics are requoted in terms of the effective resistance, i.e., by rearranging (3.5) and (3.10), we note that

$$J_\mu(\mathcal{G},\mathcal{R}) = J_\sigma(\mathcal{G},\mathcal{R}) + \frac{1}{n}\sum_{v_i \notin \pi(\mathcal{E}_R)} E_{\text{eff}}(v_i)\,. \tag{3.11}$$

We can requote the two metrics in terms of the error signal $z(t) = x(t) - u(t)$ and using the stochastic interpretation of $J_\sigma(\mathcal{G}, \mathcal{R})$ [27] as

$$J_\mu(\mathcal{G}, \mathcal{R}) = \mathbb{E}_{\|\tilde{x}(0)\|=1}\left(2\int_0^\infty \mathbb{E}(z(t))^T \mathbb{E}(z(t))\,dt\right), \text{ and}$$

$$J_\sigma(\mathcal{G}, \mathcal{R}) = \mathbb{E}\left(\lim_{T\to\infty}\frac{1}{nT}\int_{-T}^{T} z(t)^T z(t)dt\right).$$

The two metrics can also be interpreted to characterize different components of the output signal. The metric $J_\mu(\mathcal{G}, \mathcal{R})$ with respect to the mean is mainly influenced by the initial deviations of $z(t)$, or in other words, the *transient* response. On the other hand, $J_\sigma(\mathcal{G}, \mathcal{R})$ with respect to the variance is more sensitive to long-term fluctuations or *steady state* response.

The adaptive Protocols 1–4 and a subset of our results are specific to tree graphs. A preliminary extension to more general connected graphs can be made by considering an arbitrary spanning tree $\mathcal{T}$ of a connected graph $\mathcal{G}$. In terms of our electrical resistance analogy, the resistor network $\mathcal{T}$ is formed by removing resistors from $\mathcal{G}$. Applying Rayleigh's Monotonicity Principle leads to $J_\mu(\mathcal{G}, \mathcal{R}) \leq J_\mu(\mathcal{T}, \mathcal{R})$ and $J_\sigma(\mathcal{G}, \mathcal{R}) \leq J_\sigma(\mathcal{T}, \mathcal{R})$, i.e., both metrics on the graph are bounded above by the corresponding measures on its spanning trees.

3.5 Fusing Adaptive Protocols

Protocols 1–4 in Sects. 3.3 and 3.4 possess guarantees on increasing (or decreasing) either the mean tracking or variance damping measures of the network. The weakness of these protocols is that they tend to converge to graphs associated with a local minimum (or maximum) of either $J_\mu(\mathcal{T}, \mathcal{R})$ or $J_\sigma(\mathcal{T}, \mathcal{R})$, with potentially sub-optimal performance. Furthermore the protocols cannot be applied concurrently, e.g., for security applications where poor tracking of the mean (high $J_\mu(\mathcal{T}, \mathcal{R})$) and good noise damping (low $J_\sigma(\mathcal{T}, \mathcal{R})$) is favorable. We now present a protocol that exhibits these attributes, i.e., the final graphs are within guaranteed bounds of the optimal network over all graphs for maximizing $J_\mu(\mathcal{T}, \mathcal{R})$ and minimizing $J_\sigma(\mathcal{T}, \mathcal{R})$, but the protocol no longer possess strictly increasing $J_\mu(\mathcal{T}, \mathcal{R})$ and decreasing $J_\sigma(\mathcal{T}, \mathcal{R})$. We will present the protocol and use a game-theoretic formalism to bound the protocol's performance.

In the following, our game-theoretic objective is to increase $J_\mu(\mathcal{T}, \mathcal{R})$ and decrease $J_\sigma(\mathcal{T}, \mathcal{R})$; in terms of effective resistance, the aim is to increase the final term in (3.11) while keeping $J_\sigma(\mathcal{G}, \mathcal{R})$ small. This produces a graph that both damps the external agents' effect on the system's state mean and variance.

The proposed Protocol 5 concurrently applies Protocols 1 and 3 with a slight adaption to the latter, specifically, relaxing the condition $\{v_j, v_k\} = \mathcal{I}(v_i)$ to $v_j, v_k \in \mathcal{I}(v_i)$. This adaption guarantees that the main path subgraph will converge

Protocol 5 Increased mean resilience and decreased variance resilience edge swap

```
foreach Agent $v_i$ do
    if $\exists v_j, v_k \in \mathcal{N}(v_i)$ and $v_j \neq v_k$, and $(v_j, v_k \notin \mathcal{I}(v_i))$ or $(v_i \notin \pi_2(\mathcal{E}_R)$ and $v_j, v_k \in \mathcal{I}(v_i))$
    then
        $E \rightarrow E - \{v_i, v_j\} + \{v_j, v_k\}$
    end
end
```

to a graph of only native agents where the external agents *directly* attach, i.e., $v_i \in \pi\ (\mathcal{E}_R)$. The remaining nodes in the graph, in the meantime, will form paths connected to an agent in $\pi\ (\mathcal{E}_R)$. There are many graphs and external agents pairs $(\mathcal{T}, \mathcal{R})$ that satisfy these properties; we call the set of such pairs the *acquirable* set $\mathcal{A}$. In fact, the specific "equilibrium" that Protocol 5 converges to will depend on the initial graph structure and the sequence of edge swaps prescribed by the protocol. It turns out that the convergence of the protocol falls under a special class of repeated games called *potential game*s [28], and as such, exhibits certain sub-optimality guarantees that will be explored further.

3.5.1 Game-Theoretic Analysis

Game theory supplies tools to assess the optimality properties of equilibria reached following local decisions. Two metrics are generally used for this purpose; the price of stability which is the ratio between the "best" equilibria obtained from local decisions and the global optimum, and the price of anarchy which is the ratio of the "worst" equilibria obtained from local decisions and the global optimum. For our case these metrics will capture the success of our local protocol with respect to the mean tracking and variance damping measures.

First we need to establish that the protocol indeed converges to some equilibrium; for this task, we use the concept of a potential game. A potential function Φ is a function that maps a strategy vector (a vector of each agent's edge swap) $S = (S_1, S_2, \ldots, S_n)$ to some real valued number. The implementation of a strategy on graph $\mathcal{T}$ will alter it to produce a graph $\mathcal{T}(S)$. If a protocol is a potential game then: if $S_i' \neq S_i$ is an alternate strategy (edge swap) for agent i, then the local cost benefit to the agent $u_i(S') - u_i(S)$ will mirror the change in the potential, i.e., condition[10]

$$\operatorname{sgn}(\Phi(S) - \Phi(S')) = \operatorname{sgn}(u_i(S') - u_i(S)). \tag{3.12}$$

[10] The signum function is represented by sgn(·).

Consider the potential function

$$\Phi\left(\mathcal{T}(S), \mathcal{R}\right) = -\sum_{i=1}^{n} d\left(v_i, \Gamma\left(v_i\right)\right),$$

where $\Gamma\left(v_i\right)$ is defined in Sect. 3.3.1. Therefore if the local cost of agent v_i is

$$u_i(\mathcal{T}(S), \mathcal{R}) = d\left(v_i, \Gamma\left(v_i\right)\right), \tag{3.13}$$

then (3.12) is met. Since Protocol 5 satisfies (3.13), it can be considered as a potential game.[11] An important consequence of this observation is that Protocol 5 will always converge to an equilibrium [28].

We can now find the price of stability and anarchy with respect to the maximization of measures $J_\mu\left(\mathcal{T}, \mathcal{R}\right)$ and $1/J_\sigma\left(\mathcal{T}, \mathcal{R}\right)$ under Protocol 5.

Proposition 3.19. *Under Protocol 5, for $J_\mu\left(\mathcal{T}, \mathcal{R}\right)$ the price of stability is equal to 1 and the price of anarchy is less than or equal to r.*

Proof. As the graph corresponding to the smallest $J_\mu\left(\mathcal{T}, \mathcal{R}\right)$ (3.4) is in the acquirable set $\mathcal{A}$ (by Proposition 3.9), the price of stability is equal to 1.

From Proposition 3.9 the maximum $J_\mu\left(\mathcal{T}, \mathcal{R}\right)$ is bounded as

$$\max_{(\mathcal{T},\mathcal{R})} J_\mu\left(\mathcal{T}, \mathcal{R}\right) \leq \frac{1}{2n}\left((n-r)^2 + 3\left(n-r\right) + r + \frac{2}{r+1}\right).$$

An acquirable graph with the smallest $J_\mu\left(\mathcal{T}, \mathcal{R}\right)$ corresponds to a network with the main path subgraph as a path $\mathcal{P}$ (by Proposition 3.16) with

$$\min_{(\mathcal{T},\mathcal{R}), v_j \in \mathcal{M}} E_{\text{eff}}\left(v_j\right) \geq \frac{1}{\sqrt{5}}.$$

The equilibrium graph in $\mathcal{A}$ corresponding to the smallest $J_\mu\left(\mathcal{T}, \mathcal{R}\right)$ compared to the tree from $\text{argmax}_{(\mathcal{T},\mathcal{R})} J_\mu\left(\mathcal{T}, \mathcal{R}\right)$ will have $\left\lfloor \frac{n-r}{r} \right\rfloor = \frac{n-r}{r}$ agents attached as a path to each of the main path agents.[12] Applying Lemma 3.6 leads to the inequality

$$\begin{aligned}\min_{(\mathcal{T},\mathcal{R}) \subseteq \mathcal{A}} J_\mu\left(\mathcal{T}, \mathcal{R}\right) &\geq \frac{1}{n}\left[\frac{r}{\sqrt{5}} + r\sum_{i=1}^{(n-r)/r}\left(\frac{1}{\sqrt{5}} + i\right)\right] \\ &= \frac{1}{2nr}\left((n-r)^2 + \left(\frac{2}{\sqrt{5}} + 1\right) r\left(n-r\right) + \frac{2}{\sqrt{5}} r^2\right).\end{aligned}$$

[11] This approach is similar to other network game problems [6, 28].

[12] $\lfloor x \rfloor$ is defined as the 'floor' of x.

For $r = 1$, the protocol always acquires the optimal equilibrium of a path graph with an external agent connected to one end node so for this case the price of anarchy is equal to 1. For $1 < r \leq n$, on the other hand, the

$$\begin{aligned}\text{Price of anarchy} &= \frac{\max_{(\mathcal{T},\mathcal{R})} J_\mu\left(\mathcal{T},\mathcal{R}\right)}{\min_{(\mathcal{T},\mathcal{R})\subseteq\mathcal{A}} J_\mu\left(\mathcal{T},\mathcal{R}\right)} \\ &\leq r\frac{\left(n-r\right)^2 + 3\left(n-r\right) + r + \frac{2}{r+1}}{\left(n-r\right)^2 + \left(\frac{2}{\sqrt{5}}+1\right) r\left(n-r\right) + \frac{2}{\sqrt{5}}r^2} < r,\end{aligned}$$

thus proving the proposition. □

Proposition 3.20. *Under Protocol 5, for $J_\sigma\left(\mathcal{T},\mathcal{R}\right)$ (3.8) the price of stability is equal to* 1 *and the price of anarchy is less than* $\frac{11\sqrt{5}}{20} \approx 1.23$.

Proof. As the graph corresponding to the maximum $J_\sigma\left(\mathcal{T},\mathcal{R}\right)$ is in $\mathcal{A}$ (by Proposition 3.16), the price of stability is equal to 1.

From Proposition 3.16, the optimal $J_\sigma\left(\mathcal{T},\mathcal{R}\right)$ equilibrium corresponds to a network with the main path subgraph as a path $\mathcal{P}$ with

$$\min_{(\mathcal{T},\mathcal{R})} J_\sigma\left(\mathcal{T},\mathcal{R}\right) \geq \frac{r}{\sqrt{5}n}.$$

From (3.6), the largest $J_\sigma\left(\mathcal{T},\mathcal{R}\right)$ equilibrium graph in $\mathcal{A}$ is associated with main path subgraph as a star $\mathcal{S}$ with

$$\max_{(\mathcal{T},\mathcal{R})\subseteq\mathcal{A}} J_\sigma\left(\mathcal{T},\mathcal{R}\right) = \frac{r^2+r+2}{2n(r+1)}.$$

For $r = 1, 2, 3$, the protocol always acquires the optimal equilibrium corresponding to main path subgraph as a path $\mathcal{P}$ so for this case the price of anarchy is equal to 1. For $3 < r \leq n$, on the other hand, the

$$\begin{aligned}\text{Price of anarchy} &= \frac{\max_{(\mathcal{T},\mathcal{R})\subseteq\mathcal{A}} J_\sigma\left(\mathcal{T},\mathcal{R}\right)}{\min_{(\mathcal{T},\mathcal{R})} J_\sigma\left(\mathcal{T},\mathcal{R}\right)} \\ &< \frac{\sqrt{5}}{2}\frac{r^2+r+2}{r^2+r} < \frac{11\sqrt{5}}{20},\end{aligned}$$

thus proving the proposition statement. □

Protocol 5 was applied to a 40-node tree graph with seven external agents attached. For comparison, Protocol 1 (increasing mean tracking measure) and Protocol 3 (decreasing variance damping measure) were applied to the same graph. The original and final graphs for each protocol appear in Fig. 3.11 while the metrics $J_\mu\left(\mathcal{T},\mathcal{R}\right)$ and $J_\sigma\left(\mathcal{T},\mathcal{R}\right)$ for each as compared with the optimal tree graphs for $J_\mu\left(\mathcal{T},\mathcal{R}\right)$ and $J_\sigma\left(\mathcal{T},\mathcal{R}\right)$ are displayed in Fig. 3.12.

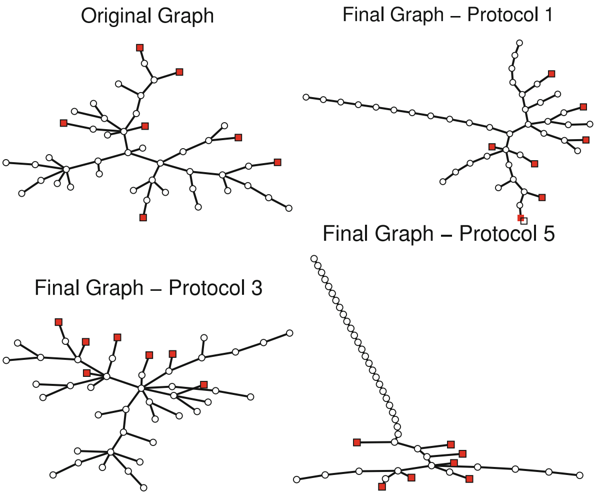


Fig. 3.11 Original and final tree graphs with seven external agents attached (*squares*) after applying Protocols 1, 3, and 5

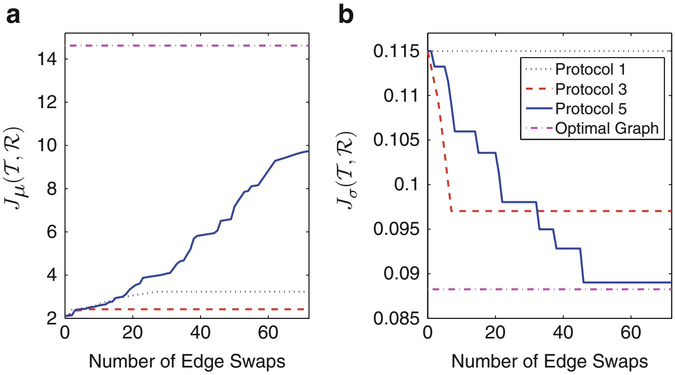


Fig. 3.12 (**a**) $J_\mu(\mathcal{T},\mathcal{R})$ and (**b**) $J_\sigma(\mathcal{T},\mathcal{R})$ after each edge swap from Protocols 1, 3, and 5 applied to the original graph in Fig. 3.11 as well as the optimal tree graphs with 40 nodes and seven external agents

We note that Protocol 5 outperformed Protocols 1 and 3. The ratio of the optimal to the final equilibrium under Protocol 5 was less than 1.51 for $J_\mu(\mathcal{T},\mathcal{R})$ and 1.08 for $1/J_\sigma(\mathcal{T},\mathcal{R})$, agreeing with the game-theoretic bounds stated in Propositions 3.19 and 3.20.

3.6 Remarks

The main aim of the present work is to propose a system-theoretic approach to examine the notion of semi-autonomy. In particular, the chapter presents a class of consensus-type networks under the influence of external agents. Metrics were introduced to quantify the network's ability, via its topology, to promote or resist the influence of external agents. Four decentralized protocols were proposed for tree graphs to vary the mean tracking and variance damping measures within the network. The protocols were applied to time synchronization and UAV flocking applications.

The proposed metrics were then analyzed and an effective resistance analogy was established by modeling the interconnection as a resistive network. The resistance interpretation provided a method to compare the two metrics and illustrated their relationship. The challenge of presenting a protocol that increased one metric while decreasing the other was addressed for tree graphs with a hybrid protocol and analyzed using a game-theoretic approach. The extension of these protocols to more general networks will be discussed in a subsequent chapter. Finally, an area for future research is the performance and security of networks exploiting *both* topological and agent dynamic features of the network.

References

1. B. Bollobás, *Modern Graph Theory*. New York: Springer, 1998.
2. R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.
3. H. G. Tanner, G. J. Pappas, and V. Kumar, "Leader-to-formation stability," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 443–455, 2004.
4. A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.
5. Y. Hatano and M. Mesbahi, "Agreement over random networks," *IEEE Transactions on Automatic Control*, vol. 50, no. 11, pp. 1867–1872, 2005.
6. M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton: Princeton University Press, 2010.
7. R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York: Cambridge University Press, 1990.
8. S. Graham and P. R. Kumar, "Time in general-purpose control systems: the control time protocol and an experimental evaluation," in *Proc. 43rd IEEE Conference on Decision and Control*, 2004, pp. 4004–4009.
9. A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt, "Controllability of multi-agent systems from a graph-theoretic perspective," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 162–186, 2009.
10. S. Sundaram and C. N. Hadjicostis, "Distributed function calculation via linear iterations in the presence of malicious agents - part I: attacking the network," in *Proc. American Control Conference*, 2008, pp. 1350–1355.
11. R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

12. D. Zelazo and M. Mesbahi, “Edge agreement: graph-theoretic performance bounds and passivity analysis,” *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 544–555, Mar. 2011.
13. A. Ghosh and S. Boyd, “Growing well-connected graphs,” in *Proc. 45th IEEE Conference on Decision and Control*. IEEE, 2006, pp. 6605–6611.
14. Y. Wan, S. Roy, and A. Saberi, “Network design problems for controlling virus spread,” in *Proc. 46th IEEE Conference on Decision and Control*, 2007, pp. 3925–3932.
15. A. Fagiolini, G. Valenti, L. Pallottino, G. Dini, and A. Bicchi, “Decentralized intrusion detection for secure cooperative multi-agent systems,” in *Proc. 46th IEEE Conference on Decision and Control*, 2007, pp. 1553–1558.
16. F. Pasqualetti, A. Bicchi, and F. Bullo, “Distributed intrusion detection for secure consensus computations,” in *Proc. 46th IEEE Conference on Decision and Control*, 2007, pp. 5594–5599.
17. A. Giridhar and P. R. Kumar, “Distributed clock synchronization over wireless networks: algorithms and analysis,” in *Proc. 45th IEEE Conference on Decision and Control*, 2006, pp. 4915–4920.
18. F. Pasqualetti, A. Bicchi, and F. Bullo, “On the security of linear consensus networks,” in *Proc. 48th IEEE Conference on Decision and Control*, 2009, pp. 4894–4901.
19. P. Barooah and J. P. Hespanha, “Graph effective resistance and distributed control: spectral properties and applications,” in *Proc. 45th IEEE Conference on Decision and Control*, 2006, pp. 3479–3485.
20. M. Mavronicolas, V. G. Papadopoulou, A. Philippou, and P. G. Spirakis, “A graph-theoretic network security game,” *International Journal of Autonomous and Adaptive Communications Systems*, vol. 1, no. 4, p. 390, 2008.
21. S. D. Antonio, S. P. Romano, S. Simpson, and P. Smith, “A semi-autonomic framework for intrusion tolerance in heterogeneous networks,” in *Proc. 3rd International Workshop on Self-Organizing Systems*, 2008, pp. 230–241.
22. D. S. Callaway, M. E. J. Newman, S. H. Strogatz, and D. J. Watts, “Network robustness and fragility: percolation on random graphs,” *Phys. Rev. Lett.*, vol. 85, no. 25, p. 4, Jul. 2000.
23. G. Tyson, A. T. Lindsay, S. Simpson, and D. Hutchison, “Improving wireless sensor network resilience with the INTERSECTION framework,” in *Proc. 2nd International Conference on Mobile Lightweight Wireless Systems, Critical Information Infrastructure Protection*, 2010.
24. S. Sundaram and C. N. Hadjicostis, “Distributed function calculation via linear iterations in thepresence of malicious agents - part II: overcoming malicious behavior,” in *Proc. American Control Conference*, 2008, pp. 1356–1361.
25. S. Salsa, *Partial Differential Equations in Action: From Modelling to Theory*. New York: Springer, 2008.
26. M. Petrovic and I. Gutman, “The path is the tree with smallest greatest Laplacian eigenvalue,” *Kragujevac Journal of Mathematics*, vol. 24, pp. 67–70, 2002.
27. S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*. West Sussex: Wiley, 2005.
28. N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, *Algorithmic Game Theory*. New York: Cambridge University Press, 2007.
29. A. Chapman, E. Schoof, and M. Mesbahi, “Semi-autonomous networks: theory and decentralized protocols,” in *Proc. of the IEEE International Conference on Robotics and Automation*, 2010, pp. 1958–1963.
30. A. Chapman, M. Nabi-Abdolyousefi, and M. Mesbahi, “Identification and infiltration in consensus-type networks,” in *1st IFAC Workshop on Estimation and Control of Networked Systems*, 2009, pp. 84–89.

Chapter 4
Distributed Online Topology Design for Disturbance Rejection

Abstract In this chapter, we examine a networked multi-agent system running consensus susceptible to mis-information from its environment. The influenced dynamics are modeled with leader–follower dynamics and the impact of the foreign input is measures through the open loop $\mathcal{H}_2$ norm of the network dynamics. To dampen the external disturbances a novel decentralized edge reweighting method is proposed. The method is composed of a decentralized conjugate gradient method coupled with a decentralized online optimization algorithm. The uncertainties of the effect of local rewiring and unknown environmental influences are demonstrated to be well-suited to the online regret framework. A simulation of the reweighting method is discussed and shown to have small regret.

4.1 Introduction

Control theory presents many mechanisms to reject or dampen disturbances in a dynamic system. In general they fall in two categories: active control such as dynamic feedback, and passive control such as structural damping. Typically in networked dynamics systems the network structure is considered a passive element. If feedback is unavailable or global network knowledge is insufficient to apply feedback, adapting the topology may in fact act as an active system via dynamic selection of interconnection (edge) weights. Further this is often a favorable solution when global information is limited and decisions are tentative. Edge reweighting is a conservative response with algorithms like consensus able to perform in the presence of variations in edge weight.

We consider a scenario where an agent i can communicate with any other nearby agents $j \neq i$. This communication does not implicitly involve knowing the location of the peer agents. In the course of a mission, the agent might mistake an object in the environment for a peer agent. The agent would know there is a discrepancy in the data since it knows how many peers it has, and how many objects it can see. It might want to limit the spread of misinformation via network reweighting since its removal might render the network unconnected, especially if other agents are also making the same decisions.

To model such a scenario we consider the agent's false positives as foreign input signals and use the leader–follower dynamics [2] to model the system. The main

A. Chapman, *Semi-Autonomous Networks*, Springer Theses,
DOI 10.1007/978-3-319-15010-9_4

tool for investigating the susceptibility of these influenced subset of agents is the open loop $\mathcal{H}_2$ norm of these leader–follower dynamics. In particular, the open loop $\mathcal{H}_2$ norm for the network can be employed as a measure to dynamically reweight the interaction topology, reducing the effect of the foreign inputs into the network. Increasing the open loop $\mathcal{H}_2$ norm tends to increase the receptiveness of the system to control.

Unique to the aforementioned methods we propose a fully *distributed* reweighting method composed of two stages.

The first stage is the distributed estimation of the local effect of the foreign signals, achieved via the formulation of a distributed conjugate gradient method. In the 1950s, Hestens and Stiefel [5] formulated the linear conjugate gradient method as an iterative method to solve $A^{-1}b$. An attraction of this method is theoretical guaranteed convergence in less than n iterations, and with typical performance even faster. In addition, it boasts small storage requirements. In fact, only samples of the range space of A are required rather than knowledge of the complete A. We show that, if A encodes the network structure, this feature makes it possible to form a distributed version of the algorithm with the aid of two consensus updates per timestep.

The second stage is the implementation of a distributed online algorithm to reweight the network. The attraction of the online method is that, irrespective of the changing in the network unseen by a single agent, certain performance guarantees can still be made. The proposed online regret algorithm is a distributed version of a centralized algorithm proposed by Hazan et al. [6]. The algorithm is similar to gradient decent or incremental gradient methods surveyed by Bertsekas [7]. The attraction of the proposed algorithm is it displays $O(\log(T))$ regret, meaning that on average the algorithm performs as well as the best fixed case solution in hindsight. Related online distributed optimization is the work by Yan et al. [8] examining a strongly convex cost function decomposable into smaller convex functions. Sundhar et al. [9] explored a stochastic subgradient showing convergence to an optimal solution with probability 1. Recently, Raginsky et al. [10] focused on a network of agents performing a subgradient-based sequential convex optimization scheme robust to the network structure.

The novelty of our distributed online formulation is that, unlike the aforementioned distributed algorithms, our cost function is not decomposable. We derive a method to instead decompose the gradient of the $\mathcal{H}_2$ norm with respect to the edges. Consequently, it is this formulation that allows our algorithm to be decentralized.

The chapter is organized as follows. Section 4.2 contains the relevant background pertaining to online optimization. Weighted consensus-based, leader–follower dynamics is described for the network dynamics in Sect. 4.3 as well as the main dynamics performance measure the open loop $\mathcal{H}_2$ norm. The distributed conjugate gradient method is then presented in Sect. 4.4 as well as its counterpart the distributed online gradient descent method. The proposed method is applied and the performance of the algorithm is examined in the regret framework. The chapter is concluded with a few remarks in Sect. 4.5.

4.2 Online Convex Optimization

Online convex optimization is formulated as a game where, at each time t, a player selects a point x_t in a convex set $\mathcal{P}$, referred to as an action. After the player has committed to the action a convex function $f_t(\cdot)$ is revealed to the player at which point a penalty of $f_t(x_t)$ is paid by the player. The objective of this game is to minimize the accumulative penalty. *Regret* is a common measure for the performance of this highly uncertain online system. Regret is defined as the difference between the cost of the sequence of actions taken by the player and the performance of the best single action x^* taken at every time step if the sequence $\{f_t(\cdot)\}$ is known *a priori*. Hence, the regret of an algorithm with action sequence $\{x_t\}$ is

$$\mathcal{R}_T = \sum_{t=1}^{T} \left(f_t(x_t) - f_t(x^*) \right).$$

The objective of a good online algorithm is to achieve a guaranteed low regret. Specifically, one that guarantees sublinear $\mathcal{R}_T$ or $\mathcal{R}_T/T \rightarrow 0$. The reasoning is that when $\mathcal{R}_T/T \rightarrow 0$, "on average" the algorithm performs as well as the best fixed action in hindsight.

4.3 Model and Measure

4.3.1 Weighted Leader–Follower Consensus Dynamics

As discussed in the last chapter, one of the advantages of the consensus dynamics is that the additional agents can seamlessly integrate into the dynamics by entering the communication range of the network. The detriment is that incorrect identification of an agent adds an unwanted signal into the network. Subsequently, an agent incorrectly "follows" this unwanted signal. The dynamics governing this foreign signal takes the form of the popular leader–follower consensus dynamics [2], where the incorrectly identified agent plays the role of a leader, and the native agents in the network as followers.

To form the *weighted* leader–follower consensus dynamics, the network graph $\mathcal{G} = (V, E, W)$ is extended to incorporate the foreign agents/signals into the graph similarly to the unweighted case of Chap. 3. This is accomplished by considering the foreign agent pair $\mathcal{R} = (R, \mathcal{E}_R)$, where R is the r element foreign node set and $\mathcal{E}_R \subseteq R \times V$ is the set of false edges attached to the network. It is assumed that a foreign agent $r_j \in R$ is mis-observed at position $u_j(t) \in \mathbb{R}$ by only one native agent, and one native agent mistakes at most one foreign agent at a time. Thus, there is exactly one edge $\mathcal{E}_R$ for each agent in R and no more than one $\mathcal{E}_R$ for each V.

The resulting weighted leader–follower consensus system now assumes the form,

$$\dot{x}_i(t) = \sum_{\{v_i,v_j\}\in E} w_{ij}\left(x_j(t) - x_i(t)\right) + \sum_{\{v_i,r_j\}\in\mathcal{E}_R} \left(u_j(t) - x_i(t)\right) \tag{4.1}$$

with the full dynamics

$$\dot{x}(t) = A(\mathcal{G},\mathcal{R})x(t) + B(\mathcal{R})u(t), \tag{4.2}$$

where, as before, $B(\mathcal{R}) \in \mathbb{R}^{n\times r}$ with $[B(\mathcal{R})]_{ij} = 1$ when $\{r_j, v_i\} \in \mathcal{E}_R$ and $[B(\mathcal{R})]_{ij} = 0$ otherwise, and

$$A(\mathcal{G},\mathcal{R}) = -\left(\mathscr{L}(\mathcal{G}) + \sum_{i\in R} e_i e_i^T\right) \in \mathbb{R}^{n\times n}. \tag{4.3}$$

4.3.2 Disturbance Rejection Using the Open Loop $\mathcal{H}_2$ Norm

A measure of the effect that an input, represented by the matrix B, on a state dynamics $\dot{x}(t) = Ax(t) + Bu(t)$ is the open loop $\mathcal{H}_2$ norm of the system $\|G(s)\|_2$ where the full-state output state-space realization is $G(s) = (sI - A)^{-1} B$. The metric represents the amplification of the mapping of inputs to the full-state outputs, i.e., $y(s) = x(s) = G(s)u(s)$. More precisely, $y\,(s)$ is the energy of the system at the states from a unit impulse input $u(t)$ when $x(0) = 0$. Consequently, in general, decreasing $\|G(s)\|_2$ has the effect of dampening disturbances in the system inputs though the matrix B.

A convenient method of representing the $\|G(s)\|_2$ is using the trace of the controllability gramian defined as $P(A, B) := \int_0^\infty e^{A\tau}BB^T e^{A^T\tau}d\tau$. From this relationship, when A is invertible and symmetric,

$$\begin{aligned}
\|G(s)\|_2^2 &= \mathbf{tr}\left(P\,(A, B)\right)\\
&= \mathbf{tr}\left(\int_0^\infty e^{A\tau}BB^T e^{A^T\tau}d\tau\right)\\
&= \mathbf{tr}\left(\int_0^\infty BB^T e^{A^T\tau}e^{A\tau}d\tau\right)\\
&= \mathbf{tr}\left(BB^T \int_0^\infty e^{2A\tau}d\tau\right)\\
&= -\frac{1}{2}\mathbf{tr}\left(B^T A^{-1} B\right).
\end{aligned}$$

From these observations, in general, inputs perturb the outputs more effectively as $\|G(s)\|_2$ increases. In other words, systems with a larger open loop $\mathcal{H}_2$ norm are more easily perturbed. With this in mind, the focus of this work is to minimize $\|G(s)\|_2^2$.

The $\mathcal{H}_2$ norm of the system also has attractive graph-theoretic properties linked through the concept of the effective resistance of the graph discussed in Chap. 3. A consequence of this connection is that the weights on the edges can be considered as conductance in an equivalent electrical network realization of the graph [4]. Since the effective resistance always decreases with increasing conductance, $\|G(s)\|_2^2$ will decrease or remain constant with any increase in edge weights. This useful property leads to the following section on the reweighting of edges to decrease $\|G(s)\|_2$.

4.4 Distributed Online Topology Design Algorithm

As many networked systems, such as UAV swarms, require only non-physical interconnections for their coordinated behavior, they have the advantage that their inter-vehicle coordination graph can be *reweighted*. This observation leads to an online method for improving network manageability, namely via a judicious topology reweighting. The goal is to adapt the network topology distributively and with only minimum local knowledge of the network topology so as to improve disturbance rejection. The metric for "good rejection" used in the following analysis is the open loop $\mathcal{H}_2$ norm of the network dynamics, described in Sect. 4.3.2.

The challenge of dynamic distributed reweighting is that agents are unable to coordinate with non-neighboring agents. Consequently, local edge reweightings in light of other agent's reweights can be detrimental. Furthermore, by the time information pertaining to the topological effects of a foreign agent has been received, the foreign agent may no longer be present. Therefore, the online regret framework is ideal to address this problem. The changing dynamic structure manifests itself as a time-varying graph $\mathcal{G}_t$ with a corresponding time varying $\mathscr{L}(\mathcal{G}_t)$. In turn, varying foreign agents appear as a changing foreign agent pair $\mathcal{R}_t$. Therefore the state model (3.2) is

$$\dot{x}(t) = A_t x(t) + B_t u(t),$$

where $A_t = A(\mathcal{G}_t, \mathcal{R}_t)$ and $B_t = B(\mathcal{R}_t)$. The weights A_t can be represented as

$$A_t = -\sum_{(i,j)\in E} w_{ij} a_{ij} a_{ij}^T - \sum_{i\in R} e_i e_i^T, \tag{4.4}$$

which is invertible for all positive weights so long as R is nonempty and $\mathcal{G} = (V, E, I)$ is connected.

Thus, our metric of interest becomes $\|G(s)\|_2^2 = -\frac{1}{2}\mathbf{tr}\left(B_t^T A_t^{-1} B_t\right)$. The unconstrained minimization of $\|G(s)\|_2$ will increase the weights on the network arbitrarily. In addition, large edge weights can have adverse effects on the network responsiveness. Consequently, we consider the following minimization so as to balance good rejection with the penalization of overly large weights,

$$f_t(W) = -\frac{1}{2}\mathbf{tr}\left(B_t^T A_t^{-1} B_t\right) + \frac{1}{2}h\mathbf{1}^T W^T W\mathbf{1},$$

where constant $h > 0$. The arbitrary removal of weights is also unwanted as the graph can be rendered disconnected. Therefore no edges can be reduced below some positive vector $q_{\sigma(ij)}$, i.e., $w_{ij} \geq q_{\sigma(ij)}$. Additional constraints relating to maximum edge deviations and node degree constraints can also be considered. These constraints are specified via vectors $u_{\sigma(ij)} \in \mathbb{R}^m$ and $s \in \mathbb{R}^n$ such that $w_{ij} \leq u_{\sigma(ij)}$ and $\sum_{j\in N(i)} w_{ij} \leq s_i$.

$$\begin{aligned} &\min f_t(W) \\ &\text{s.t.} \quad q_{\sigma(ij)} \leq w_{ij} \leq u_{\sigma(ij)} \\ &\qquad \sum_{j\in N(i)} w_{ij} \leq s_i. \end{aligned} \tag{4.5}$$

A convex constraint set denoted by $\mathcal{P}$ is defined via these linear constraints. The analysis of the derivative and hessian of the metric $\|G(s)\|_2^2$ with respect to the edge weights is described in Proposition 4.1 proving its convexity via a positive semi-definite hessian. Hence noting that $\mathbf{1}^T W^T W\mathbf{1} = \sum w_{ij}^2 = 0$ if and only if $w_{ij} = 0$ for all $(i, j) \in E$, then $f_t(W)$ is a strictly convex function with hessian $\nabla^2 f_t(W) \succeq hI$.

Proposition 4.1. *The derivative w.r.t. w_{ij} and hessian of $\|G(s)\|_2^2$ are*

$$\frac{\partial \|G(s)\|_2^2}{\partial w_{ij}} = -\frac{1}{2}\sum_{p\in R}\left(\left[A_t^{-1}e_p\right]_i - \left[A_t^{-1}e_p\right]_j\right)^2$$

and

$$\nabla^2 \|G(s)\|_2^2 = -\mathscr{E}(\mathcal{G})^T A_t^{-1} B_t B_t^T A_t^{-T} \mathscr{E}(\mathcal{G}) \circ \mathscr{E}(\mathcal{G})^T A_t^{-1} \mathscr{E}(\mathcal{G}).$$

Further, $\nabla^2 \|G(s)\|_2^2$ *is positive semidefinite.*

Proof. The $\mathcal{H}_2$ norm squared is

$$\begin{aligned}\|G(s)\|_2^2 &= -\frac{1}{2}\mathbf{tr}\left(B^T A_t^{-1} B\right)\\ &= -\frac{1}{2}\sum_{p\in R} e_p^T A_t^{-1} e_p\\ &= -\frac{1}{2}\sum_{p\in R} g_p(\mathcal{G}),\end{aligned}$$

where $g_p(\mathcal{G}) = e_p^T A_t^{-1} e_p$. Further,

$$\frac{\partial g_p(A_t(w_{ij}))}{\partial w_{ij}} = \mathbf{tr}\left[\left(\frac{\partial g_p(A_t)}{\partial A_t}\right)^T \frac{\partial A_t}{\partial w_{\ ij}}\right]. \tag{4.6}$$

The relevant derivatives are

$$\begin{aligned}\frac{\partial g_p(A_t)}{\partial A_t} &= \frac{\partial\left(e_p^T A_t^{-1} e_p\right)}{\partial A_t}\\ &= -A_t^{-T} e_p e_p^T A_t^{-T}\end{aligned} \tag{4.7}$$

and from (4.4)

$$\begin{aligned}\frac{\partial A_t}{\partial w_{\ ij}} &= \frac{\partial\left(-\sum_{(i,j)\in E} w_{ij} a_{ij} a_{ij}^T - \sum_{i\in R} e_i e_i^T\right)}{\partial w_{ij}}\\ &= -a_{ij} a_{ij}^T.\end{aligned} \tag{4.8}$$

Substituting (4.7) and (4.8) into (4.6), we have

$$\begin{aligned}\frac{\partial g_p\left(A_t\left(w_{ij}\right)\right)}{\partial w_{ij}} &= \mathbf{tr}\left[\left(-A_t^{-T} e_p e_p^T A_t^{-T}\right)^T a_{ij} a_{ij}^T\right]\\ &= \mathbf{tr}\left[A_t^{-1} e_p e_p^T A_t^{-1} a_{ij} a_{ij}^T\right]\\ &= \mathbf{tr}\left[\left(a_{ij}^T A_t^{-1} e_p\right)\left(e_p^T A_t^{-1} a_{ij}\right)\right]\\ &= \left(a_{ij}^T A_t^{-1} e_p\right)^2\\ &= \left(\left[A_t^{-1} e_p\right]_i - \left[A_t^{-1} e_p\right]_j\right)^2.\end{aligned}$$

Hence,

$$\frac{\partial \left\|G(s)\right\|_2^2}{\partial w_{ij}} = -\frac{1}{2} \sum_{p \in R} \left(\left[A_t^{-1} e_p\right]_i - \left[A_t^{-1} e_p\right]_j \right)^2 .$$

Examining the hessian of g_p,

$$\begin{aligned} \frac{\partial g_p^2 \left(A_t \left(w_{ij}\right)\right)}{\partial w_{ij}^2} &= \frac{\partial}{\partial w_{ij}} \left(a_{ij}^T A_t^{-1} e_p\right)^2 \\ &= \mathbf{tr} \left[\left(\frac{\partial \left(\left(a_{ij}^T A_t^{-1} e_p\right)^2 \right)}{\partial A_t} \right)^T \frac{\partial A_t}{\partial w_{ij}} \right]. \end{aligned} \tag{4.9}$$

The relevant derivatives are

$$\frac{\partial \left(\left(a_{ij}^T A_t^{-1} e_p\right)^2 \right)}{\partial A_t} = 2 a_{ij}^T A_t^{-1} e_p \left(-A_t^{-T} a_{ij} e_p^T A_t^{-T} \right). \tag{4.10}$$

Substituting (4.8) and (4.10) then (4.9) becomes

$$\begin{aligned} \frac{\partial g_p^2 \left(A_t \left(w_{ij}\right)\right)}{\partial w_{ij}^2} &= \mathbf{tr} \left[\left(2 a_{ij}^T A_t^{-1} e_p \left(A_t^{-T} a_{ij} e_p^T A_t^{-T} \right) \right)^T a_{ij} a_{ij}^T \right] \\ &= 2 a_{ij}^T A_t^{-1} e_p \mathbf{tr} \left[A_t^{-1} e_p a_{ij}^T A_t^{-1} a_{ij} a_{ij}^T \right] \\ &= 2 \left(a_{ij}^T A_t^{-1} e_p \right) \left(a_{ij}^T A_t^{-1} e_p \right) \left(a_{ij}^T A_t^{-1} a_{ij} \right) \\ &= 2 \left(a_{ij}^T A_t^{-1} e_p e_p^T A_t^{-T} a_{ij} \right) \left(a_{ij}^T A_t^{-1} a_{ij} \right) \\ &= 2 \left[\mathscr{E}(\mathcal{G})^T A_t^{-1} e_p e_p^T A_t^{-T} \mathscr{E}(\mathcal{G}) \right]_{(\sigma(ij),\sigma(ij))} \cdot \left[\mathscr{E}(\mathcal{G})^T A_t^{-1} \mathscr{E}(\mathcal{G}) \right]_{(\sigma(ij),\sigma(ij))}, \end{aligned}$$

as well as

$$\begin{aligned} \frac{\partial g_p^2 \left(A_t \left(w_{ij}\right)\right)}{\partial w_{ij} w_{kl}} &= \mathbf{tr} \left[\left(2 a_{ij}^T A_t^{-1} e_p \left(A_t^{-T} a_{ij} e_p^T A_t^{-T} \right) \right)^T a_{kl} a_{kl}^T \right] \\ &= 2 a_{ij}^T A_t^{-1} e_p \mathbf{tr} \left[A_t^{-1} e_p a_{ij}^T A_t^{-1} a_{kl} a_{kl}^T \right] \\ &= 2 \left(a_{ij}^T A_t^{-1} e_p \right) \left(a_{kl}^T A_t^{-1} e_p \right) \left(a_{ij}^T A_t^{-1} a_{kl} \right) \end{aligned}$$

$$= 2\left(a_{ij}^T A_t^{-1} e_p e_p^T U^{-1} a_{kl}\right)\left(a_{ij}^T A_t^{-1} a_{kl}\right)$$
$$= 2\left[\mathscr{E}(\mathcal{G})^T A_t^{-1} e_p e_p^T A_t^{-T} \mathscr{E}(\mathcal{G})\right]_{(\sigma(ij),\sigma(kl))} \cdot \left[\mathscr{E}(\mathcal{G})^T A_t^{-1} \mathscr{E}(\mathcal{G})\right]_{(\sigma(ij),\sigma(kl))}.$$

Consequently,

$$\nabla^2 g_p = 2\mathscr{E}(\mathcal{G})^T A_t^{-1} e_p e_p^T A_t^{-T} \mathscr{E}(\mathcal{G}) \circ \mathscr{E}(\mathcal{G})^T A_t^{-1} \mathscr{E}(\mathcal{G},$$

and so

$$\nabla^2 \|G(s)\|_2^2 = -\mathscr{E}(\mathcal{G})^T A_t^{-1} B_t B_t^T A_t^{-T} \mathscr{E}(\mathcal{G}) \circ \mathscr{E}(\mathcal{G})^T A_t^{-1} \mathscr{E}(\mathcal{G}).$$

Now, $\mathscr{E}(\mathcal{G})^T A_t^{-1} B_t B_t^T A_t^{-T} \mathscr{E}(\mathcal{G}) = \left(B_t^T A_t^{-T} \mathscr{E}(\mathcal{G})\right)^T \left(B_t^T A_t^{-T} \mathscr{E}(\mathcal{G})\right) \succcurlyeq 0$ and as A is negative definite then $-\mathscr{E}(\mathcal{G})^T A_t^{-1} \mathscr{E}(\mathcal{G}) \succcurlyeq 0$. Finally, the Hadamard product boasts the property that the Hadamard product of two positive semi-definite matrices is positive semi-definite [3] and so the result follows. □

We proceed to provide a decentralized version of the conjugate gradient method to form $\nabla \|G(s)\|_2^2$ distributively. The gradient is then applied to a distributed formulation of online gradient descent leading to a logarithmic regret bound for problem (4.5).

4.4.1 Local Gradient via Distributed Conjugate Gradient

Necessary to the distributed gradient descent algorithm is the local evaluation of the gradient of $f_t(W)$ with respect to a weight w_{ij}. Examining the derivative result in Proposition 4.1 for $y^s := A(\mathcal{G}, \mathcal{R})^{-1} e_s$ for all $s \in R$ then

$$\frac{\partial \|G(s)\|_2^2}{\partial w_{ij}} = -\frac{1}{2} \sum_{s \in R} \left(y_i^s - y_j^s\right)^2.$$

Consequently, if every agent i has access to y_i^s for all $s \in R$, then neighboring agents i and j can negotiate and calculate the gradient with respect to the edge connecting them.

To this end we propose a distributed form of the linear conjugate gradient method that for each run s, provides β_i^s to each agent i. The centralized version of the conjugate gradient method solves $A^{-1}b$, where $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$. For $A = A(\mathcal{G})$ and $b = e_p$, the method can be used to find β^s. An attraction of the algorithm is that updates only require b and the evaluation of Ay for a given $y \in \mathbb{R}^n$. For the case where A is encoded in the graph structure, this is ideal whereby if each agent i only has knowledge of y_i then $[Ay]_i$ can be calculated by agent i simply by querying its neighbors. Specifically, $[A(\mathcal{G}, \mathcal{R})y]_i = \sum_{j \in N(i)} w_{ij} y_j$.

Protocol 6 Distributed Conjugate Gradient Method

```
Given $y_i[0]$ for each node $i \in V$
Initialize:
$k = 0$
$r_i[0] = \sum_{j \in N(i)} a_{ij} x_j[0] - b_i$
$p_i[0] = -r_i[0]$
$\tilde{r}_i[0] \leftarrow$ Consensus on $r_i^2 \approx \frac{1}{n} r_i^T r_i$
while $|\tilde{r}_i[k]| > \epsilon$ do
    $b_i[k] = \sum_{j \in N(i)} a_{ij} p_j[k]$
    $\tilde{p}_i[k] \leftarrow$ Consensus on $p_i b_i \approx \frac{1}{n} p_i^T A p_i$
    $\alpha_i = \frac{\tilde{r}_i[k]}{\tilde{p}_i[k]}$
    $y_i[k+1] = y_i[k] + \alpha_i p_i[k]$
    $r_i[k+1] = r_i[k] + \alpha_i b_i[k]$
    $\tilde{r}_i[k+1] \leftarrow$ Consensus on $(r_i)^2 \approx \frac{1}{n} r_i^T r_i$
    $\beta_i = \frac{\tilde{r}_i[k+1]}{\tilde{r}_i[k]}$
    $p_i[k+1] = \beta_i p_i[k] - r_i[k+1]$
    $k = k + 1$
end
```

Our distributed conjugate gradient is featured in Algorithm 6, with timesteps indicated by k. The estimate y, residues r, and conjugate p are the main components of the algorithm updates in lines 11, 12, and 15 respectively. Our method takes the same form as the traditional conjugate gradient with the exception of two agreement variables $\tilde{p}$ and $\tilde{r}$ required per timestep, shown in lines 9 and 13, respectively. These are of the form $\frac{1}{n} \sum_{i \in V} r_i^2$ and $\frac{1}{n} \sum p_i b_i$, respectively, and so can be calculated distributively using a traditional unweighted information-based consensus model (1) with $W = I$. Termination occurs when the average residue $\tilde{r}$ falls below a small threshold value ϵ, which for our application $\epsilon = 1e - 6$. Convergence is typically fast with theoretical guarantees to converge in less than n steps.

Interestingly, the convergence rates are strongly tied to the spectrum of A, and for our application the eigenvalues of $\mathcal{G}$. If the eigenvalues are clustered, then the algorithm tends to be more performant. Networks that exhibit this trait are regular graphs and, in general, graphs with many symmetries [1]. In practice the n step guarantee is not always met due to rounding errors and for our case errors in the consensus updates. There are a myriad of techniques to combat this, we found a restart criteria when $\tilde{r}$ increased significantly to be effective and easy to coordinate through the network. For more details on the intricacies of the conjugate gradient method, we invite the reader to examine the text [11] by Nocedal and Wright.

Figure 4.1 shows the residue for a sample run of the distributed conjugate gradient method run on a 20-node graph with four foreign agents. The restart condition was triggered at $k = 7$ due to the increase in $\tilde{r}$. The algorithm converged to $\epsilon = 1e - 10$ in only ten timesteps, much less than the theoretical $n = 20$.

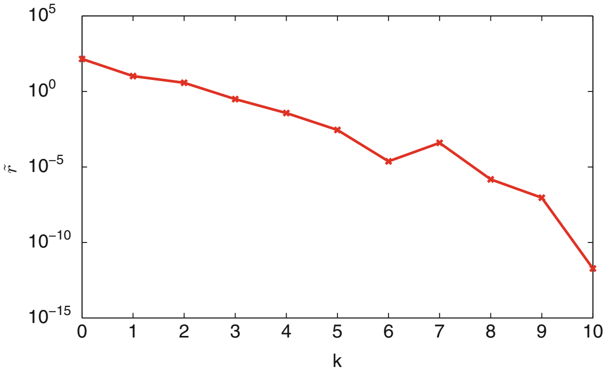


Fig. 4.1 Average residue per timestep of the conjugate gradient method on a 20-node graph

Protocol 7 Distributed Online Gradient Descent

```
*Given convex set* $\mathcal{P} \subset \mathbb{R}^n$ *and some* $w_{ij}[1] \in \mathcal{P}$, $\forall \{v_i, v_j\} \in E$
**Initialize:**
$k = 1$
**foreach** $\{v_i, v_j\} \in E$ **do**
    $g_{ij}[k] = -\frac{1}{2}\sum_{s \in R}\left(y_i^s[k] - y_j^s[k]\right)^2 + hw_{ij}[k]$
    $w_{ij}[k+1] = \prod_{\mathcal{P}}\left(w_{ij}[k] - \eta[k]g_{ij}[k]\right)$
    $k = k + 1$
    Here, $\prod_{\mathcal{P}}(z) = \arg\min_{x \in \mathcal{P}} \|x - z\|_{\infty}$.
**end**
```

4.4.2 Online Algorithm

Given the online nature of problem (4.5) whereby a member of the arbitrary sequence of strictly convex functions $\{f_t(W)\}$ arrives at each timestep, an online convex algorithm is a natural choice. We present a distributed version of the online gradient descent algorithm formulated by Hazan et al. [6]. The algorithm is a variation of the traditional gradient descent algorithm and has run time $O(n)$ per iteration. The distributed version of the algorithm is presented for our cost function $f_t(W)$ in Algorithm 7. At each time step $k+1$ the local gradient $g_{ij}[k]$ (line 4) of edge w_{ij} at time step k is revealed to agents i, j which perform a gradient descent step in line 5. As the step may be infeasible, a projection $\prod_{\mathcal{P}}(\cdot)$ corrects the step by projecting to the closest point on the constraint set $\mathcal{P}$ under the ∞-norm. For the linear constraints in problem (4.5) this can be accomplished distributively and cheaply using the ∞-norm. An attraction of this algorithm is that Hazan et al. [6] proved that if step size $\eta[k] = \frac{1}{hk}$ the regret is $\mathcal{R}_T \leq O\left(\log(T)\right)$, i.e., "on average" the algorithm performs as well as the best fixed strategy in hindsight.

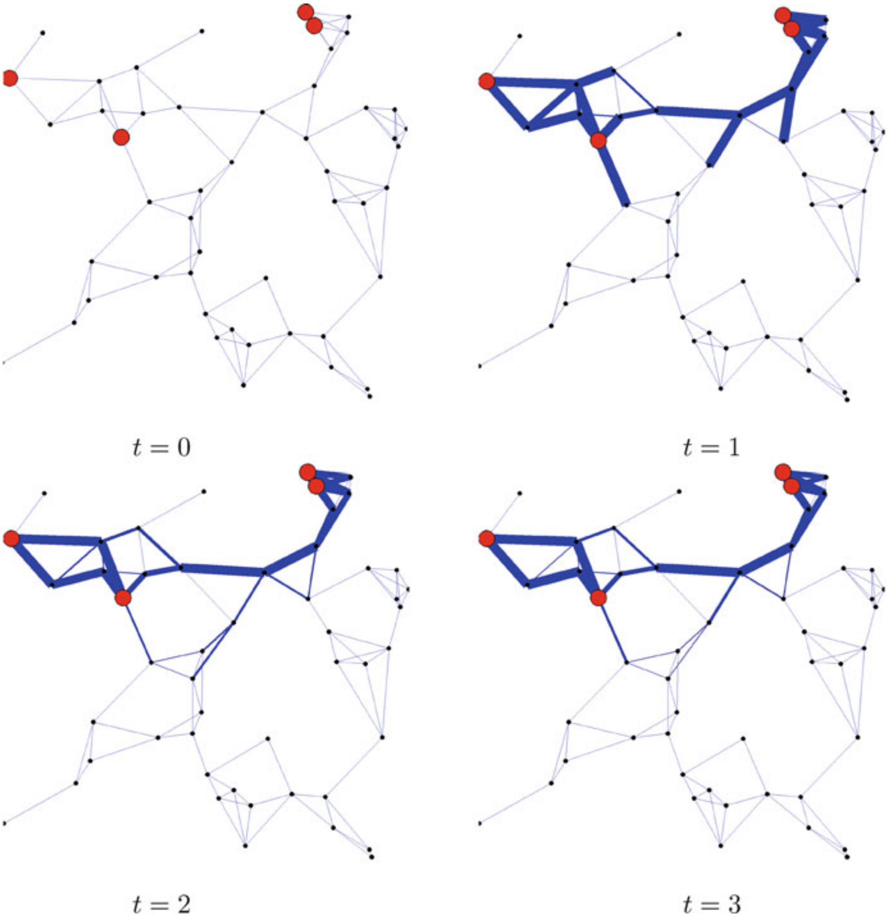
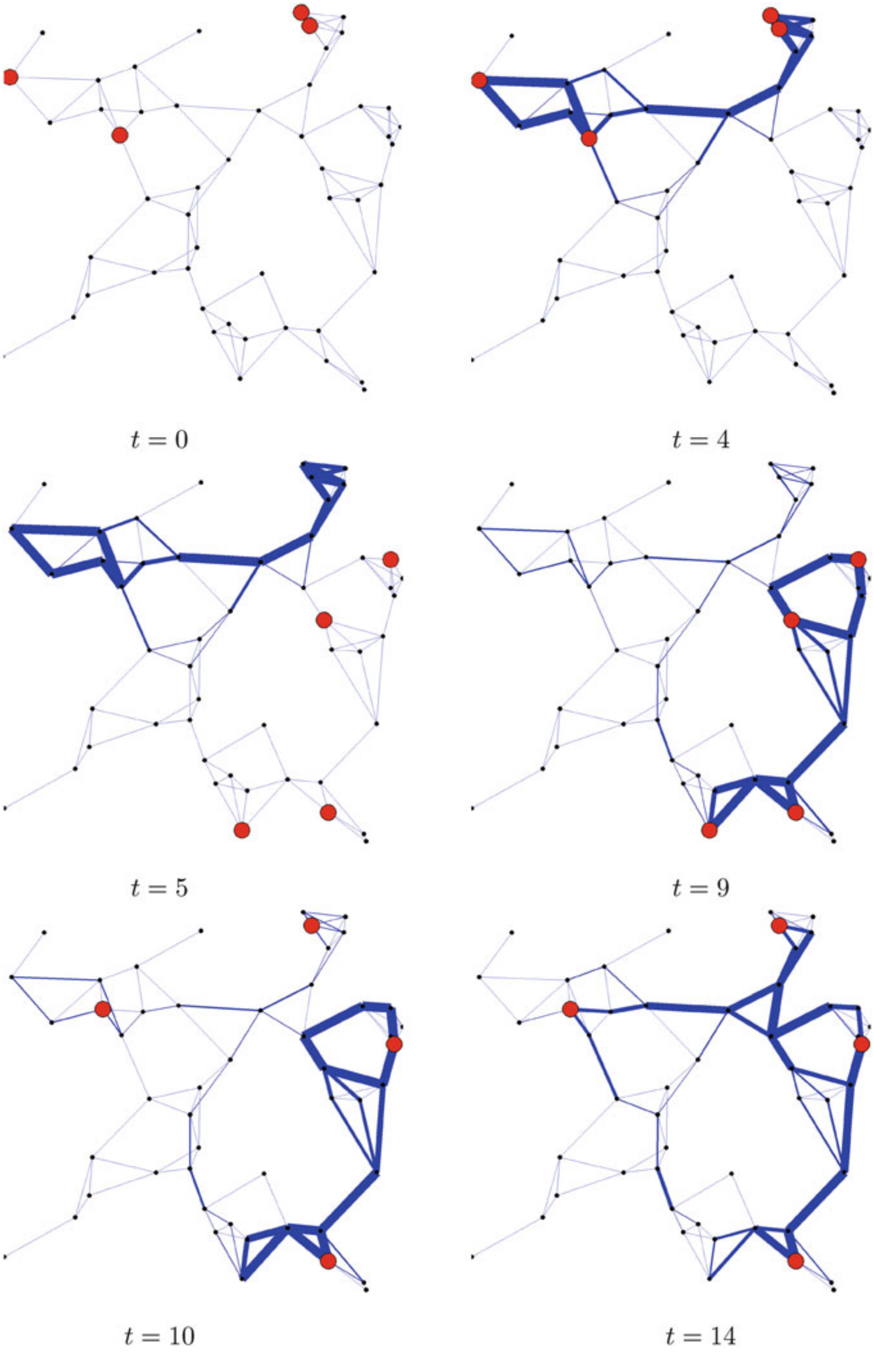


Fig. 4.2 Distributed reweighting with a fixed leader set

The distributed gradient descent algorithm coupled with the distributed conjugate gradient method was applied to a random 50-node graph depicted in Fig. 4.2. Four fixed foreign agents were detected by the agents marked with large circles. The edge weights were initialized uniformly as 0.25 and constrained such that $q_l = 0.05$ and $u_l = 1$ for all edges in E, and $s_i = 4$ for all nodes $i \in V$. The first few iterations are displayed in Fig. 4.2. The notable characteristic is the increasing of edge weight on those edges close to the foreign agents. This aligns with the conductance analog mentioned in Sect. 4.3.2, where additional conductance is added to edges in the neighborhood of the foreign agents. These edges are those most likely to dampen the effect of the foreign agents' signal.

Sub-figures in Fig. 4.3 denote the algorithm's evolution for varying foreign agent locations over time. The time evolution provides insight into the inner working of the online regret framework. Once a foreign agent location is revealed the algorithm


$t = 0$

$t = 4$

$t = 5$

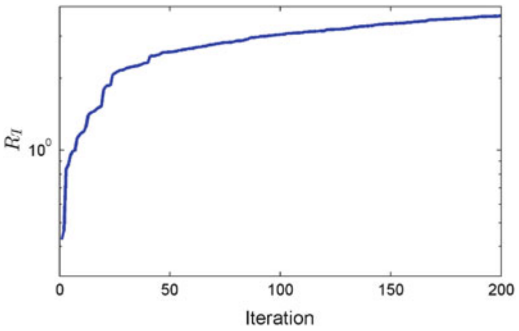
$t = 9$

$t = 10$

$t = 14$

Fig. 4.3 Distributed reweighting with a variable leader set

Fig. 4.4 Regret over time of Algorithm 7



is penalized for the absence of heavily weighted edges in its vicinity, for example, at timesteps $t = 0$ and 5. The algorithm recovers by timesteps $t = 4$ and 9. This process provides the algorithm with a 'memory' of the foreign agent's location notable from the residue edge weights from previous heavily weighted edges.

The best static graph for this foreign agent evolution was calculated over 1,000 iterations. The resultant regret is depicted in Fig. 4.4 emphasizing the performance agreement with the $\mathcal{O}(\log(T))$ bound found by Hazan et al. [6].

4.5 Remarks

The chapter presents a distributed method for the reweighting of network edges so as to dampen the inputs of external signals. The open loop $\mathcal{H}_2$ norm was presented as a metric to quantify the network's susceptibility to such signals. The reweighting algorithm involved the formulation of a distributed conjugate gradient method and a distributed online gradient descent method. The work presents a first foray into the realm of online topology design approaches with proven small regret. The online approach forms an attractive framework to highly uncertain optimization problems. Our future research aims to explore the application of the distributed online approach to the myriad of highly uncertain networked problem.

References

1. B. Bollobás, *Modern Graph Theory*. New York: Springer, 1998.
2. M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton: Princeton University Press, 2010.
3. R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York: Cambridge University Press, 1990.
4. P. Barooah and J. P. Hespanha, "Graph effective resistance and distributed control: spectral properties and applications," in *Proc. 45th IEEE Conference on Decision and Control*, 2006, pp. 3479–3485.

5. M. R. Hestenes and E. Stiefel, “Methods of conjugate gradients for solving linear systems,” *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 409–436, 1952.
6. E. Hazan, A. Agarwal, and S. Kale, “Logarithmic regret algorithms for online convex optimization,” *Machine Learning*, vol. 69, no. 2–3, pp. 169–192, Aug. 2007.
7. D. Bertsekas, “Incremental gradient, subgradient, and proximal methods for convex optimization: A survey,” *Optimization for Machine Learning*, vol. 2010, no. December 2010, pp. 1–38, 2011.
8. F. Yan and S. Sundaram, “Distributed autonomous online learning: regrets and intrinsic privacy-preserving properties,” *arXiv:1006.4039v3*, pp. 1–24, 2010.
9. S. Sundhar Ram, A. Nedić, and V. V. Veeravalli, “Distributed Stochastic Subgradient Projection Algorithms for Convex Optimization,” *Journal of Optimization Theory and Applications*, vol. 147, no. 3, pp. 516–545, Jul. 2010.
10. M. Raginsky, N. Kiarashi, and R. Willett, “Decentralized online convex programming with local information,” in *Proc. American Control Conference*, 2011, pp. 5363–5369.
11. J. Nocedal and S. J. Wright, *Numerical Optimization*. New York: Springer, 2006.

Chapter 5
Network Topology Design for UAV Swarming with Wind Gusts

Abstract The cornerstone of effective topology design for networked systems is the appreciation of the interplay between system performance and network structure. In this chapter, we examine the problem of unmanned aerial vehicle (UAV) swarming in the presence of wind gusts. Firstly, we model an altitude consensus-based leader–follower model exposed to gust disturbances. We then proceed to examine system-theoretic and topological features that promote network disturbance rejection. Specifically the open loop $\mathcal{H}_2$ norm of the system is selected as a performance metric. Its topological features are highlighted via a realization of the open loop $\mathcal{H}_2$ norm in terms of the effective resistance of the corresponding electrical network. We subsequently utilize mixed-integer semidefinite programming (MISDP) to generate the optimal unweighed network to minimize this metric. This is then followed by exploiting the open loop $\mathcal{H}_2$ norm related topological features to design a network rewiring protocol to maximize this metric. Finally, these topology design tools are applied to wind gust rejection in disturbed swarming scenarios, demonstrating the viability of topology-assisted design for improved performance.

5.1 Introduction

Manageability of unmanned aerial vehicle (UAV) swarms presents new challenges at the intersection of interdisciplinary fields. The first challenge pertains to advances in designing small UAVs and micro aerial vehicles (MAVs) involved in complex mission scenarios [13–16]. The reduced physical scale of such vehicles limits their sensing and actuation capabilities, often requiring their cooperative operation for successful missions. The second challenge relates to the improved autonomy of individual and multiple UAVs, removing the need of a single operator for each UAV [17–21].

The effect of wind gusts on aircraft is a well-researched area [22, 23]. The single and collective response of micro aerial vehicle (MAV) swarms to wind gusts [24], on the other hand, is an emerging research field. [25] has investigated a bio-inspired MAV design to combat environmental disturbances. A myriad of controllers have been proposed for such a task; for example, $\mathcal{H}_2$ and $\mathcal{H}_\infty$ controllers have been proposed in [26] and [27]. One of the few chapters that has studied a gust-exposed network-based UAV swarm is [28] who proposed a hybrid fault-detection

A. Chapman, *Semi-Autonomous Networks*, Springer Theses,
DOI 10.1007/978-3-319-15010-9_5

and isolation approach involving thresholding between small tolerable and large disturbances in the system, *without* explicit attention to the coordination topology of the UAV swarm. The approach in this chapter aims instead to maintain the underlying consensus dynamics and favorably alter the coordination topology.

One of the challenges in reducing the size of a UAV is its increased susceptibility to external disturbances, such as wind gusts. An example is provided that examines the wind-gust swarming scenario which involves a swarm of UAVs with the objective of maintaining a constant altitude aided by relative sensing between neighboring UAVs. An $\mathcal{H}_2$ controller is implemented by operators to reject the effect of wind gust disturbance on the overall network. As the swarm is cooperative, favorable controller disturbance corrections can be inadvertently dampened via coordination between UAVs. This chapter presents techniques to manipulate the UAV coordination network topology to improve the swarm's resilience to wind gusts.

Unmanned aerial vehicle (UAV) swarming, the focus of this chapter, is an example of a leader–follower model and involves distributing tasks over many small vehicles which can be coordinated by leader UAVs. One of the costs of the decentralized architecture is increased susceptibility of the small vehicles to external disturbances, such as wind gusts. As the swarm is cooperative, disturbances can be inadvertently amplified via coordination between UAVs. Similarly, favorable disturbance correction can be inadvertently dampened. This chapter manipulates the UAV coordination network topology to improve the swarm's resilience to wind gusts.

We model the UAV swarming with leader–follower dynamics running a consensus-based protocol with the objective of reaching agreement on altitude. Consensus provides a framework for simple, but effective, distributed information-sharing and control for networked, multi-agent systems in settings such as multi-vehicle control, formation control, swarming, and distributed estimation. See, for example, Olfati-Saber et al. [2] and Mesbahi and Egerstedt [5]. One of the popular adaptions of traditional consensus is leader–follower dynamics [3, 4] in which leader agents, that do not conform to traditional consensus, can impact the network by exploiting the other agents' consensus dynamics. This can be done intentionally by knowledgeable controllers, for example leader UAVs correcting for wind disturbances, or unintentionally by powerless agents, for example a UAV affected by a wind gust skews the consensus results. In the former case this impact is favorable and in the latter case is detrimental. We investigate the relationship between the network topology and the impact of the leaders' input in a leader–follower system. Topological features are investigated which vary the leaders' impact, measured in terms of the open loop $\mathcal{H}_2$ norm of the system. Favorable network topologies are then designed optimally using a mixed-integer semidefinite programming (MISDP) framework and suboptimally via a decentralized rewiring protocol, exploiting the aforementioned topological features.

Two specialized cases of wind-gust swarming are considered. The first case is when the UAVs are spread over a large area, for example for *sparse* surveillance of a region, and as such localized wind gusts act on a subset of the UAVs in the swarm.

The UAVs are assumed to be equipped with sensors and controllers to perform simple dampening of their measured wind gusts. In the second case, we consider a *dense* swarm formation, for example high-precision ground sensing. A consequence of the UAV's proximity within a dense swarm is that single wind gusts affect all agents in the swarm. For this second case, full state monitoring of all UAVs can be performed by a subset of agents due to the graph's compactness. An $\mathcal{H}_2$ controller (Linear Quadratic Gaussian (LQG) controller) is implemented on leader agents to reject the wind gust disturbance. These motivating examples are dubbed *sparse* and *dense swarming*, respectively.

Aircraft being exposed to wind gusts is a well-researched area [22, 23]. Gust-exposed MAV [24], on the other hand, is an emerging research field and is highly relevant to this chapter since the agents used in large-scale swarming are often MAVs. Shyy et al. [25] have investigated biology-inspired MAV design to combat environmental disturbances. A myriad of controllers have been proposed for such a task including $\mathcal{H}_\infty$ controllers [27] and $\mathcal{H}_2$ controllers [26]. One of the few chapters that has studied a network-based gust-exposed UAV swarm is Meskin et al. [28] who proposed a hybrid fault-detection and isolation approach involving thresholding between tolerable and large disturbances in the system, ignoring the coordination topology of the UAV swarm. Our approach aims to instead maintain the underlying consensus dynamics and favorably alter the coordination topology.

One of the attractions of networked systems is that system performance is strongly coupled to the underlying network structure. For traditional consensus, system performance and its ties to network structure is a well-researched problem where the second smallest eigenvalue of the graph Laplacian is a favored metric to quantify the convergence rate [4, 6]. Interest has also been shown with other network measures, for example the largest eigenvalue of the graph Laplacian [29]. However, these metrics prove less attractive in leader–follower consensus where convergence rates can vary dramatically depending on where the leaders are located within the network. For the leader–follower consensus, our selected performance metric is the open loop $\mathcal{H}_2$ norm. The system-theoretic interpretation of this metric provides a more tangible link to system performance in the presence of disturbances and leaders. Further, the equivalent circuit representation of the network provides the presentation of the open loop $\mathcal{H}_2$ norm as the total *effective resistance* of the graph agents.

The open loop $\mathcal{H}_2$ norm is studied when the network system is subjected to disturbances or unknown noise, or when the controllability of the system is a concern. Under such circumstances, reducing the open loop $\mathcal{H}_2$ norm, where the input is from the gust-affected UAVs, will decrease the susceptibility of the system to the gust, and increasing the open loop $\mathcal{H}_2$ norm, where the input is from the leaders, tends to increase the receptiveness of the system to control. Work in this area focuses on allocating the weights on the coordination links between UAVs (the edges of the coordination topology) [8]. We examine the coordination graph topologies that achieve the minimum and maximum open loop $\mathcal{H}_2$ norm, given a set of leader UAVs. If the existence or absence of a coordination link between two random UAVs is represented by a Boolean value, the off-diagonal entries of the adjacency matrix can all be expressed by the binary variables one or zero. Our objective is to determine these binary values in order to optimize the performance index.

We address the optimization of the open loop $\mathcal{H}_2$ norm on two fronts. The first uses MISDP [31] to minimize the metric under network connectivity constraints to produce the optimal graph. Unfortunately, maximizing the metric in an optimization setting is NP-hard (a provably intractable problem) and we propose a suboptimal decentralized protocol to improve this metric. Our approach is to perform edge trades among neighboring agents within the network that tend to increase the total effective resistance of the graph and consequently the open loop $\mathcal{H}_2$ norm.

The general area of designing topologies to optimize for certain metrics has been addressed by many authors: Ghosh and Boyd [30] maximized the second smallest eigenvalue of the graph Laplacian, Zelazo and Mesbahi [7] optimized the network $\mathcal{H}_2$ performance where the graph is not in the closed loop of the dynamics, Wan et al. [9] maximized the largest eigenvalue of the graph Laplacian. All aforementioned authors used optimization techniques over *weighted* graphs. Kim and Mesbahi [32] used power functions to approximately represent the on/off linkage relationship when searching for the maximum second-smallest eigenvalue of the Laplacian to increase convergence speeds of the network dynamics. Wu and Wang [33] have approached the same problem using genetic-algorithms. Intuitive based methods of network reconfiguration have been designed to improve network resilience, for example using thresholding methods to decide when to alter the topology [11].

The chapter is organized as follows. Section 5.2 contains the problem formulation and relevant background. Two models for a UAV swarm exposed to wind gusts are also introduced in Sect. 5.2. An analysis of the open loop $\mathcal{H}_2$ norm is presented in Sect. 5.3 and its relationship to system performance and the effective resistance of the network is established. In Sect. 5.4, two methods are proposed to design network topologies optimized for the open loop $\mathcal{H}_2$ norm and conducive to wind-gust disturbance rejection.

5.2 Model

Now consider $x_i(t) \in \mathbb{R}$ to be the i-th node's (or for our case agent's) state at time t. The continuous-time consensus protocol is in Eq. (1) with compact form $\dot{x}(t) = -\mathscr{L}(\mathcal{G})x(t)$.

We next introduce a model for leader–follower consensus which extends the model (3.3) from Chap. 3 allowing more than one input to influence any one node in the graph.

The model is defined over a graph $\breve{\mathcal{G}} = (\breve{V}, \breve{E})$ associated with a pair $\mathcal{R} = (R, \mathcal{E}_R)$, where $R \in \breve{V}$ is the cardinality r leader agents set and $\mathcal{E}_R \subseteq \breve{E}$ is the set of edges used by the leader agents to inject signals into the network. It is assumed that for a leader agent $r_j \in R$ the same signal $u_j(t) \in \mathbb{R}$ is delivered along every

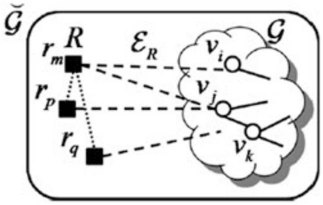


Fig. 5.1 Example of leader–follower notation. The leaders in set R are connected to the followers in V via the edges $\mathcal{E}_R$. The followers V with coordination edges E form a graph $\mathcal{G}$ (some of which is depicted in the image with agents v_i, v_j, v_k and *solid edges*). Edges between leaders (*dotted*) can be ignored and followers. The complete leader and follower graph containing all agents $\breve{V}$ and edges $\breve{E}$ is denoted as $\breve{\mathcal{G}}$

edge adjacent to it. The remaining edges and agents of $\breve{\mathcal{G}}$ form the subgraph $\mathcal{G}$, with the exception of those edges between leaders which are removed. This assumption is justified as we assume leaders are working cooperatively and do not require coupled dynamics of the other leaders' states. Figure 5.1 provides a graphical representation of this notation and setup.

The resulting leader–follower system now assumes the form,

$$\dot{x}_i(t) = \sum_{\{v_i,v_j\}\in E} \big(x_j(t) - x_i(t)\big) + \sum_{\{v_i,r_j\}\in\mathcal{E}_R} \big(u_j(t) - x_i(t)\big) : \tag{5.1}$$

with full dynamics

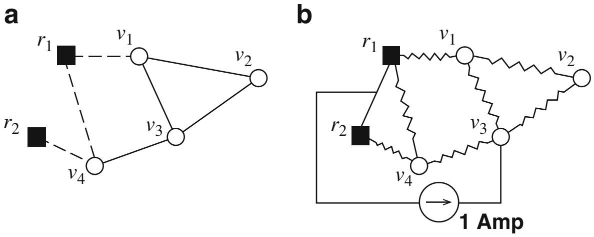
$$\dot{x}(t) = A(\mathcal{G},\mathcal{R})x(t) + B(\mathcal{R})u(t), \tag{5.2}$$

where $B(\mathcal{R}) \in \mathbb{R}^{n\times r}$ with $[B(\mathcal{R})]_{ij} = 1$ when $\{r_j, v_i\} \in \mathcal{E}_R$ and $[B(\mathcal{R})]_{ij} = 0$ otherwise, and

$$A(\mathcal{G},\mathcal{R}) = -\left(\mathscr{L}(\mathcal{G}) + D\left(\mathcal{R}\right)\right) \in \mathbb{R}^{n\times n}, \tag{5.3}$$

where $D\left(\mathcal{R}\right) \in \mathbb{R}^{n\times n}$ with $[D\left(\mathcal{R}\right)]_{ii} = \delta_i^r$ where δ_i^r is the number of leaders adjacent to v_i and $[D(\mathcal{R})]_{ij} = 0$ otherwise. We define δ_j^v as the number of follower agents adjacent to r_j. We distinguish two special cases of this setup: one in which there is exactly one leader for each edge $\mathcal{E}_R$ and so a *distinct* control signal is delivered through each edge, denoted with the leader pair $\mathcal{R}_d$, and one where there exists only one leader node so a *common* signal is delivered through each edge of $\mathcal{E}_R$, denoted with pair $\mathcal{R}_c$. A sample model and its system matrices as well as the special cases of $\mathcal{R}_d$ and $\mathcal{R}_c$ are depicted in Fig. 5.2. We also denote the set of agents v_i such that $\{r_j, v_i\} \in \mathcal{E}_R$ by $\pi\left(\mathcal{E}_R\right)$. In other words, $\pi\left(\mathcal{E}_R\right)$ is simply the set of agents that directly connect to leader agents.

We recognize $A\left(\mathcal{G},\mathcal{R}\right)$ in Eq. (5.3) as the Dirichlet matrix, or grounded Laplacian [10, 12]. The spectrum of $A\left(\mathcal{G},\mathcal{R}\right)$ relates closely to the spectrum of



c

$$A(\mathcal{G},\mathcal{R}) = \begin{bmatrix} -3 & 1 & 1 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & 1 & -3 & 1 \\ 0 & 0 & 1 & -3 \end{bmatrix}, \; B(\mathcal{R}) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$$

$$B(\mathcal{R}_d) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \; B(\mathcal{R}_c) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 2 \end{bmatrix}$$

Fig. 5.2 (**a**) Network graph with leader (control) agents r_1 and r_2 attached to agents v_1 and v_4, leading to an altered Laplacian $A(\mathcal{G},\mathcal{R})$ and input matrix $B(\mathcal{R})$ of model (5.2). (**b**) Equivalent electrical network. The potential difference $V_{v_3} - V_{\mathcal{R}}$ is the effective resistance between v_3 and common resistor node $\{r_1, r_2\}$. (**c**) The control matrices relating to the special cases of distinct and common control

$\mathscr{L}(\mathcal{G})$. In this way, the structure of the underlying graph is related to the dynamics of model (5.2). In the next section we proceed to disturb the leader–follower system with a wind gust. Before this an auxiliary observation about the Dirichlet matrix, to be used subsequently, is the following.

Proposition 5.1 ([35]). *The matrix $A(\mathcal{G},\mathcal{R})$ of model (5.2) is negative definite (and so invertible) if the original graph is connected.*

The following is the characterization of the wind gust in state space form which will be subsequently used in the swarming models.

5.2.1 Wind Model

A vertical wind gust w_g is not white, but has a power spectral density given in Dryden form [23] as

$$\Phi_w(\omega) = 2L\sigma^2 \frac{1 + 3L^2\omega^2}{(1 + L^2\omega^2)^2}, \tag{5.4}$$

with w the frequency in rad/s, σ the turbulence intensity, and L the turbulence scale length divided by true airspeed.

The power spectral density (5.4) can be factored[23] as,

$$\Phi_w(s) = H_w(s)H_w(-s),$$

where

$$H_w(s) = \sigma\sqrt{\frac{6}{L}}\frac{s + 1/L\sqrt{3}}{s^2 + 2s/L + 1/L^2}.$$

Now a realization of $H_w(s)$ corresponding to w, a white noise input with zero mean and unit variance W, is given by

$$\begin{aligned} \dot{z}(t) &= \begin{bmatrix} 0 & 1 \\ -\frac{1}{L^2} & -\frac{2}{L} \end{bmatrix} z(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(t), \\ &= A_w z(t) + B_w w(t) \end{aligned} \tag{5.5}$$

where $z(t) \in \mathbb{R}^2$ are the internal states of the wind gust and

$$\begin{aligned} w_g(t) &= \gamma \begin{bmatrix} \frac{1}{L\sqrt{3}} & 1 \end{bmatrix} z(t) \\ &= C_w z(t) \end{aligned} \tag{5.6}$$

where $\gamma = \sigma\sqrt{6/L}$.

We can now proceed to formulate the full UAV swarming scenarios and incorporate the wind model dynamics.

5.2.2 Swarming Models

We present two swarming models with the objective of reaching and maintaining a common altitude among a network (swarm) of UAVs by using the consensus protocol (5.3). In the first model we consider the swarm being spread over a large area. Due to the large separation of vehicles, different wind gusts affect agents in isolation. Basic gust correction is present on each UAV which has the effect of dampening their wind gust. This model is dubbed *sparse swarming*. The second model examines a swarm grouped together over a small area. Wind gusts no longer act on agents in isolation but affect all agents in the swarm simultaneously. A set of control UAVs (leaders) that can monitor all follower UAVs are introduced and integrated into the consensus dynamics. These control UAVs have onboard altitude maintenance and vary their altitudes so as to correct for such disturbances in the swarm through the consensus dynamics. This model is dubbed *dense swarming*.

For both models, the swarm of UAVs contains a subset of follower UAVs. Each follower examines the relative altitude of vehicles in their neighborhood, via onboard relative altitude sensors, and applies the consensus protocol. The remaining UAVs are considered leader agents which do not obey the consensus protocol, the reason for which distinguishes our two models. For sparse swarming, the gust-affected agents, which are the leaders in this case, *inadvertently* disobey the consensus protocol as each are instead dampening their individual wind gusts. In dense swarming, the leaders are special controller UAVs that are *actively* ignoring the protocol with the objective to correct for wind gusts that are perturbing the velocities of the followers in the network. We subsequently model, for both swarming scenarios, each follower agent as a single integrator with dynamics dictated by the consensus algorithm for altitude alignment. The follower altitude states are $x(t)$ and the leader states are $q(t)$ and $u(t)$ for the sparse and dense swarming models, respectively. The zero altitude ($x(t) = 0, q(t) = 0$ and $u(t) = 0$) can be selected arbitrarily, which for simplicity can be set such that $x(0) = 0$, $q(t) = 0$ and $u(0) = 0$. We now proceed to present our two swarming models for sparse and dense swarming.

5.2.2.1 Sparse Swarming

In sparse swarming, we assume distinct gusts act upon r UAVs of the network. Each UAV has the same capability to perform basic gust correction. They are each equipped with sensors and actuators that can measure and dampen the local wind gust. The subsequent altitude dynamics $q(t) \in \mathbb{R}^r$ of the gust are modeled by

$$q(t) = \bar{h}\omega_g(t) + \bar{x}(0)$$

where $\bar{h} \in \mathbb{R}$ is the dampening effect (common to all UAVs) of the wind gusts $\omega_g(t) \in \mathbb{R}^r$ and $\bar{x}(0) \in \mathbb{R}^r$ is the altitudes of the affected UAVs before the gusts hit.

Incorporating the filter dynamics of Eq. (5.5) with filter state $z_i(t) \in \mathbb{R}^2$ and white noise $w_i(t) \in \mathbb{R}$ corresponding to gust $\left[\omega_g(t)\right]_i$, $\zeta(t) = \left[z_1(t)^T, \ldots, z_r(t)^T\right]^T \in \mathbb{R}^{2r}$ and $\xi(t) = [w_1(t), \ldots, w_r(t)]^T \in \mathbb{R}^r$ then

$$\begin{aligned}
\begin{bmatrix} \dot{x}(t) \\ \dot{\zeta}(t) \end{bmatrix} &= \begin{bmatrix} A(\mathcal{G}, \mathcal{R})x(t) + B(\mathcal{R})q(t) \\ \dot{\zeta}(t) \end{bmatrix} \\
&= \begin{bmatrix} A(\mathcal{G}, \mathcal{R})x(t) + B(\mathcal{R})\left(\bar{h}\omega_g(t) + \bar{x}(0)\right) \\ \dot{\zeta}(t) \end{bmatrix} \\
&= \begin{bmatrix} A(\mathcal{G}, \mathcal{R})x(t) \\ \dot{\zeta}(t) \end{bmatrix} + \begin{bmatrix} \bar{h}B(\mathcal{R}) \\ 0 \end{bmatrix} (I_{r\times r} \otimes C_w)\,\zeta(t) + \begin{bmatrix} B(\mathcal{R})\bar{x}(0) \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} A(\mathcal{G}, \mathcal{R})x(t) \\ (I_{r\times r} \otimes A_w)\,\zeta(t) + (I_{r\times r} \otimes B_w)\,\xi(t) \end{bmatrix}
\end{aligned}$$

$$
\begin{aligned}
&+ \begin{bmatrix} \bar{h}B(\mathcal{R})\,(I_{r\times r} \otimes C_w) \\ 0 \end{bmatrix} \zeta(t) + \begin{bmatrix} B(\mathcal{R})\bar{x}(0) \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} A(\mathcal{G},\mathcal{R}) & \bar{h}B(\mathcal{R})\,(I_{r\times r} \otimes C_w) \\ 0 & (I_{r\times r} \otimes A_w) \end{bmatrix} \begin{bmatrix} x(t) \\ \zeta(t) \end{bmatrix} + \begin{bmatrix} 0 \\ (I_{r\times r} \otimes B_w) \end{bmatrix} \xi(t) \\
&+ \begin{bmatrix} B(\mathcal{R})\bar{x}(0) \\ 0 \end{bmatrix}. \qquad (5.7)
\end{aligned}
$$

We now make the simplifying assumption for modeling that $q(t)$ are arbitrary. For simulation results, we will use the full dynamics in Eq. (5.7).

Noting that all the non-gust-affected UAVs (followers) states $x(t)$ are of interest, the dynamics can be represented as

$$
\begin{aligned}
\dot{x}(t) &= A(\mathcal{G},\mathcal{R})x(t) + B(\mathcal{R})q(t) := \mathcal{A}_S x(t) + \mathcal{B}_S q(t) \qquad (5.8) \\
y(t) &= \mathcal{C}_S x(t),
\end{aligned}
$$

where $y(t) \in \mathbb{R}^n$ is the controller observations, $\mathcal{C}_S = I_{n\times n}$, and $\mathcal{A}_S$ and $\mathcal{B}_S$ are apparent from Eq. (5.8)

The objective of the chapter is to adapt the coordination topology $\mathcal{G}$ and $\mathcal{R}$ to reduce the network amplification of the locally dampened wind gust $q(t)$ on the altitude of the non-gust affected vehicles $y(t)$. In other words, for a fixed $\bar{x}(0)$, to dampen the input to output mapping of $q(t)$ to $y(t)$.

5.2.2.2 Dense Swarming

The assumptions for the dense swarming model are that a controller corresponding to the leader's states $u(t) \in \mathbb{R}^r$ is present and that the same gust $w_g(t) \in \mathbb{R}$ acts upon all agents in the network. Then the full dynamics are

$$
\begin{aligned}
\begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} &= \begin{bmatrix} A(\mathcal{G},\mathcal{R})x(t) + B(\mathcal{R})u(t) \\ \dot{z}(t) \end{bmatrix} + \begin{bmatrix} H \\ 0 \end{bmatrix} w_g(t) \\
&= \begin{bmatrix} A(\mathcal{G},\mathcal{R})x(t) \\ A_w z(t) + B_w w(t) \end{bmatrix} + \begin{bmatrix} B(\mathcal{R}) \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} H \\ 0 \end{bmatrix} C_w z(t) \\
&= \begin{bmatrix} A(\mathcal{G},\mathcal{R})x(t) \\ A_w z(t) + B_w w(t) \end{bmatrix} + \begin{bmatrix} B(\mathcal{R}) \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} A_z \\ 0 \end{bmatrix} z(t) \\
&= \begin{bmatrix} A(\mathcal{G},\mathcal{R}) & HC_w \\ 0 & A_w \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} B(\mathcal{R}) \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ B_w \end{bmatrix} w(t) \qquad (5.9)
\end{aligned}
$$

where $H = [h_1, \ldots, h_n]^T$ and $h_i \in \mathbb{R}$ is a scaling factor specific to UAV i's gust–vehicle interaction.

We assume that the agent states of the system can all be sensed with some measurement noise $v(t) \in \mathbb{R}^n$ which is a white noise vector with correlation matrix $V \in \mathbb{R}^{n \times n}$. Restating Eq. (5.9) in compact form, we have

$$\begin{aligned} \begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} &= \mathcal{A}_D \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + \mathcal{B}_D u(t) + G_D w(t) \\ y(t) &= \mathcal{C}_D \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + v(t), \end{aligned} \tag{5.10}$$

where $\mathcal{A}_D$, $\mathcal{B}_D$, and G_D are apparent from Eq. (5.9), $y(t) \in \mathbb{R}^n$ is the controller observations of which we assume the leaders can observe all the followers' velocities and so $\mathcal{C}_D = \begin{bmatrix} I_{n \times n} & 0_{n \times 2} \end{bmatrix}$.

The objective of the chapter, explored further in the next section, is to adapt the coordination topology $\mathcal{G}$ and $\mathcal{R}$ to increase the amplification of the control input $u(t)$ on the altitude of all UAVs $y(t)$. In other words, to amplify the input to output mapping of $u(t)$ to $y(t)$. To quantify the amplification of $u(t)$ to $y(t)$ as well as the dampening of $q(t)$ to $y(t)$ we use the open loop $\mathcal{H}_2$ norm of the system, which will be discussed in the following section.

5.3 Open Loop $\mathcal{H}_2$ Norm

As a group of networked UAVs do not require physical interconnections for their coordinated behavior, they have the advantage that their inter-vehicle coordination graph can be *rewired*. This observation leads us into our next form of gust correction, namely via topology design. We present a system-theoretic metric that can be exploited to adapt the network topology with the objective of improving the nominal $\mathcal{H}_2$ performance. The open loop $\mathcal{H}_2$ norm of the system proves to be particularly suitable for such an analysis. For a system of the form

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t), \end{aligned}$$

the open loop $\mathcal{H}_2$ norm can be defined in terms of the controllability gramian, $P_T(A, B) := \int_0^T e^{A\tau} BB^T e^{A^T\tau} d\tau$ for the system. The square of open loop $\mathcal{H}_2$ norm is

$$\|G(s)\|_2^2 = \mathbf{tr}\left(CP_\infty(A, B)\, C^T\right), \tag{5.11}$$

where the state-space realization is $G(s) = C\,(sI - A)^{-1}\,B$. For $C = I$, this is a scaled version of the previous chapter's variance damping measure (3.8), specifically, $\|G(s)\|_2^2 = \frac{n}{2} J_\sigma\,(\mathcal{G}, \mathcal{R})$.

b) The energy of the states at the output from a unit impulse input u when $x(0) = 0$ is

$$\int_0^\infty x(t)^T x(t) dt = \|G(s)\|_2^2 .$$

From these observations we can state that inputs perturb the outputs more as $\|G(s)\|_2$ increases. With this in mind we are motivated to generate network topologies that increase this scalar metric when the leader inputs $u(t)$ are favorable as in model (5.10) where the leaders correcting for wind disturbances in the network and decrease this metric for unfavorable gust related inputs $q(t)$ as in model (5.8) where the leaders are delivering adverse wind gusts into the network. Before proceeding with the network generation we examine some properties of $\|G(s)\|_2$ with respect to our specific models (5.8) and (5.10). For esthetics, we will present results and optimize for the square of the open loop $\mathcal{H}_2$ norm $\|G(s)\|_2^2$ but make the observation that maximizing or minimizing $\|G(s)\|_2^2$ inadvertently maximizes and minimizes $\|G(s)\|_2$.

In our sparse swarming model (5.8), if we consider the wind gust effect is analogous to a control input we can formulate the system's controllability gramian and corresponding $\|G(s)\|_2$. The related gramian and $\|G(s)\|_2$ for the sparse and dense swarming models are presented in the following proposition.

Proposition 5.2. *The controllability gramian as $t \rightarrow \infty$ corresponding to model (5.8) from input $q(t)$ to output $y(t)$ is*

$$P_\infty(\mathcal{A}_S, \mathcal{B}_S) = P_\infty(A(\mathcal{G}, \mathcal{R}), B(\mathcal{R})),$$

and corresponding to model (5.10) from input $u(t)$ to output $y(t)$ is

$$P_\infty(\mathcal{A}_D, \mathcal{B}_D) = \begin{bmatrix} P_\infty(A(\mathcal{G}, \mathcal{R}), B(\mathcal{R})) & 0 \\ 0 & 0 \end{bmatrix},$$

consequently the wind gust related states are uncontrollable. Thus for, $G_S(s) = \mathcal{C}_S(sI - \mathcal{A}_S)^{-1}\mathcal{B}_S$ and $G_D(s) = \mathcal{C}_D(sI - \mathcal{A}_D)^{-1}\mathcal{B}_D$ then $\|G_S(s)\|_2^2 = \|G_D(s)\|_2^2 = \mathbf{tr}(P_\infty(A(\mathcal{G}, \mathcal{R}), B(\mathcal{R})))$.

Proof. The sparse models results follow directly from the definition of the controllability gramian and Eq. (5.11). The matrix $P_\infty(\mathcal{A}_D, \mathcal{B}_D)$ is the controllability gramian and therefore $\mathcal{A}_D P_\infty(\mathcal{A}_D, \mathcal{B}_D) + P_\infty(\mathcal{A}_D, \mathcal{B}_D)\mathcal{A}_D^T = -\mathcal{B}_D\mathcal{B}_D^T$, and so

$$\mathcal{A}_D \begin{bmatrix} P_\infty(A(\mathcal{G}, \mathcal{R}), B(\mathcal{R})) & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} P_\infty(A(\mathcal{G}, \mathcal{R}), B(\mathcal{R})) & 0 \\ 0 & 0 \end{bmatrix} \mathcal{A}_D^T$$

$$= \begin{bmatrix} A(\mathcal{G}, \mathcal{R}) & A_z \\ 0 & A_w \end{bmatrix} \begin{bmatrix} P_\infty(A(\mathcal{G}, \mathcal{R}), B(\mathcal{R})) & 0 \\ 0 & 0 \end{bmatrix}$$

$$
\begin{aligned}
&+ \begin{bmatrix} P_\infty(A(\mathcal{G},\mathcal{R}), B(\mathcal{R})) & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} A(\mathcal{G},\mathcal{R})^T & 0 \\ A_z^T & A_w^T \end{bmatrix} \\
&= \begin{bmatrix} A(\mathcal{G},\mathcal{R})P_\infty(A(\mathcal{G},\mathcal{R}), B(\mathcal{R})) + P_\infty(A(\mathcal{G},\mathcal{R}), B(\mathcal{R}))A(\mathcal{G},\mathcal{R})^T & 0 \\ 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} -B(\mathcal{R})B(\mathcal{R})^T & 0 \\ 0 & 0 \end{bmatrix} \\
&= -\begin{bmatrix} B(\mathcal{R}) \\ 0 \end{bmatrix} \begin{bmatrix} B(\mathcal{R})^T & 0 \end{bmatrix} \\
&= -\mathcal{B}_D \mathcal{B}_D^T .
\end{aligned}
$$

Applying Eq. (5.11) the proposition follows. □

From Proposition 5.2, the open loop $\mathcal{H}_2$ norm is common to both sparse and dense swarming models. The distinct difference is that the $\mathcal{H}_2$ norm of interest for the sparse model is mapping the dampened wind gust $q(t)$ to the output $y(t)$ and for the dense model the leader control $u(t)$ to the output $y(t)$. We will henceforth refer to this metric as $\|G_{\mathcal{G},\mathcal{R}}(s)\|_2$ as it is solely dependent on $\mathcal{G}$ and $\mathcal{R}$. Similarly, we will denote $P_\infty(A(\mathcal{G},\mathcal{R}), B(\mathcal{R}))$ as $P(\mathcal{G},\mathcal{R})$.

Directly from the definition of the controllability gramian, one has

$$
\begin{aligned}
\|G_{\mathcal{G},\mathcal{R}}(s)\|_2^2 &= \mathbf{tr}\left(\int_0^\infty e^{A(\mathcal{G},\mathcal{R})\tau} B\,(\mathcal{R})\, B\,(\mathcal{R})^T\, e^{A(\mathcal{G},\mathcal{R})^T\tau} d\tau\right) \qquad (5.12)\\
&= \mathbf{tr}\left(B\,(\mathcal{R})\, B\,(\mathcal{R})^T \int_0^\infty e^{2A(\mathcal{G},\mathcal{R})\tau} d\tau\right) \\
&= -\frac{1}{2}\mathbf{tr}\left(M\,(\mathcal{R})\, A(\mathcal{G},\mathcal{R})^{-1}\right), \qquad (5.13)
\end{aligned}
$$

where $M\,(\mathcal{R}) = B\,(\mathcal{R})\, B\,(\mathcal{R})^T$.

In the previous chapter a resistive electrical network interpretation was provided for the diagonal of the matrix $-A(\mathcal{G},\mathcal{R})^{-1}$. An example of the equivalent electrical network is displayed in Fig. 3.3.

We proceed to analyze this metric for our two special leader-agent cases; with $|R| = |\mathcal{E}_R|$ and $|R| = 1$ corresponding to $\mathcal{R}_d$ and $\mathcal{R}_c$ (defined in Sect. 2.2), respectively.

Proposition 5.3. *For a connected graph $\mathcal{G}$, if each leader agent has exactly one edge and so each edge in $\mathcal{R}_d$ can have an independent signal then,*

$$
\|G_{\mathcal{G},\mathcal{R}_d}(s)\|_2^2 = \frac{1}{2} \sum_{v_i \in \pi(\mathcal{E}_R)} E_{\text{eff}}\,(v_i)\,. \qquad (5.14)
$$

Proof. This result follows directly from Lemma 3.13. □

Proposition 5.4. *For a connected graph $\mathcal{G}$, and all agents apply the same signal then*

$$\|G_{\mathcal{G},\mathcal{R}_c}(s)\|_2^2 = \frac{1}{2}\left|\mathcal{E}_R\right|.$$

Proof. We have $B(\mathcal{R}_c) = B(\mathcal{R}_d)\mathbf{1}_u$ and $A(\mathcal{G},\mathcal{R})^{-1}B(\mathcal{R}_c) = A(\mathcal{G},\mathcal{R})^{-1}B(\mathcal{R}_d)$ $\mathbf{1}_u = -\mathbf{1}_x$ hence

$$\begin{aligned}\mathbf{tr}\left(M\left(\mathcal{R}\right)\int_0^\infty e^{2A(\mathcal{G},\mathcal{R})\tau}d\tau\right) &= \mathbf{tr}\left(B(\mathcal{R}_c)^T A(\mathcal{G},\mathcal{R})^{-1}B(\mathcal{R}_c)\right)\\ &= \mathbf{tr}\left(\mathbf{1}_u^T B(\mathcal{R}_d)^T\mathbf{1}_x\right) = \left|\mathcal{E}_R\right|.\end{aligned}$$

From (5.13), the statement of the lemma now follows. □

Corollary 5.5. *For a graph $\breve{\mathcal{G}}$ and model (3.2) with one leader,*

$$\|G_{\mathcal{G},\mathcal{R}}(s)\|_2^2 = \frac{1}{2}\delta(r_1),$$

where $\delta(r_1)$ is the degree of the leader in the graph $\breve{\mathcal{G}}$.

Proof. The statement of the corollary follows directly from Proposition 5.4. □

Remark 5.6. The implication of Corollary 5.5 is that for the case of a single leader, selecting the agent within the network with the highest degree will maximize the open loop $\mathcal{H}_2$ norm of the system, regardless of the structure of the network.

We now relate the controllability gramian of a generic $\mathcal{R}$ with the controllability gramians of special cases of $\tilde{\mathcal{R}}_d$ and $\tilde{\mathcal{R}}_c$. We design these special leader sets cases such that: a control u_c of system $\tilde{\mathcal{R}}_c$ maps to a control u of system $\mathcal{R}$ so that $B(\tilde{\mathcal{R}}_c)u_c = B(\mathcal{R})u$ when $H_1u_c = u$ and so that $\|u_c\|_2 = 1 \implies \|u\|_2 = 1$, for some H_1. Similarly, $B(\mathcal{R})u = B(\tilde{\mathcal{R}}_d)u_d$ when $H_2u = u_d$ and so that $\|u\|_2 = 1 \implies \|u_d\|_2 = 1$, for some H_2. This corresponds to $B\left(\mathcal{R}\right) = B(\tilde{\mathcal{R}}_d)H_2$, $B(\tilde{\mathcal{R}}_c) = B(\mathcal{R})H_1$, where $H_1 = \frac{1}{\sqrt{|R|}}\mathbf{1} \in \mathbb{R}^{|R|\times 1}$, $H_2 \in \mathbb{R}^{|R_d|\times|R|}$ and

$$[H_2]_{ij} = \begin{cases}\frac{1}{\sqrt{\delta_j^v}} & \text{if } \left(r_j, v_i\right) \in \mathcal{E}_R\\ 0 & \text{otherwise.}\end{cases}$$

Therefore,

$$\left[B(\tilde{\mathcal{R}}_c)\right]_i = \begin{cases}\frac{n_i}{\sqrt{|R|}} & \text{if } \left(r_j, v_i\right) \in \mathcal{E}_R\\ 0 & \text{otherwise,}\end{cases}$$

and

$$\left[B(\tilde{\mathcal{R}}_d)\right]_{ij} = \begin{cases} \sqrt{\delta_j^v} & \text{if } \left(r_j, v_i\right) \in \mathcal{E}_R \\ 0 & \text{otherwise.} \end{cases}$$

Proposition 5.7. *For a graph $\mathcal{G}$ and the leader sets defined above for $\tilde{\mathcal{R}}_c$ and $\tilde{\mathcal{R}}_d$,*

$$P(\mathcal{G}, \tilde{\mathcal{R}}_c) \preceq P(\mathcal{G}, \mathcal{R}) \preceq P(\mathcal{G}, \tilde{\mathcal{R}}_d).$$

Proof. We note that by design for every $\|u_c\|_2 = 1$ there exists a $\|u\|_2 = 1$ such that $\left|B(\tilde{\mathcal{R}}_c)u_c\right|_2 \leq |B(\mathcal{R})u|_2$ and similarly for every $\|u\|_2 = 1$ there exists a $\|u_d\|_2 = 1$ such that $|B(\mathcal{R})u_c|_2 \leq \left|B(\tilde{\mathcal{R}}_d)u\right|_2$. The statement of the proposition follows. □

Lemma 5.8. *For a graph $\mathcal{G}$ and leader set $\mathcal{R}$,*

$$\frac{1}{2\,|R|}\,|\mathcal{E}_R| \leq \|G_{\mathcal{G},\mathcal{R}}(s)\|_2^2 \leq \frac{1}{2} \sum_{v_i \in \pi(\mathcal{E}_R)} \alpha_i\, E_{\text{eff}}\,(v_i)\,,$$

where

$$\alpha_i = \sum_{(r_j, v_i) \in \mathcal{E}_R} \delta_j^v,$$

i.e., α_i is the sum of the non-leader degrees for each leader attached to v_i.

Proof. From Proposition 5.7,

$$\left\|G_{\mathcal{G},\tilde{\mathcal{R}}_c}(s)\right\|_2^2 \leq \|G_{\mathcal{G},\mathcal{R}}(s)\|_2^2 \leq \left\|G_{\mathcal{G},\tilde{\mathcal{R}}_d}(s)\right\|_2^2,$$

and

$$\begin{aligned} \mathbf{tr}\left(P(\mathcal{G}, \tilde{\mathcal{R}}_c)\right) &= \mathbf{tr}\left(B(\tilde{\mathcal{R}}_c)^T A(\mathcal{G}, \mathcal{R})^{-1} B(\tilde{\mathcal{R}}_c)\right) \\ &= \frac{1}{|R|}\mathbf{tr}\left(B(\mathcal{R}_c)^T A(\mathcal{G}, \mathcal{R})^{-1} B(\mathcal{R}_c)\right) \\ &= \frac{1}{|R|}\mathbf{tr}\left(P(\mathcal{G}, \mathcal{R}_c)\right). \end{aligned}$$

We note that $M\,(\mathcal{R}_d)$ is a diagonal matrix with $[M\,(\mathcal{R}_d)]_{ii} = \delta_j^v$ if $\left(r_j, v_i\right) \in \mathcal{E}_R$ and $[M\,(\mathcal{R}_d)]_{ii} = 0$, otherwise, and so

$$\left[M\,(\mathcal{R}_d)\, A(\mathcal{G}, \mathcal{R}_d)^{-1}\right]_{ii} = \begin{cases} \alpha_i \left[A(\mathcal{G}, \mathcal{R}_f)^{-1}\right]_{ii} & \text{if } v_i \in \pi\,(\mathcal{E}_R) \\ 0 & \text{otherwise.} \end{cases}$$

The proof of the lemma follows. □

Remark 5.9. An in-depth analysis of $\left\| G_{\mathcal{G},\tilde{\mathcal{R}}_d}(s) \right\|_2^2$ was undertaken for the specialized class of graphs, namely trees $\mathcal{T}$ [34]. To apply some of these results to more generalized connected graphs we consider any spanning tree $\mathcal{T}$ of a connected graph $\mathcal{G}$. In terms of our electrical resistance analogy, the resistor network corresponding to $\mathcal{T}$ is formed by removing resistors from the resistor network corresponding to $\mathcal{G}$. Applying Rayleigh's Monotonicity Principle[1] leads to $\left\| G_{\mathcal{G},\tilde{\mathcal{R}}_d}(s) \right\|_2^2 \leq \left\| G_{\mathcal{T},\tilde{\mathcal{R}}_d}(s) \right\|_2^2$, i.e., the open loop $\mathcal{H}_2$ norm of the system with the coordination graph $\mathcal{G}$ is bounded above by the open loop $\mathcal{H}_2$ norm system with the coordination graph $\mathcal{T}$ corresponding to any of the spanning trees of $\mathcal{G}$.

5.4 Topology Design

We now consider the design of network topologies that exhibit favorable gust rejection characteristics for sparse and dense swarming. We will proceed to introduce two methods to generate these networks using the open loop $\mathcal{H}_2$ norm as the optimizing metric.

Before continuing we will introduce constraints on the underlying graph structure that are products of the swarming application. Firstly, for all UAVs in the network to reach a common consensus there must be some path connecting each and every UAV along some subset of graph edges, i.e., the graph must be connected. The graph nodes represent UAVs. As the number of UAVs is constant in time, the number of nodes n is fixed. An edge $\{v_i, v_j\} \in E$ in the graph indicates that UAV i is sensing the relative altitude of UAV j and vice versa. The total sensing load on the network is subsequently $2|E|$ and applying an upper bound on $2|E|$ equates to limiting this load. UAV i's sensors can also be limited to a certain quantity of concurrently relative altitude measurements of neighboring UAVs. From a graph-theoretic perspective, this is represented as an upper bound on node i's degree δ_i. Finally, the ability of UAV i to perform accurate relative sensing of UAV j is limited by the Euclidean separation of the two UAVs. A Euclidean based constraint graph can be formed where edges exist between UAVs when they lie in each other's sensing range. For topology design, only a subgraph of this constraint graph can be selected. We assume that inter-UAVs distances in the horizontal plane are roughly constant and much greater than the vertical inter-UAV distance, as such the constraint graph is fixed as perturbations in the vertical position do not alter the constraint graph.

[1] Rayleigh's Monotonicity Law states that if the edge resistance in an electrical network is decreased, then the effective resistance between any two agents in the network can only decrease [1].

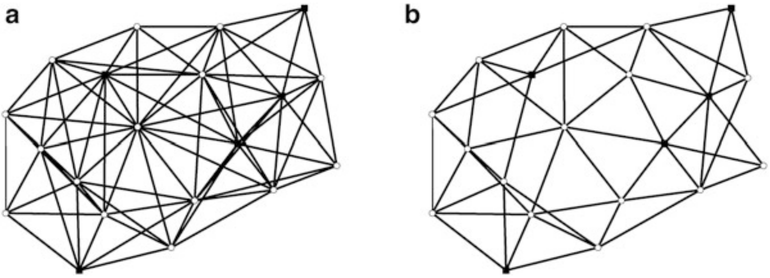


Fig. 5.3 Agent graph with leader agents (*squares*). (**a**) A Euclidean constraint graph (**b**) A connected subgraph of (**a**) meeting the global and local edge constraints, $|E| \leq 51$ and $\delta_i \leq 7$ for all $v_i \in V$

Figure 5.3a depicts a Euclidean based constraint graph on 20-nodes. Further, we provide a graph in Fig. 5.3b that satisfies the connected Euclidean subgraph constraint with the added global and local load sensing constraints such that $|E| \leq 51$ and $\delta_i \leq 7$ for all $v_i \in V$.

We now have the necessary framework to formulate our methods for network topology design which are MISDP and distributed adaptive network design suitable for sparse and dense swarming, respectively.

5.4.1 MISDP Design and Sparse Swarming

According to the symmetric property of the adjacent matrix $\mathscr{A}(\mathcal{G})$, a graph with n nodes can use a set of binary variables comprised of $n(n-1)/2$ elements to determine off-diagonal entries of the $\mathscr{A}(\mathcal{G})$ while the diagonals are simply zeros. Using such a framework, we assign binary variable a_{ij} to represent the element $[\mathscr{A}(\mathcal{G})]_{ij}$ in $\mathscr{A}(\mathcal{G})$ such that

$$\mathscr{A}(\mathcal{G}) = \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ a_{21} & \ddots & a_{ij} & \vdots \\ \vdots & a_{ji} & \ddots & \vdots \\ a_{n1} & \cdots & \cdots & 0 \end{pmatrix}, \quad a_{ij} = a_{ji},\ (i \neq j). \tag{5.15}$$

If the cost of constructing a graph with adjacency expressed in Eq. (5.15) is estimated by the number of edge set E, then we can scale the cost by

$$C_E = \sum_{\{v_i, v_j\} \in V} a_{ij}/2; \quad a_{ij} = a_{ji},\ (i \neq j). \tag{5.16}$$

The degree matrix $\mathscr{D}(\mathcal{G})$ of the graph can also be expressed in terms of the binary variables a_{ij} as

$$\mathscr{D}(\mathcal{G}) = \begin{pmatrix} \sum a_{1j} & & 0 \\ & \ddots & \\ 0 & & \sum a_{nj} \end{pmatrix}.$$

Therefore the Laplacian $\mathscr{L}(\mathcal{G}) = \mathscr{D}(\mathcal{G}) - \mathscr{A}(\mathcal{G})$ is completely determined by the binary variables a_{ij}. We quote the following well-known Lemma [5] which can be used to determine the connectivity of the graph.

Lemma 5.10. *The graph $\mathcal{G}$ is connected if and only if $\lambda_2(\mathcal{G}) > 0$.*

When the network is injected with noise, our goal is to design the topology of the graph by determining the coordination linkages between nodes such that the effective resistance is minimized. At the same time, we want to constrain the construction cost C_E, i.e, the number of edges in the graph. Moreover, the sensing capacity of each node $v_i \in V$ is constrained by the maximum degrees $\Delta_{\max i}$. Finally, the necessary and sufficient condition for a connected graph requires that $\lambda_2(\mathcal{G}) > 0$. In summary, we can formulate the optimal graph design problem as

$$\min_{A(\mathcal{G},\mathcal{R})} \mathbf{tr}(B(\mathcal{R})B(\mathcal{R})^T A(\mathcal{G},\mathcal{R})^{-1}) \tag{5.17}$$

$$s.t. \quad a_{ij} = a_{ji}, \quad \forall\, v_i, v_j \in V, \quad i \neq j \tag{5.18}$$

$$a_{ij} \in \{0, 1\} \tag{5.19}$$

$$\sum_{i,j=1, i\neq j}^{n} a_{ij}/2 \leq C_E \tag{5.20}$$

$$\sum_{j=1}^{n} a_{ij} \leq \Delta_{\max i}, \quad v_i \in V \tag{5.21}$$

$$\lambda_2(\mathcal{G}) > 0 \tag{5.22}$$

The next step is to transform the objective and constraint functions into linear matrix equalities or inequalities as an MISDP problem. Equations (5.18)–(5.21) are already expressed in linear form. Our concern therefore is with Eqs. (5.17) and (5.22). We first present a linear inequality constraint to ensure the second smallest eigenvalue of graph Laplacian is positive via the following proposition and corollary.

Proposition 5.11. *For a graph Laplacian $\mathscr{L}(\mathcal{G})$ the eigenvalues of $\mathscr{L}(\mathcal{G})$ are equivalent to the eigenvalues of $P^T\mathscr{L}(\mathcal{G})P$, where $P = [p_1, p_2, \ldots, p_{n-1}, \frac{1}{n}\boldsymbol{1}]$ and the unit vector are chosen such that*

$$p_i^T\boldsymbol{1} = 0, \quad i = 1, 2, \ldots, n-1 \tag{5.23}$$

and

$$p_i^T p_j = 0, \quad i \neq j. \tag{5.24}$$

Proof. It is well known the Laplacian $\mathscr{L}(\mathcal{G})$ is positive semidefinite matrix and has one eigenvalue equal to zero with related eigenvector $\mathbf{1}$. The eigenvector x associated with the nonzero eigenvalue λ satisfies the following condition

$$x^T \mathscr{L}(\mathcal{G}) = \lambda x, \quad \text{for all nonzero} \quad x \in \mathbf{1}^{\perp} \tag{5.25}$$

where eigenvector x corresponds to a nonzero eigenvalue and is contained in the set of all vectors orthogonal to $\mathbf{1}$, denoted by $\mathbf{1}^{\perp}$. However, without loss of generality, vector x could be the linear combination

$$x = \alpha_1 p_1 + \alpha_2 p_2 + \cdots + \alpha_n p_n,$$

such that $x = Py$ and $y = [\alpha_1, \alpha_2, \cdots, \alpha_n]$ has nonzero elements. Substituting x in Eq. (5.25) one has

$$(Py)^T \mathscr{L}(\mathcal{G}) = \lambda P y \iff y^T (P^T \mathscr{L}(\mathcal{G}) - \lambda P) y = 0 \tag{5.26}$$

Obviously, Eq. (5.26) exists if and only if $P^T \mathscr{L}(\mathcal{G}) = \lambda P$. □

Corollary 5.12. *For a graph Laplacian $\mathscr{L}(\mathcal{G})$ the constraint $\lambda_2(\mathcal{G}) > 0$ is equivalent to $\mathscr{L}(\mathcal{G}) + \mathbf{11}^T/n \succ 0$.*

Proof. It is easy to confirm that the matrix $\mathbf{11}^T/n$ has one eigenvalue equal to 1 with corresponding eigenvector of $\mathbf{1}$ and the remaining eigenvalue are all equal to zero. We assume the eigenvectors of matrix $\mathbf{11}^T/n$ are denoted by $P = [p_1, p_2, \ldots, p_{n-1}, \mathbf{1}]$ where all elements of P satisfies exact the same condition as stated in Eqs. (5.23) and (5.24). From Proposition 5.11, one has

$$P^T \mathscr{L}(\mathcal{G}) = \begin{pmatrix} \lambda_1(\mathcal{G}) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_n(\mathcal{G}) \end{pmatrix} P,$$

where $\lambda_1 = 0$ with eigenvector $\mathbf{1}$. We proceed with calculating the eigenvalues of matrix $\mathscr{L}(\mathcal{G}) + \mathbf{11}^T/n$,

$$P^T (\mathscr{L}(\mathcal{G}) + \mathbf{11}^T/n) = \begin{pmatrix} \lambda_1(\mathcal{G}) + 1 & 0 & 0 & 0 \\ 0 & \lambda_2(\mathcal{G}) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \lambda_n(\mathcal{G}) \end{pmatrix} P.$$

As we assigned $0 = \lambda_1(\mathcal{G}) \le \lambda_2(\mathcal{G}) \le \cdots \le \lambda_n(\mathcal{G})$, $\lambda_2(\mathcal{G}) > 0$ is satisfied if and only if $\mathscr{L}(\mathcal{G}) + \mathbf{11}^T/n \succ 0$. □

Next, we introduce a slack symmetric matrix $S_{n\times n}$ such that

$$\begin{pmatrix} S & I_{n\times n} \\ I_{n\times n} & A(\mathcal{G},\mathcal{R}) \end{pmatrix} \succeq 0 \Longleftrightarrow S \succeq A(\mathcal{G},\mathcal{R})^{-1}.$$

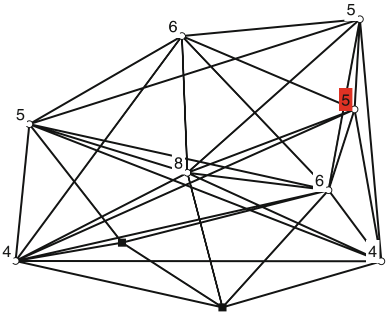
By minimizing the trace of upper bounds of $A(\mathcal{G},\mathcal{R})^{-1}$, which we relabel matrix S, we simultaneously reach the minima of the desired $\mathbf{tr}(A(\mathcal{G},\mathcal{R})^{-1})$ without performing the nonlinear inverse calculation of $A(\mathcal{G},\mathcal{R})$. The optimization problem proposed from Eqs. (5.17)–(5.22) is now transformed into an MISDP problem summarized in the following formulation:

$$\begin{aligned} &\min_{A(\mathcal{G},\mathcal{R})} \mathbf{tr}(B(\mathcal{R})B(\mathcal{R})^T S) \\ s.t.\quad & a_{ij} = a_{ji}, \quad \forall\, v_i, v_j \in V, \quad i \ne j \\ & a_{ij} \in \{0,1\} \\ & \sum_{i,j=1, i\ne j}^{n} a_{ij}/2 \le C_E \\ & \sum_{j=1}^{n} a_{ij} \le \Delta_{\max i}, \quad v_i \in V \\ & \mathscr{L}(\mathcal{G}) + \mathbf{11}^T/n \succ 0 \\ & \begin{pmatrix} S & I_{n\times n} \\ I_{n\times n} & A(\mathcal{G},\mathcal{R}) \end{pmatrix} \succeq 0 \end{aligned}$$

We now apply the above MISDP formulation to the sparse swarming model (5.7) with 10 UAVs, two of which are acted on by wind gusts. The open loop $\mathcal{H}_2$ norm of the mapping from $q(t)$ to $y(t)$ is $\|G_{\mathcal{G},\mathcal{R}}(s)\|_2^2 = -\frac{1}{2}\mathbf{tr}\left(M(\mathcal{R})A\,(\mathcal{G},\mathcal{R})^{-1}\right)$ (Eq. 5.13). Further, to limit the relative sensing demands on the network, we make the following constraints on the network; that $|E| \le 25$, $|G|$ must be subgraph of the Euclidean-based constraint graph in Fig. 5.4, and each node has a maximum degree constraints also indicated in Fig. 5.4.

Applying these constraints to the MISDP formulation we acquire the graph in Fig. 5.5 and for comparison a random original graph that meets these constraints is also indicated. The metric $\|G_{\mathcal{G},\mathcal{R}}(s)\|_2^2$ for the optimized and comparative graphs are 1.89 and 2.07, respectively. The output of the UAV marked in Fig. 5.4 was compared in Fig. 5.6 with the constants of model (5.7) set as $L = 3.49$, $\sigma = 10$, $\bar{h} = 1$, and initial altitudes $x(0) = 0$ and $\bar{x}(0) = 0$.

Fig. 5.4 Edge constraint graph with maximum degree constraints indicated. The leader agents (those agents affected by wind gusts) are indicated with *squares* and the unique degree label corresponds to the agent whose dynamics are displayed in Fig. 5.6



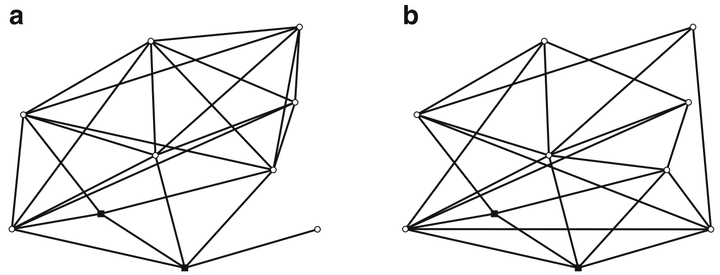


Fig. 5.5 (**a**) Original graph and (**b**) Optimized (adapted) graph both satisfying the edge and degree constraints from Fig. 5.4 with leader agents (*squares*)

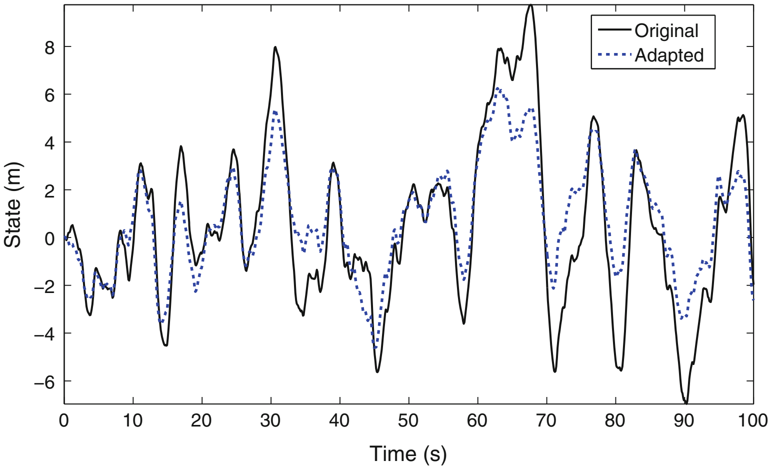


Fig. 5.6 The altitude state of the uniquely marked agent in Fig. 5.4 for the graphs in Fig. 5.5

5.4.2 *Adaptive Network Design and Dense Swarming*

It has previously been demonstrated that leader agents applying a *grounded signal* (all leaders apply a common constant signal) are able to control the mean consensus value of a wind gust disturbed system [34]. We now investigate controlling the consensus value of a wind gust disturbed system using a $\mathcal{H}_2$ control framework *assuming a nominal* $\mathcal{G}$ *in (5.9)*.

A consequence of Proposition 3.2 is that $A(\mathcal{G}, \mathcal{R})$ has only negative eigenvalues. Examining A_w, $\lambda_i(A_w) = -\frac{1}{L}$ for $i = 1, 2$ and as $L > 0$, it follows that A_w has only negative eigenvalues. Therefore the state matrix $\mathcal{A}_D$ is stabilizable and detectable so a stabilizing compensator can be formed.

Hence given a performance measure

$$J = \lim_{t \to \infty} \mathbb{E}\left\{x(t)^T Qx(t) + u(t)^T Ru(t)\right\},$$

a compensator that minimizes J can be solved of the form

$$\begin{aligned} \dot{\hat{x}}(t) &= \left(\mathcal{A}_D - \mathcal{B}_D K_c - K_f \mathcal{C}_D\right)\hat{x}(t) + K_f y(t) \\ u(t) &= K_c \hat{x}(t), \end{aligned}$$

where $\hat{x}(t)$ is an estimate of $\left[\, x(t)^T \; z(t)^T \,\right]^T$, $K_c \in \mathbb{R}^{r \times n}$ is the compensator gain and $K_f \in \mathbb{R}^{n \times n}$ is the filter gain. The compensator and filter gains satisfy $K_c = R^{-1}\mathcal{B}_D^T P$ and $K_f = S\mathcal{C}_D^T V^{-1}$ and P and S are the solutions of the algebraic Riccati Equations,

$$\begin{aligned} 0 &= \mathcal{A}_D^T P + P\mathcal{A}_D + Q - P\mathcal{B}_D R^{-1}\mathcal{B}_D^T P \\ 0 &= \mathcal{A}_D S + S\mathcal{A}_D^T + G_D W G_D^T - S\mathcal{C}_D^T V^{-1}\mathcal{C}_D S. \end{aligned}$$

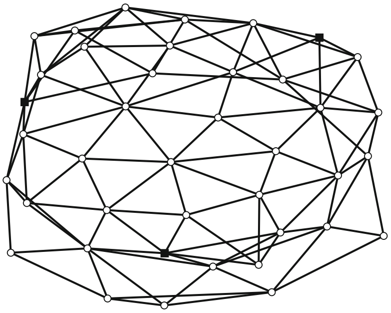
The corresponding optimal performance cost is then

$$J^* = \mathbf{tr}(PK_f VK_f^T) + \mathbf{tr}(SQ). \tag{5.27}$$

For the dense swarming model (5.10), the stabilizing compensator was applied to a 40-agent UAV network exposed to wind gusts composed of three leaders and 37 followers with parameters; $Q = I_{37\times 37}$, $R = 10I_{3\times 3}$, $V = 0.1I_{37\times 37}$, $H = \mathbf{1}$, $W = 1$, $L = 3.49$, $\sigma = 10$, and initial altitude set as $x(0) = 0$. The network topology is depicted in Fig. 5.7. The performance with respect to the average state and control with a desired altitude hover command ($x(t) = 0$) are compared to the grounded signal control in Fig. 5.8.

We now present a network design protocol to improve the $\mathcal{H}_2$ controller performance. We note that the previous formulation using mixed-integer design is no longer feasible as it equates to maximizing a convex function which is NP-hard.

Fig. 5.7 Original agent graph with leader agents (*squares*)



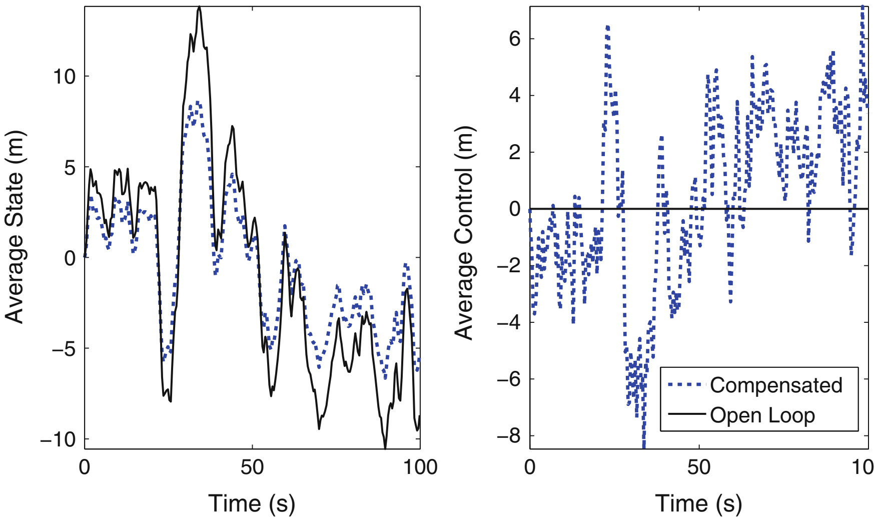


Fig. 5.8 Average state and control over time for the compensated (*closed loop* $\mathcal{H}_2$ controller) and grounded constant (*open loop*) control for the wind perturbed graph in Fig. 5.7

It is with this in mind that we propose an edge trading (rewiring), adaptive topology controller to improve the nominal $\mathcal{H}_2$ performance.

The protocol runs over the spanning trees of $\mathcal{G}$ with the objective of increasing $\|G_{\mathcal{G},\mathcal{R}}(s)\|_2^2$ via increasing $|\mathcal{E}_R|$ and $E_{\text{eff}}(v_i)$ for all $v_i \in \pi(\mathcal{E}_R)$ as illustrated through Lemma 5.8. The protocol involves edge trades between neighboring agents executed concurrently and/or in a random agent order while maintaining a connected tree at each iteration. The general procedure is to randomly select a spanning tree $\mathcal{T}$ of $\mathcal{G}$ and apply **Protocol 8** for some finite number of edge trades and then repeat with a new spanning tree. Within the protocol, we denote edge removal and addition by the set notation "$-/+$", and denote by $\mathcal{I}(v_i)$, the set of all agents that are neighbors of v_i and lie on the shortest path between v_i and any $r_j \in R$. In this direction, let us first define the special set of agents that lie on any of the shortest paths between agents in $\mathcal{R}$ as *main path agents*, i.e., those agents such that are leader or with $\mathcal{I}(v_i) > 1$. The protocol involves two conditions: one to increase the degree of

Protocol 8 Increasing $trP(\mathcal{T}, \mathcal{R})$ edge swap

```
foreach Agent $v_i$ do
    if $v_k = \mathcal{I}(v_i)$, $\exists v_j \in \mathcal{N}(v_i)$ and $v_j \neq v_k$ then
        $E \to E - \{v_i, v_j\} + \{v_j, v_k\}$
    end
    if $|\mathcal{I}(v_i)| > 1$, $\exists v_j, v_k \in \mathcal{N}(v_i)$, $v_j \in \mathcal{I}(v_i)$ and $v_k \notin \mathcal{I}(v_i)$ then
        $E \to E - \{v_i, v_j\} + \{v_j, v_k\}$
    end
end
```

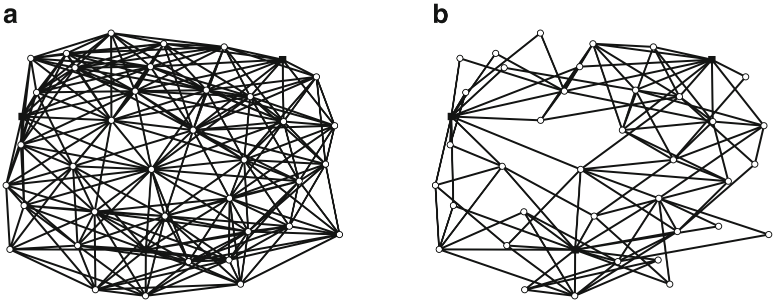


Fig. 5.9 (**a**) Edge constraint graph and (**b**) Resultant leader–follower network after applying **Protocol 8** to 50 spanning trees of the graph in Fig. 5.7. The leader agents are marked with *squares*

agents in R, thus increasing $|\mathcal{E}_R|$ and $\left\|G_{\mathcal{G},\tilde{\mathcal{R}}_c}(s)\right\|_2^2$, the other to increase the effective resistance for agents in set $\pi(\mathcal{E}_R)$ and thus increasing $\left\|G_{\mathcal{T},\tilde{\mathcal{R}}_d}(s)\right\|_2^2$. The following results are consequences of Lemma 3.17 and direct observation of the **Protocol 8**.

Lemma 5.13. *Under both conditions of* ***Protocol 8****,* $|\mathcal{E}_R|$ *only increases and the graph remains connected.*

Lemma 5.14. *For a tree* $\mathcal{T}$*, under the second conditions of* ***Protocol 8****,* $E_{\text{eff}}(v_i)$ for $v_i \in \pi(\mathcal{E}_R)$*, monotonically increases.*

Lemma 5.13 has the effect of compressing the network about the main path agents. On the other hand, Lemma 5.14 adds agents to the main path of $\mathcal{T}$ and in doing so elongates the main path.

We now apply the protocol to the dense swarming graph. We restrict the protocol to only edges within the constraint graph displayed in Fig. 5.9a and limit the maximum degree of the followers agents at 8. Running the protocol on the graph in Fig. 5.7 over 50 spanning trees, involving 289 edge trades, we produced the final graph displayed in Fig. 5.9b. The resultant leader–follower model has a $\|G_{\mathcal{G},\mathcal{R}}(s)\|_2^2 = 11.3$ compared to the original system with 4.6. The resultant graph and its response to wind gusts are displayed in Figs. 5.9 and 5.10. The optimal performance cost J^* as defined in Eq. (5.27) decreased from 14,815 to 3,265.

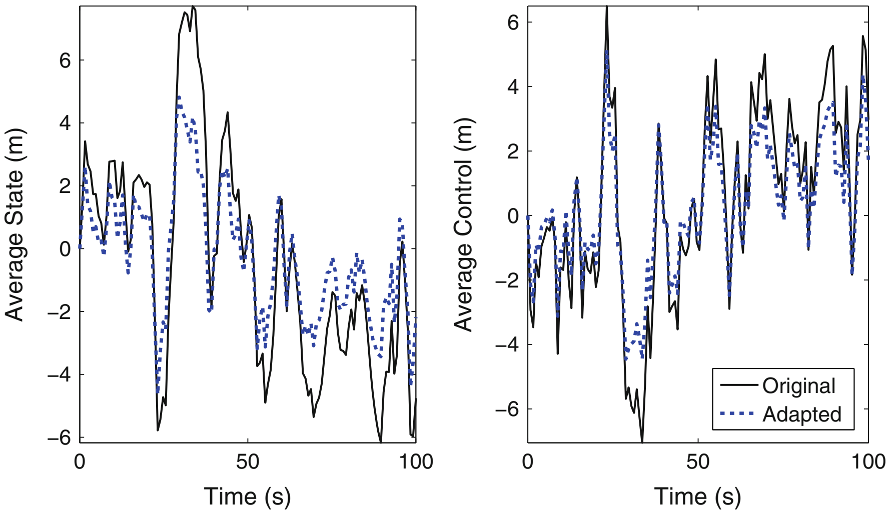


Fig. 5.10 Average state and control over time for the wind perturbed system corresponding to Fig. 5.7 before and after applying **Protocol 8**

5.5 Remarks

The main objective of the present chapter is to propose a network-theoretic approach for the efficient network topology design of leader–follower systems, specifically UAV swarming in the presence of wind gusts. The open loop $\mathcal{H}_2$ norm was analyzed as a metric to effectively quantify the performance of the underlying network. An MISDP design procedure was formulated accounting for sensing constraints relevant for a sparse swarming application. Similarly, an adaptive rewiring protocol based on the spanning trees of the graph was described for the use in the design of networks for dense swarming with UAV controllers. Both design methods were applied to UAV swarms with notable wind gust rejection improvements.

References

1. B. Bollobás, *Modern Graph Theory*. New York: Springer, 1998.
2. R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.
3. H. G. Tanner, G. J. Pappas, and V. Kumar, “Leader-to-formation stability,” *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 443–455, 2004.
4. A. Jadbabaie, J. Lin, and A. S. Morse, “Coordination of groups of mobile autonomous agents using nearest neighbor rules,” *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

5. M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton: Princeton University Press, 2010.
6. R. Olfati-Saber, “Flocking for multi-agent dynamic systems: algorithms and theory,” *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.
7. D. Zelazo and M. Mesbahi, “Edge agreement: graph-theoretic performance bounds and passivity analysis,” *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 544–555, Mar. 2011.
8. A. Ghosh, S. Boyd, and A. Saberi, “Minimizing effective resistance of a graph,” *SIAM Review*, vol. 50, no. 1, pp. 37–66, 2008.
9. Y. Wan, S. Roy, and A. Saberi, “Network design problems for controlling virus spread,” in *Proc. 46th IEEE Conference on Decision and Control*, 2007, pp. 3925–3932.
10. P. Barooah and J. P. Hespanha, “Graph effective resistance and distributed control: spectral properties and applications,” in *Proc. 45th IEEE Conference on Decision and Control*, 2006, pp. 3479–3485.
11. G. Tyson, A. T. Lindsay, S. Simpson, and D. Hutchison, “Improving wireless sensor network resilience with the INTERSECTION framework,” in *Proc. 2nd International Conference on Mobile Lightweight Wireless Systems, Critical Information Infrastructure Protection*, 2010.
12. S. Salsa, *Partial Differential Equations in Action: From Modelling to Theory*. New York: Springer, 2008.
13. T. J. Mueller, *Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications*. Danvers: AIAA, 2001.
14. D. J. Pines and F. Bohorquez, “Challenges facing future micro-air-vehicle development,” *Journal of Aircraft*, vol. 43, no. 2, pp. 290–305, 2006.
15. W. Shyy, Y. Lian, J. Tang, D. Viieru, and H. Liu, *Aerodynamics of Low Reynold Number Flyers*. New York: Cambridge University Press, 2007.
16. E. R. Ulrich, D. J. Pines, and J. S. Humbert, “Pitch and heave control of robotic Samara micro air vehicles,” *Journal of Aircraft*, vol. 47, no. 4, pp. 1290–1299, 2010.
17. S. Griffiths, J. Saunders, A. Curtis, B. Barber, T. McLain, and R. Beard, *Advances in Unmanned Aerial Vehicles: State of the art and the road to autonomy*, K. P. Valavanis, Ed. Dordrecht, The Netherlands: Springer Verlag, 2007.
18. W. Ren and R. W. Beard, *Distributed Consensus in Multi-Vehicle Cooperative Control*. New York: Springer Verlag, 2007.
19. J. P. How, B. Bethke, A. Frank, D. Dale, and V. J, “Real-time indoor autonomous vehicle test environment,” *IEEE Control Systems Magazine*, vol. 28, no. 2, pp. 51–64, 2008.
20. N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, “The GRASP multiple micro-UAV testbed,” *IEEE Robotics and Automation Magazine*, vol. 17, no. 3, pp. 56–65, 2010.
21. D. Mellinger and V. Kumar, “Minimum snap trajectory generation and control for quadrotors,” in *International Conferences on Robotics and Automation*, 2011, pp. 2520–2525.
22. L. Mutuel and R. Douglas, “Controller design for low speed flight in turbulence,” in *AIAA Guidance, Navigation, and Control Conference*, 1997, pp. 1111–1121.
23. B. L. Stevens and F. L. Lewis, *Aircraft Control and Simulation*. Hoboken: Wiley, 2003.
24. J. Hall, “Lateral control and observation of a micro aerial vehicle,” in *45th AIAA Aerospace Sciences Meeting and Exhibit*, 2007, pp. 8–11.
25. W. Shyy, Y. Lian, J. Tang, H. Liu, P. Trizila, B. Stanford, L. Bernal, C. Cesnik, P. Friedmann, and P. Ifju, “Computational aerodynamics of low Reynolds number plunging, pitching and flexible wings for MAV applications,” in *46th AIAA Aerospace Sciences Meeting and Exhibit*, vol. 24, no. 4, Jul. 2008, pp. 1–33.
26. T. Cheviron, “Robust control of an autonomous reduced scale helicopter in presence of wind gusts,” in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006, pp. 1–22.
27. C.-d. Yang, W.-h. Liu, P.-w. Chang, and H.-j. Weng, “Decoupling control for hovering flight vehicle with parameter uncertainties,” in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, no. 1, 2006, pp. 1–13.

28. N. Meskin, K. Khorasani, and C. A. Rabbath, “A hybrid fault detection and isolation strategy for a network of unmanned vehicles in presence of large environmental disturbances,” *IEEE Transactions on Control Systems Technology*, no. August, Nov. 2010.
29. X. Li and L. Cao, “Largest Laplacian eigenvalue predicts the emergence of costly punishment in the evolutionary ultimatum game on networks,” *Physical Review E - Statistical, Nonlinear and Soft Matter Physics, 80(6 Pt 2)*, vol. 80, p. 066101, 2009.
30. A. Ghosh and S. Boyd, “Growing well-connected graphs,” in *Proc. 45th IEEE Conference on Decision and Control*. IEEE, 2006, pp. 6605–6611.
31. J. Lofberg, “YALMIP : a toolbox for modeling and optimization in MATLAB,” in *2004 IEEE International Symposium*, 2004, pp. 284–289.
32. Y. Kim and M. Mesbahi, “On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian,” *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 116–120, Jan. 2006.
33. Z. Wu and R. Wang, “The consensus in multi-agent system with speed-optimized network,” *International Journal of Modern Physics B*, vol. 23, no. 10, pp. 2339–2348, 2009.
34. A. Chapman and M. Mesbahi, “Semi-autonomous consensus: network measures and adaptive trees,” *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 19–31, 2013.
35. A. Chapman and M. Mesbahi, “Semi-autonomous networks: network resilience and adaptive trees,” in *Proc. 49th IEEE Conference on Decision and Control*, no. 2, 2010, pp. 7473–7478.

Part III
Cartesian Product Networks

Chapter 6
Cartesian Products of Z-matrix Networks: Factorization and Interval Analysis

Abstract This chapter examines the relationship between the dynamics of large networks and of their smaller factor networks (factors) obtained through the factorization of the network's digraph representation. We specifically examine dynamics of networks which have Z-matrix state matrices. We perform a Cartesian product decomposition on its network structure producing factors which also have Z-matrix dynamics. A factorization lemma is presented that represents the trajectories of the large network in terms of the factors' trajectories. An interval matrix lemma provides families of network dynamics whose trajectories are bounded by the interval bounds' factors' trajectories.

6.1 Introduction

Complex dynamic networks are an integral part of the technological world with examples including the Internet, power grids, and communication networks, as well as in the world at large such as biological and chemical systems and social networks. An explosion of research in the area of network systems has eventuated [6–8]. In parallel, Z-matrices are used to model synchronization in networks [3], population migration [4], Markov processes, and supply and demand in economic systems [5].

Z-matrices also appear in the discretization of differential operators. An example is the discretization of diffusion [3] and advection [16] dynamics which generate the in-degree and out-degree Laplacian matrix respectively, and both of which are Z-matrices. The in-degree Laplacian matrix forms the basis of consensus models. These models are effective for both distributed information-sharing and control of networked, multi-agent systems in settings such as multi-vehicle control, formation control, swarming, and distributed estimation; see, for example, [1, 2].

In this work we examine large networks which are Cartesian products of smaller factor networks (factors). We present a factorization lemma which represents the larger network trajectories in terms of the factors' trajectories, provided certain initial conditions are met. This result is an extension of the related results on Cartesian products over Laplacian-based simple networks for a constrained set of initial conditions by Nguyen and Mesbahi [9]. We extend the factorization lemma to both non-decomposable networks and arbitrary nonnegative initial conditions, bounding the large network trajectories with the factors' trajectories.

A. Chapman, *Semi-Autonomous Networks*, Springer Theses,
DOI 10.1007/978-3-319-15010-9_6

The organization of the chapter is as follows. We begin by introducing relevant background material pertaining to Cartesian products and Kronecker products. We introduce the Z-matrix state dynamics. Interval matrices are used to introduce families of Z-matrices with similar trajectories. The Cartesian product over Z-matrices is introduced as a method to decompose large Z-matrix dynamics to smaller Z-matrix factor dynamics. The chapter culminates in the presentation of two factorization lemmas. The first lemma allows perfect characterization of the larger network trajectories in terms of the factors' unforced and forced trajectories. The second lemma presents a family of larger network trajectories which are bounded by factors' trajectories.

6.2 Cartesian Product

There is an abundance of effective methods via which large-scale networks (digraphs) can be synthesized from a set of smaller digraphs [10]. The Cartesian product is one such method and is defined for a pair of *factor* digraphs $\mathcal{D}_1 = (V_1, E_1, W_1)$ and $\mathcal{D}_2 = (V_2, E_2, W_2)$ and denoted by $\mathcal{D} = \mathcal{D}_1 \square \mathcal{D}_2$. The product digraph $\mathcal{D}$ has the vertex set $V_1 \times V_2$ and there is an edge from vertex (i, p) to (j, q) in $V_1 \times V_2$ if and only if either $i = j$ and (p, q) is an edge of E_2, or $p = q$ and (u, v) is an edge of E_1. The corresponding weight if an edge exists is $w_{((i,p),(j,q))} = w_{ij}^{\delta_{pq}} + w_{pq}^{\delta_{ij}}$ where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise. The Cartesian product is commutative and associative, i.e., the products $\mathcal{D}_1 \square \mathcal{D}_2$ and $\mathcal{D}_2 \square \mathcal{D}_1$ are isomorphic; similarly, $(\mathcal{D}_1 \square \mathcal{D}_2) \square \mathcal{D}_3$ and $\mathcal{D}_1 \square (\mathcal{D}_2 \square \mathcal{D}_3)$ are isomorphic.

An example of the Cartesian product of two factor digraphs is displayed in Fig. 6.1a–c.

A digraph is called *prime* if it cannot be decomposed into the product of non-trivial digraphs, otherwise a digraph is referred to as composite. Sabidussi [11] and Vizing [12] highlighted the fundamental nature of the primes, and noted that connected digraphs decompose uniquely into primes, up to reordering, specifically of the form $\mathcal{D}_1^{k_1} \square \cdots \square \mathcal{D}_m^{k_m}$, where $\mathcal{D}_i$ is prime for all i and $\mathcal{D}_i^{k_i}$ denotes k_i Cartesian products of $\mathcal{D}_i$. Further, Feigenbaum [13] demonstrated that a digraph can be factored into primes in polynomial-time.

Many features of the factors of a composite digraph transfer to the composite digraph itself. One such example is that if factors $\mathcal{D}_1$ and $\mathcal{D}_2$ are strongly connected, then so too is $\mathcal{D}_1 \square \mathcal{D}_2$. In this chapter we show that when the composite digraph underlies a dynamic system that many useful features of dynamics can be revealed by examining dynamics systems over the factor digraphs.

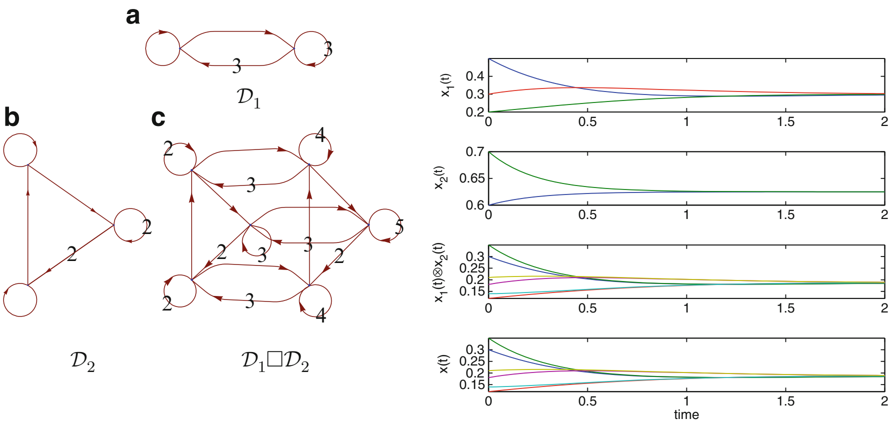


Fig. 6.1 *Left*: Factor digraphs (**a**)$\mathcal{D}_1$ and (**b**)$\mathcal{D}_2$ and composite digraph (**c**)$\mathcal{D}_1\Box\mathcal{D}_2$. Edge weights of all digraphs are 1 unless otherwise marked. *Right*: Trajectories of the Z-matrix dynamics over $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_1\Box\mathcal{D}_2$ with unforced state dynamics $x_1(t)$, $x_2(t)$ and $x(t)$, respectively, compared to $x_1(t) \otimes x_2(t)$; for details see Lemma 6.6

6.2.1 Kronecker Sum Eigenspectrum

Of particular interest to this chapter is the eigenspectrum properties of the Kronecker sum $A \oplus B$ on square matrices $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{m\times m}$. A property of the Kronecker sum is that given the left eigenvalue–eigenvector pairs of A and B as (λ_i, u_i) for $i = 1, \ldots, n$ and $\left(\mu_j, v_j\right)$ for $j = 1, \ldots, m$, respectively, then $(\lambda_i + \mu_j, u_i \otimes v_j)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, m$ are left eigenvalue–eigenvector pairs of $A \oplus B$. If A and B are diagonalizable matrices, their eigenvectors span $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively. Consequently, $u_i \otimes v_j$, for all i and j form a spanning set of eigenvectors of $A \oplus B$ as $\mathbf{rank}(U \otimes V) = \mathbf{rank}(U)\mathbf{rank}(V)$, for $U = [u_1, \ldots, u_n]$ and $V = [v_1, \ldots, v_m]$. Thus, if $\lambda_i + \mu_j$ is simple, A and B are diagonalizable and $u_i \otimes v_j$ is a left eigenvector if and only if (λ_i, u_i) and (μ_i, v_j) are left eigenvalue–eigenvectors of A and B, respectively.

We now proceed to introduce Z-matrix dynamics and their underlying digraph.

6.3 Z-matrix Dynamics

We consider a multi-agent network of n coupled nodes with the state of each node i defined as $x_i(t) \in \mathbb{R}$ at time t, and driven by a control $u(t) \in \mathbb{R}^m$. The system is described by the differential equations

$$\dot{x}_i(t) = -w_{ii}x_i(t) + \sum_{j\in\mathcal{N}(i)} w_{ij}x_j(t) + b_i^T u(t),\ i = 1, \ldots, n.$$

The unforced and forced components of the signal $x(t)$ are denoted by $x_u(t)$ and $x_f(t)$, respectively. This notation will be used throughout. In a more compact form the dynamics can be written as

$$\dot{x}(t) = -A(\mathcal{D})x(t) + Bu(t), \tag{6.1}$$

where $x(t) = [x_1(t), \ldots, x_n(t)]^T \in \mathbb{R}^n$, $B = [b_1, \ldots, b_n]^T \in \mathbb{R}^{n\times m}$, and the matrix representation of $\mathcal{G}$ is defined as $A(\mathcal{D}) := 2\Delta_s(\mathcal{D}) - \mathscr{A}(\mathcal{D})$, i.e.,

$$A(\mathcal{D}) = \begin{bmatrix} w_{11} & -w_{12} & \cdots & -w_{1n} \\ -w_{21} & w_{22} & -w_{23} & \vdots \\ \vdots & & \ddots & -w_{n-1,n} \\ -w_{n1} & \cdots & -w_{n,n-1} & w_{nn} \end{bmatrix}.$$

In this way, the matrix $A(\mathcal{D})$, as with the adjacency matrix, codifies the interconnections between nodes.

We define the class of matrices of this form as

$$Z_n = \left\{A = \left(a_{ij}\right) \in \mathbb{R}^{n\times n} : a_{ii} \geq 0, a_{ij} \leq 0, i \neq j\right\}.$$

Z-matrices are matrices with nonpositive off-diagonals, hence Z_n is a subclass of Z-Matrices, motivating the name Z-Matrix dynamics for model (6.1).

The negation of Z-matrices falls into the class of *essentially nonnegative matrices*. These are matrices which have positive off-diagonals and bounded diagonals. This connection facilitates establishing positive invariance of the positive set $S^+ = \{x(t)|x(t) \geq 0\}$ with respect to model (6.1) when $A \in Z_n$. The result that establishes positive invariance and that will be used in the subsequent chapter is as follows.

Proposition 6.1. *Let A and B be essentially nonnegative matrices with $A \geq B$; that is $A + sI \geq 0$ and $B + sI \geq 0$ for all real s sufficiently large. For all $t \geq 0$, $e^{tA} \geq e^{tB} \geq 0$.*

Proof. Let s be sufficiently large such that $A + sI \geq 0$ and $B + sI \geq 0$, then $(A + sI)^j \geq (B + sI)^j \geq 0$ for all positive integer j. Hence,

$$e^{tB} = e^{t(B+sI)} = e^{-ts}\sum_{j=0}^{\infty}\frac{(t\,(B + sI))^j}{j!} \geq 0.$$

Further,

$$e^{tA} = e^{-ts}\sum_{j=0}^{\infty}\frac{(t\,(A + sI))^j}{j!} \geq e^{-ts}\sum_{j=0}^{\infty}\frac{(t\,(B + sI))^j}{j!} = e^{tB}. \qquad \square$$

We will now introduce families of dynamics with similar attributes via the construct of interval matrices.

6.4 Interval Matrices

We commence by defining interval matrices.

Definition 6.2. If $\underline{A}$ and $\overline{A}$ are two matrices in $\mathbb{R}^{n\times m}$ with $\underline{A} \leq \overline{A}$, then the set of matrices

$$\mathbb{A} = \left[\underline{A}, \overline{A}\right] = \left\{A : \underline{A} \leq A \leq \overline{A}\right\}$$

is called an interval matrix, and the matrices $\overline{A}$ and $\underline{A}$ are called its bounds. Further, this interval is referred to as symmetric if $\overline{A}$ and $\underline{A}$ are symmetric.

Hence $A \in \mathbb{A}$ if $[\underline{A}]_{ij} \leq [A]_{ij} \leq [\overline{A}]_{ij}$ for all $i = 1, \ldots, n$ and $j = 1, \ldots, m$. We emphasize that each entry in A can vary arbitrarily in its interval independent of the other entries in A. A symmetric interval matrix can also contain asymmetric matrices.

The examination of interval matrices arises naturally in control theory in connection with the behavior of linear time invariant systems under perturbations, and has been extensively studied. We refer the reader to the survey chapters by Mansour [14] and Rohn [15] for a detailed list of references.

When the matrix intervals bounds are Z-matrices, then all matrices contained in the interval are Z-matrices. A similar result is true for our subclass of matrix Z_n, this is stated formally without proof in the following proposition:

Proposition 6.3. *Consider the matrices in the class Z_n, $A(\underline{\mathcal{D}})$ and $A(\overline{\mathcal{D}})$ with $A(\underline{\mathcal{D}}) \leq A(\overline{\mathcal{D}})$ corresponding to n-node digraphs $\underline{\mathcal{D}}$ and $\overline{\mathcal{D}}$. Every matrix in the matrix interval $\mathbb{A} = \left[A(\underline{\mathcal{D}}), A(\overline{\mathcal{D}})\right]$ is in the class Z_n.*

The motivation for using matrix interval's is that Z-matrices exhibit many useful additive ordering properties and $A(\mathcal{D}) \in \left[A(\underline{\mathcal{D}}), A(\overline{\mathcal{D}})\right]$ implies that $\mathcal{G} \in \left[\underline{\mathcal{D}}, \overline{\mathcal{D}}\right]$ if and only if self-loops and their weights in $\mathcal{D}$, $\underline{\mathcal{D}}$, and $\overline{\mathcal{D}}$ are the same. The advantage of Z-matrix intervals is that trajectory bounds can be provided for dynamic systems defined by matrices contained in a Z-matrix interval matrix, as shown in the following:

Proposition 6.4. *Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be finite digraphs. Consider $\underline{x}(t)$ and $\overline{x}(t)$ to be the respective states of the systems*

$$\begin{aligned}\dot{\underline{x}}(t) &= -A(\underline{\mathcal{D}})\underline{x}(t) + \underline{Bu}(t)\\ \dot{\overline{x}}(t) &= -A(\overline{\mathcal{D}})\overline{x}(t) + \overline{B}\overline{u}(t).\end{aligned}$$

Then, the state trajectory generated by the dynamics

$$\dot{x}(t) = -Ax(t) + Bu(t),$$

for $A \in \mathbb{A} = \left[A(\underline{\mathcal{D}}), A(\overline{\mathcal{D}})\right]$, $B \in \mathbb{B} = \left[\overline{B}, \underline{B}\right]$ *and* $0 \leq \overline{u}(t) \leq u(t) \leq \underline{u}(t)$ *is bounded as*

$$\overline{x}(t) \leq x(t) \leq \underline{x}(t),$$

for all t *when initialized from* $0 \leq \overline{x}(0) \leq x(0) \leq \underline{x}(0)$.

Proof. For every $A \in \mathbb{A}$, $A(\underline{\mathcal{D}}) \leq A$. From Proposition 6.1, the unforced dynamics is bounded as

$$x_u(t) = e^{-At}x(0) \geq e^{-A(\underline{\mathcal{D}})t}x(0) \geq e^{-A(\underline{\mathcal{D}})t}\overline{x}(0) = \overline{x}_u(t).$$

Similarly, $x_u(t) \leq \underline{x}_u(t)$. For $t \geq \tau \geq 0$, from Proposition 6.1, the forced dynamics is bounded as

$$\begin{aligned}
e^{-A(t-\tau)}Bu(\tau) &\geq e^{-A(\underline{\mathcal{D}})(t-\tau)}Bu(\tau) \\
&\geq e^{-A(\mathcal{D})(t-\tau)}\overline{B}\overline{u}(\tau) \\
\int_0^t e^{-A(t-\tau)}Bu(\tau)d\tau &\geq \int_0^t e^{-A(\underline{\mathcal{D}})(t-\tau)}\overline{B}\overline{u}(\tau)d\tau \\
x_f(t) &\geq \overline{x}_f(t).
\end{aligned}$$

Similarly, $x_f(t) \leq \underline{x}_f(t)$. Noting that the dynamics are formed by the sum of its unforced and forced dynamics, the proposition follows. □

The following section analyzes the Z-matrix dynamics and Z-matrix intervals dynamics formed from applying Cartesian products to digraphs.

6.5 Z-matrix Dynamics Over Cartesian Products of Digraphs

The Cartesian product over digraphs can be formulated in terms of their Z-matrix representations using the Kronecker sum.

Proposition 6.5. *Let* $\mathcal{D}_1$ *and* $\mathcal{D}_2$ *be a pair of digraphs of order* n *and* m*, respectively. Then* $A(\mathcal{D}_1 \square \mathcal{D}_2) = A(\mathcal{D}_1) \oplus A(\mathcal{D}_2)$.

Proof. The proposition follows directly from the definition of the digraph product and the Z-matrix realization of a digraph. □

We now present a result which we refer to as the *factorization lemma* for Z-matrix dynamics.

Lemma 6.6 (Factorization). *Consider $x_1(t)$ and $x_2(t)$ to be the respective states of the systems*

$$\begin{aligned}\dot{x}_1(t) &= -A(\mathcal{D}_1)x_1(t) + B_1u_1(t)\\ \dot{x}_2(t) &= -A(\mathcal{D}_2)x_2(t) + B_2u_2(t),\end{aligned}$$

for all time t. Then, the unforced state trajectory generated by the dynamics

$$\dot{x}(t) = -A(\mathcal{D}_1\square\mathcal{D}_2)x(t) + (B_1\otimes B_2)\,(u_1(t)\otimes u_2(t))$$

is

$$x_u(t) = x_{1u}(t)\otimes x_{2u}(t),$$

and the forced state trajectory is

$$x_f(t) = \int_0^t \dot{x}_{1f}(\tau)\otimes\dot{x}_{2f}(\tau)d\tau,$$

for all time t and with initial conditions $x(0) = x_1(0)\otimes x_2(0)$.

Proof. From Proposition 6.5, examining the unforced dynamics we have

$$\begin{aligned}x_u(t) &= e^{-A(\mathcal{D}_1\square\mathcal{D}_2)t}x(0)\\ &= e^{-A(\mathcal{D}_1)t\oplus -A(\mathcal{D}_2)t}(x_1(0)\otimes x_2(0))\\ &= \left(e^{-A(\mathcal{D}_1)t}\otimes e^{-A(\mathcal{D}_2)t}\right)(x_1(0)\otimes x_2(0))\\ &= e^{-A(\mathcal{D}_1)t}x_1(0)\otimes e^{-A(\mathcal{D}_2)t}x_2(0)\\ &= x_{1u}(t)\otimes x_{2u}(t).\end{aligned}$$

Examining the forced dynamics, we have

$$\begin{aligned}x_f(t) &= \int_0^t e^{-A(\mathcal{D}_1\square\mathcal{D}_2)(t-\tau)}(B_1\otimes B_2)(u_1(\tau)\otimes u_2(\tau))d\tau\\ &= \int_0^t \left(e^{-A(\mathcal{D}_1)(t-\tau)}\otimes e^{-A(\mathcal{D}_2)(t-\tau)}\right)(B_1\otimes B_2)(u_1(\tau)\otimes u_2(\tau))d\tau\\ &= \int_0^t z_1(\tau)\otimes z_2(\tau)d\tau,\end{aligned}$$

where $z_1(\tau) = e^{-A(\mathcal{D}_1)(t-\tau)}B_1u_1(\tau)$ and $z_2(\tau) = e^{-A(\mathcal{D}_2)(t-\tau)}B_2u_2(\tau)$. Noting that $\dot{x}_{if}(\tau) = \frac{d}{d\tau}\int_0^\tau z_i(\bar{t})d\bar{t} = z_i(\tau)$, for $i = 1, 2$, the lemma follows. □

Figure 6.1 displays an example the unforced trajectories in Lemma 6.6. Note that, due to the associativity of the Cartesian product, the result extends to arbitrary chains of Cartesian products.

Observing that if $A(\underline{\mathcal{D}}_1) \leq A(\overline{\mathcal{D}}_1)$ and $A(\underline{\mathcal{D}}_2) \leq A(\overline{\mathcal{D}}_2)$ then $A(\underline{\mathcal{D}}_1 \square \underline{\mathcal{D}}_2) \leq A(\overline{\mathcal{D}}_1 \square \overline{\mathcal{D}}_2)$, we can define a Z-matrix interval $\left[A(\underline{\mathcal{D}}_1 \square \underline{\mathcal{D}}_2), A(\overline{\mathcal{D}}_1 \square \overline{\mathcal{D}}_2)\right]$ with composite interval matrix bounds. The following lemma shows that trajectories of dynamics defined by Z-matrices in this interval are bounded above and below by the trajectories of dynamics defined by the factor Z-matrices $A(\underline{\mathcal{D}}_1)$, $A(\underline{\mathcal{D}}_2)$, $A(\overline{\mathcal{D}}_1)$ and $A(\overline{\mathcal{D}}_2)$.

Lemma 6.7 (Interval Factorization). *Consider $\underline{x}_1(t), \overline{x}_1(t), \underline{x_2}(t)$ and $\overline{x}_2(t)$ to be the respective states of the systems*

$$
\begin{aligned}
\dot{\underline{x}}_1(t) &= -A(\underline{\mathcal{D}}_1)\underline{x}_1(t) + \underline{B}_1\underline{u}_1(t)\\
\dot{\overline{x}}_1(t) &= -A(\overline{\mathcal{D}}_1)\overline{x}_1(t) + \overline{B}_1\overline{u}_1(t)\\
\dot{\underline{x}}_2(t) &= -A(\underline{\mathcal{D}}_2)\underline{x}_2(t) + \underline{B}_2\underline{u}_2(t)\\
\dot{\overline{x}}_2(t) &= -A(\overline{\mathcal{D}}_2)\overline{x}_2(t) + \overline{B}_2\overline{u}_2(t).
\end{aligned}
$$

Then, the unforced state trajectory generated by the dynamics

$$
\dot{x}(t) = -Ax(t) + Bu(t),
$$

for $A \in \mathbb{A} = \left[A(\underline{\mathcal{D}}_1 \square \underline{\mathcal{D}}_2), A(\overline{\mathcal{D}}_1 \square \overline{\mathcal{D}}_2)\right]$, $B \in \mathbb{B} = \left[\overline{B}_1 \otimes \overline{B}_2, \underline{B}_1 \otimes \underline{B}_2\right] \geq 0$, $\overline{u}_1(t) \otimes \overline{u}_2(t) \leq u(t) \leq \underline{u}_1(t) \otimes \underline{u}_2(t)$ and $0 \leq \overline{u}_i(t) \leq \underline{u}_i(t)$ for $i = 1, 2$ is bounded as

$$
\overline{x}_{1u}(t) \otimes \overline{x}_{2u}(t) \leq x_u(t) \leq \underline{x}_{1u}(t) \otimes \underline{x}_{2u}(t),
$$

and the forced state trajectory is bounded as

$$
\int_0^t \dot{\overline{x}}_{1f}(\tau) \otimes \dot{\overline{x}}_{2f}(\tau) d\tau \leq x_f(t) \leq \int_0^t \dot{\underline{x}}_{1f}(\tau) \otimes \dot{\underline{x}}_{2f}(\tau) d\tau,
$$

for all time t with initial conditions

$$
\overline{x}_1(0) \otimes \overline{x}_2(0) \leq x(0) \leq \underline{x}_1(0) \otimes \underline{x}_2(0),
$$

and $0 \leq \overline{x}_i(t) \leq \underline{x}_i(t)$ for $i = 1, 2$.

Proof. Consider the dynamics

$$
\begin{aligned}
\dot{\underline{z}}(t) &= -A(\underline{\mathcal{D}}_1 \square \underline{\mathcal{D}}_2)\underline{z}(t) + (\underline{B}_1 \otimes \underline{B}_2)\left(\underline{u}_1(t) \otimes \underline{u}_2(t)\right)\\
\dot{\overline{z}}(t) &= -A(\overline{\mathcal{D}}_1 \square \overline{\mathcal{D}}_2)\overline{z}(t) + \left(\overline{B}_1 \otimes \overline{B}_2\right)(\overline{u}_1(t) \otimes \overline{u}_2(t))\\
\dot{x}(t) &= -Ax(t) + Bu(t)
\end{aligned}
$$

where $0 \le \overline{z}(0) \le x(0) \le \underline{z}(0)$. From Proposition 6.4, $\overline{z}(t) \le x(t) \le \underline{z}(t)$ for all $t > 0$. Further letting $\overline{z}(0) = \overline{x}_1(0) \otimes \overline{x}_2(0)$ and $\underline{z}(0) = \underline{x}_1(0) \otimes \underline{x}_2(0)$ using Lemma 6.6, the lemma follows. □

The significance of this result is that digraph-based dynamics need not be composite to take advantage of the factorization lemma. Indeed its Z-matrix digraph representation need only be bounded by the Z-matrix representations of its composite digraphs. Further, the factor digraph dynamics provides bounds on nonnegative composite dynamics independent of where the composite trajectory is initialized, i.e., $x(0)$ can be chosen anywhere in $(\mathbb{R}^{nm})_{\ge 0}$ rather than $\mathbb{R}^n \otimes \mathbb{R}^m \cong \mathbb{R}^{n+m}$ as for Lemma 6.6.

Figure 6.2 displays sample digraphs $\underline{\mathcal{D}}_1$, $\overline{\mathcal{D}}_1$, $\underline{\mathcal{D}}_2$, $\overline{\mathcal{D}}_2$, $\underline{\mathcal{D}}_1 \square \underline{\mathcal{D}}_2$, $\overline{\mathcal{D}}_1 \square \overline{\mathcal{D}}_2$, and $\mathcal{D}$ pertaining to Lemma 6.7. The related trajectories of the 16 states are in Fig. 6.3.

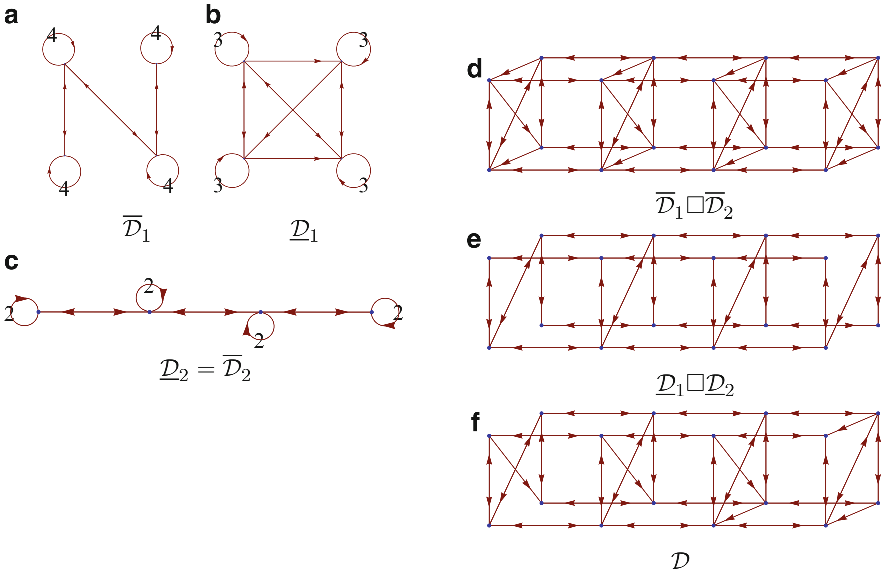

a
4
4
4
4
$\overline{\mathcal{D}}_1$
b
3
3
3
3
$\underline{\mathcal{D}}_1$
c
2
2
2
2
$\underline{\mathcal{D}}_2 = \overline{\mathcal{D}}_2$
d
$\overline{\mathcal{D}}_1 \square \overline{\mathcal{D}}_2$
e
$\underline{\mathcal{D}}_1 \square \underline{\mathcal{D}}_2$
f
$\mathcal{D}$

Fig. 6.2 The factor digraphs (**a**) $\overline{\mathcal{D}}_1$, (**b**) $\underline{\mathcal{D}}_1$ and (**c**) $\underline{\mathcal{D}}_2 = \overline{\mathcal{D}}_2$. The composite digraphs $\overline{\mathcal{D}}_1 \square \overline{\mathcal{D}}_2$ and $\underline{\mathcal{D}}_1 \square \underline{\mathcal{D}}_2$ and digraph $\mathcal{D}$ where $A(\mathcal{D}) \in \left[A(\underline{\mathcal{D}}_1 \square \underline{\mathcal{D}}_2), A(\overline{\mathcal{D}}_1 \square \overline{\mathcal{D}}_2)\right]$. For clarity, self-loops are unmarked for the digraphs (**d**), (**e**) and (**f**) and have weight 5 in $\overline{\mathcal{D}}_1 \square \overline{\mathcal{D}}_2$ and $\mathcal{D}$ and weight 6 in $\underline{\mathcal{D}}_1 \square \underline{\mathcal{D}}_2$. All other edges have weight 1

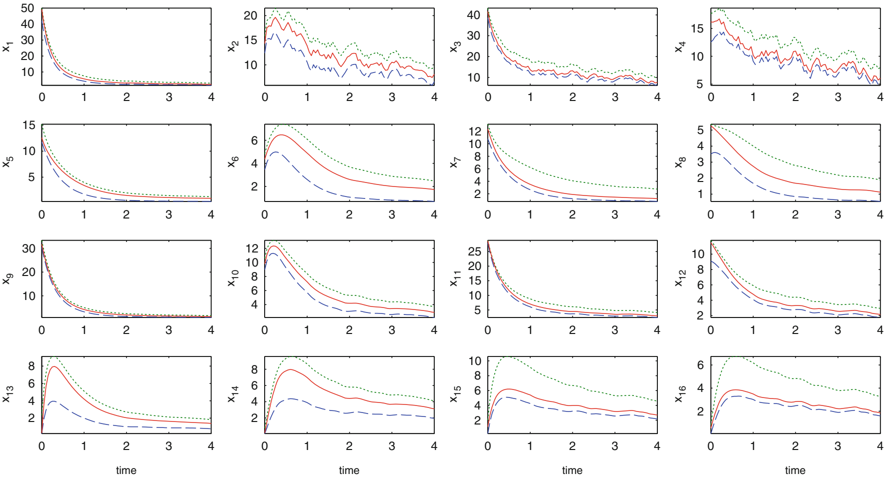


Fig. 6.3 Trajectories $\overline{x}_{1u}(t) \otimes \overline{x}_{2u}(t) + \int_0^t \dot{\overline{x}}_{1f}(\tau) \otimes \dot{\overline{x}}_{2f}(\tau)d\tau$ (*blue/dashed*), $x(t)$ (*red/solid*), $\underline{x}_{1u}(t) \otimes \underline{x}_{2u}(t) + \int_0^t \dot{\underline{x}}_{1f}(\tau) \otimes \dot{\underline{x}}_{2f}(\tau)d\tau$ (*green/dotted*) with underlying digraph structure in Fig. 6.2. The control matrices are $\overline{B}_1 = \underline{B}_1 = e_1$, $\overline{B}_2 = \underline{B}_2 = [e_2, e_3 + e_4]$ and $B = \overline{B}_1 \otimes \overline{B}_2$. The controls are positive random signals satisfying the ordering requirement of Lemma 6.7

6.6 Remarks

This chapter presents an analysis for a class of dynamic networks involving Z-matrices. We explored the decomposition of such networks into smaller factor-networks. The trajectories of the composite network were generated from the factors' trajectories. Also, families of networks similar to the composite network were bounded by the factors' trajectories.

References

1. R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.
2. H. G. Tanner, G. J. Pappas, and V. Kumar, "Leader-to-formation stability," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 443–455, 2004.
3. M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton: Princeton University Press, 2010.
4. R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York: Cambridge University Press, 1990.
5. A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*. Academic Press, 1979.
6. S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. Hwang, "Complex networks: structure and dynamics," *Physics Reports*, vol. 424, no. 4–5, pp. 175–308, 2006.
7. N. Ganguly, A. Deutsch, and A. Mukherjee, *Dynamics on and of Complex Networks: Applications to Biology, Computer Science, and the Social Sciences*. Boston: Birkhauser, 2009.

8. L. Kocarev and G. Vattay, *Complex Dynamics in Communication Networks*. Berlin: Springer, 2005.
9. A. Nguyen and M. Mesbahi, "A factorization lemma for the agreement dynamics," in *46th IEEE Conference on Decision and Control*, no. 1, 2007, pp. 288–293.
10. W. Imrich and S. Klavzar, *Product Graphs: Structure and Recognition*. New York: Wiley, 2000.
11. G. Sabidussi, "Graph multiplication," *Mathematische Zeitschrift*, vol. 72, pp. 446–457, 1960.
12. V. G. Vizing, "The Cartesian product of graphs," *Vycisl. Sistemy*, pp. 30–43, 1963.
13. J. Feigenbaum, "Directed graphs have unique Cartesian factorizations that can be found in polynomial time," *Discrete Applied Mathematics*, vol. 15, pp. 105–110, 1986.
14. M. Mansour, "Robust stability of interval matrices," in *Proc. 28th IEEE Conference on Decision and Control*, 1989, pp. 46–51.
15. J. Rohn, "Forty necessary and sufficient conditions for regularity of interval matrices: a survey," *Electronic Journal Of Linear Algebra*, vol. 18, pp. 500–512, 2009.
16. A. Chapman and M. Mesbahi, "Advection on graphs," in *Proc. 50th IEEE Conference on Decision and Control*, 2011, pp. 1461–1466.

Chapter 7
Controllability and Observability of Cartesian Product Networks

Abstract The chapter presents an analysis framework for a class of dynamic composite networks. These networks are formed from smaller factor networks via graph Cartesian products. We provide a composition method for extending the controllability and observability of the factor networks to that of the composite network. We then delve into the effectiveness of designing control and estimation algorithms for the composite network via symmetry in the network. Examples and applications are provided throughout the chapter to demonstrate the results including a distributed output feedback stabilizer and a social network application.

7.1 Introduction

Network controllability and observability arises in situations where a networked system is influenced or observed by an external entity, such scenarios include networked robotic systems, human–swarm interaction, and network security [1, 14], as well as in areas such as quantum networks [3].

In this direction, Godsil [2] made an intriguing conjecture regarding the dynamics driven by the adjacency matrix of a graph. The conjecture states that the ratio of graphs that are uncontrollable, with a common input to all nodes, to the total number of graphs of the same order tends to zero as the order of the graph increases. On another front, extensive simulations have demonstrated that it is "unlikely" that single leader-follower Laplacian-based consensus networks are controllable [4]. Together these observations imply that establishing network controllability is nontrivial and may be strongly dependent on the manifestation of the graph embedded in the dynamics.

Controllability for Laplacian networks has been established for special classes of graphs such as paths, circulants, grids, random and distance regular graphs [1, 5–8], but to the authors' knowledge, no other large-scale networks have been investigated. Liu et al. in [9] studied the structural controllability of complex networks.

In this chapter, we consider network controllability for a special class of digraphs, namely large-scale networks which are Cartesian products of smaller factor-networks (factors). We present two control schemes for extending controllability of factors to the controllability of the composite network, dubbed the control product and layered control. The schemes are relevant to dynamics driven by

A. Chapman, *Semi-Autonomous Networks*, Springer Theses,
DOI 10.1007/978-3-319-15010-9_7

a large class of network based matrices, including the Laplacian and adjacency matrices. Redundancy of the control product is addressed by exploiting the digraph automorphisms of the network, similar to [1] and [6]. The utility of this approach is demonstrated through a distributed scheme for output feedback stabilization and a social network application.

The organization of the chapter is as follows. We begin by introducing relevant background material pertaining to Kronecker sums and graph automorphisms. We describe the network based state dynamics over a large class of matrix representations of digraphs. We then provide a control scheme that forms a controllable state matrix for a composite digraph from the controllable state matrices of the factor digraphs. Symmetries in the digraph provide a sufficient condition for the control scheme to produce the minimal number of controllable input nodes. A second control scheme extends the control of a single factor digraph to the composite digraph by exploiting the layering structure of the Cartesian product. A layered output feedback controller is presented pertaining to the layered control scheme. Finally, we examine the problem of estimating the opinion dynamics of a social influence network utilizing the observability properties of the Cartesian product.

7.2 Digraph Automorphisms

Formally, a digraph automorphism is a permutation σ of the node set such that $\mathcal{D}$ contains an edge (i, j) with weight if and only if it also contains an edge $(\sigma(i), \sigma(j))$ with weight $w_{\sigma(j)\sigma(i)}$. The set of automorphisms, which forms a group, is denoted as Aut($\mathcal{D}$). Every digraph automorphism can be represented uniquely as a permutation matrix J which commutes with the adjacency matrix, i.e., $J\mathscr{A}(\mathcal{D}) = \mathscr{A}(\mathcal{D})J$. An automorphism σ *fixes* node i if $\sigma(i) = i$.

We now proceed to introduce the system dynamics and its relationship to the underlying digraph.

7.3 Problem Setup

There are a number of ways to construct a matrix $A(\mathcal{D}) \in \mathbb{R}^{n\times n}$ from the edges and nodes of an n node digraph $\mathcal{D}$. Some examples we have already touched upon are the adjacency, self-loop, in-degree and out-degree matrices.

One of the properties common to all of the aforementioned matrix representations is that they preserve the symmetries in the digraph. By this we mean that for a representations $A(\cdot)$ if there exists a permutation matrix J corresponding to some digraph automorphism of $\mathcal{D}$ then $A(\mathcal{G})J = JA(\mathcal{G})$. We refer to this matrix representations feature as *symmetry preserving*.

In this chapter, we will be considering the specific family of symmetry preserving matrix representations which also exhibit the added property that the representation is invariant under the Cartesian product. Formally, these representations satisfy

$$A(\mathcal{G}_1 \square \mathcal{G}_2) = A(\mathcal{G}_1) \oplus A(\mathcal{G}_2),$$

for all digraphs $\mathcal{G}_1$ and $\mathcal{G}_2$. In general, any matrix representation of the form[1]

$$[A(\mathcal{G})]_{ij} = \begin{cases} rw_{ii} + \sum_{j \neq i} f\left(w_{ij}, w_{ji}\right) & \text{for } i = j \\ g\left(w_{ij}, w_{ji}\right) & \text{otherwise,} \end{cases}$$

in where $r \in \mathbb{R}$ and $f(\cdot)$ and $g(\cdot)$ are real-valued functions such that $f(0,0) = g(0,0) = 0$, satisfies these properties.We denote this family of matrix representation as $\boldsymbol{A}_\oplus$.

Well known member of $\boldsymbol{A}_\oplus$ is the adjacency matrix $\mathscr{A}(\mathcal{D})$, in-degree matrix $\mathscr{D}_{\text{in}}(\mathcal{D})$, out-degree matrix $\mathscr{D}_{\text{out}}(\mathcal{D})$, in-degree digraph Laplacian (or Laplacian) matrix $\mathscr{L}(\mathcal{D})$, out-degree digraph Laplacian $\mathscr{L}_{\text{out}}(\mathcal{D})$ and the M-matrix representation $M(\mathcal{D})$ where $[M(\mathcal{D})]_{ij} = -[\mathscr{A}(\mathcal{D})]_{ij}$ for $i \neq j$ and $[M(\mathcal{D})]_{ii} = [\mathscr{A}(\mathcal{D})]_{ii}$ investigated in later chapters. The class $\boldsymbol{A}_\oplus$ of representations is by no means a small one and other members will be featured in examples throughout the chapter. It is easy to show that $\boldsymbol{A}_\oplus$ is closed under addition providing a simple mechanism to generate new members.

In this chapter, we explore controllability and observability of systems of the form

$$\dot{x}(t) = A(\mathcal{D})x(t) + Bu(t), \quad y(t) = Cx(t), \tag{7.1}$$

where $A(\cdot) \in \boldsymbol{A}_\oplus$. For brevity, we refer to these dynamics by defining the matrix triplet $(A(\mathcal{D}), B, C)$, or if only the inputs (outputs) are of interest by the matrix pair $(A(\mathcal{D}), B)$ $((A(\mathcal{D}), C))$.

It should be mentioned that, due to the linear system duality between controllability and observability, the pair $(A(\mathcal{D}), B)$ is controllable if and only if the pair $(A(\mathcal{D}), B^T)$ is observable. Hence, the results throughout this chapter will be presented in terms of controllability but are equally applicable to the network observability problem.

A helpful tool to establish controllability is the well-known Popov–Belevitch–Hautus (PBH) test [10]. The test states that (A, B) is uncontrollable if and only if there exists a left eigenvalue–eigenvector pair (λ, v) of A, such that $v^T B = 0$.

It is often of interest where the inputs and outputs of system (7.1) are in terms of the nodes of the digraph $\mathcal{D}$. If the set of input nodes in the n node digraph is $S = \{i_1, i_2, \ldots, i_p\}$ for $i_1 < i_2 < \cdots < i_p$, the corresponding input matrix is $B = \left[e_{i_1}, e_{i_2}, \ldots, e_{i_p}\right] \in \mathbb{R}^{n \times p}$. We uniquely denote the input matrices of this form as $B_n(S)$. Similarly, the output matrices are defined as $C_n(S) := B_n(S)^T$. If it is clear from the context, we remove the subscript n for brevity.

[1] Here we assume that if there is no edge $(i, j) \in E$ then $w_{ji} = 0$.

7.4 Control Product

The following theorem introduces our first method of extending control of the factors of the composite digraph $\mathcal{D}$ to $\mathcal{D}$ itself for the case where $A(\mathcal{D})$ has simple eigenvalues. Specifically, we examine the controllability of the pair $(A(\mathcal{D}), B)$, first discussed in [1], where $\mathcal{D} = \mathcal{D}_1 \square \mathcal{D}_2$ and $B = B_1 \otimes B_2$.

Theorem 7.1. *Consider $A(\cdot) \in \boldsymbol{A}_{\oplus}$ and digraphs $\mathcal{D}_1$ and $\mathcal{D}_2$. Assume $A(\mathcal{D}) = A(\mathcal{D}_1 \square \mathcal{D}_2)$ has simple eigenvalues. Then, the pairs $(A(\mathcal{D}_1), B_1)$ and $(A(\mathcal{D}_2), B_2)$ are controllable if and only if $(A(\mathcal{D}), B)$ is controllable, where $B = B_1 \otimes B_2$.*

Proof. Assuming the eigenvalues of $A(\mathcal{D})$ are simple, w is a left eigenvector of $A(\mathcal{D})$ if and only if $w = u \otimes v$ for some left eigenvector u and v of $A(\mathcal{D}_1)$ and $A(\mathcal{D}_2)$, respectively (see Sect. 6.2.1). Hence, $w^T B = (u \otimes v)^T (B_1 \otimes B_2) = (u^T \otimes v^T)(B_1 \otimes B_2) = u^T B_1 \otimes v^T B_2$. Now, $w^T B = 0$ if and only if $u^T B_1 = 0$ or $v^T B_2 = 0$ (or both). Thus, by the PBH test, $(A(\mathcal{D}), B)$ is controllable if and only if $(A(\mathcal{D}_1), B_1)$ and $(A(\mathcal{D}_2), B_2)$ are controllable. □

Consider the case where $B_1 := B(S_1)$ and $B_2 := B(S_2)$, then for $S = S_1 \times S_2 := \{(i, j) | i \in S_1 \text{ and } j \in S_2\}$ we have $B(S) = B_1 \otimes B_2$. This motivates the name *control product* for the scheme.

The following example demonstrates Theorem 7.1 and the form of set S.

Example 7.2. Consider digraphs $\mathcal{D}_1$ and $\mathcal{D}_2$ in Fig. 7.1. Let the eigenvalues of $\mathscr{L}(\mathcal{D}_1)$ be $\lambda_1, \lambda_2, \lambda_3$ and similarly the eigenvalues for $\mathscr{L}(\mathcal{D}_2)$ be $\mu_1, \mu_2, \mu_3, \mu_4$. As $\lambda_i + \mu_j$ for $i = 1, \ldots, 3$ and $j = 1, \ldots 4$ are the eigenvalues of $\mathscr{L}(\mathcal{D})$, it is efficient to check that the eigenvalues are distinct. For $S_1 = \{1'\}$ and $S_2 = \{1, 2\}$, the pairs $(-\mathscr{L}(\mathcal{D}_1), B_3(S_1))$ and $(-\mathscr{L}(\mathcal{D}_2), B_4(S_2))$ are controllable. The nodes corresponding to sets S_1 and S_2 are half shaded in Fig. 7.1. Now, $B_3(S_1) \otimes B_4(S_2) = B_{12}(S)$ where $S = \{(1', 1), (1', 2)\}$, denoted by full shaded nodes in Fig. 7.1. Therefore from Theorem 7.1, $(-\mathscr{L}(\mathcal{D}), B_{12}(S))$ is controllable.

We now provide an example to illustrate the requirement that the composite digraph in Theorem 7.1 has simple eigenvalues.

Example 7.3. It is a well-known property that $\mathscr{L}(\mathcal{P}_2)$ has simple eigenvalues 0 and 2 and for $S = \{1\}$, $(-\mathscr{L}(\mathcal{P}_2), B_2(S))$ is controllable. Further, $\mathcal{P}_2 \square \mathcal{P}_2 = \mathcal{C}_4$ a length four cycle graph, with non-distinct eigenvalues 0, 2, 2, and 4, and $B_2(S) \otimes B_2(S) = B_4(S)$. But all cycle graphs are uncontrollable from one node (see [5, 6]) so $(-\mathscr{L}(\mathcal{C}_4), B_4(S))$ is uncontrollable.

As an aside, the distinctness of the eigenvalues of $A(\mathcal{D})$ has implication on the structure of $\mathcal{D}$, for example on its prime factor decomposition described in the following proposition, a feature which will be applied later.

Proposition 7.4. *For $A(\cdot) \in \boldsymbol{A}_{\oplus}$, if the eigenvalues of $A(\mathcal{D})$ are simple, then the prime factor decomposition of $\mathcal{D}$ contains no powers of prime digraphs.*

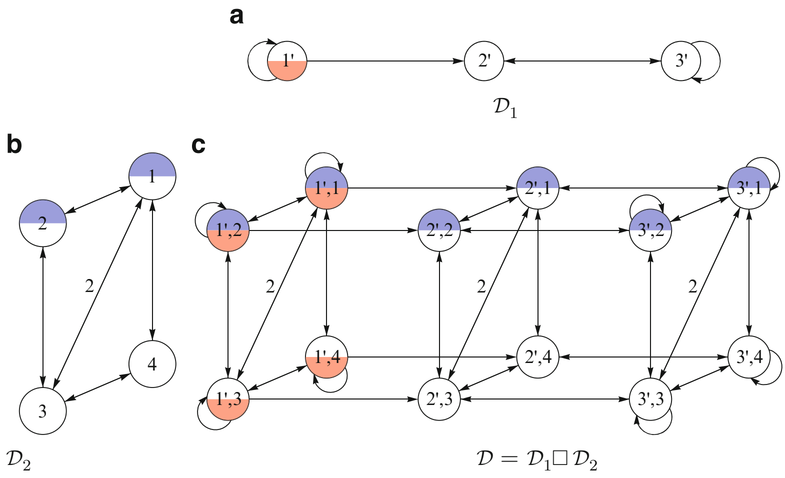


Fig. 7.1 Factor digraphs (**a**) $\mathcal{D}_1$ and (**b**) $\mathcal{D}_2$ and composite digraph (**c**) $\mathcal{D}_1 \square \mathcal{D}_2$. Edge weights of all digraphs are 1 unless otherwise marked. The shading on the nodes pertains to Examples 7.2 and 7.12

Proof. Let the prime factor decomposition of the n node digraph $\mathcal{D}$ be $\mathcal{D} = \mathcal{D}_1^{k_1} \square \cdots \square \mathcal{D}_m^{k_m}$. If $\mathcal{D}$ is a power of a prime digraph, then there exists a $k_i \geq 2$ corresponding to the m node prime factor $\mathcal{D}_i$. Hence, $\mathcal{D} = \mathcal{D}_i^2 \square \mathcal{D}_b$ where $\mathcal{D}_b = \mathcal{D}_1^{k_1} \square \cdots \square \mathcal{D}_i^{k_i - 2} \square \cdots \square \mathcal{D}_m^{k_m}$. Therefore $A(\mathcal{D})$ has pairs of eigenvalues with the value $\lambda_i + \lambda_j + \mu_k$ for $i, j = 1, \ldots, m, i \neq j$ and $k = 1, \ldots, \frac{n}{m}$, where λ_i and λ_j are eigenvalues of $A(\mathcal{D}_i)$, and μ_k is an eigenvalue of $A(\mathcal{D}_b)$. This follows from the eigenvalue properties of the Cartesian sum. □

7.4.1 Breaking Symmetry

The automorphisms of a digraph describe its symmetries and have been previously shown to play an important role in the controllability of $(-\mathscr{L}(\mathcal{D}), B(S))$ for an undirected, unweighted graph $\mathcal{G}$ [1]. The following is a generalization of these results found in [1] and so is quoted here without proof.

Proposition 7.5. *A digraph $\mathcal{D}$ is uncontrollable from any pair $(A(\mathcal{D}), B(S))$ where $A(\cdot)$ is symmetry preserving and $A(\mathcal{D})$ has spanning eigenvectors, if there exists an automorphism of $\mathcal{D}$ which fixes all inputs in the set S.*

We note that Proposition 7.5 is not sufficient for controllability as discussed in [1]. Further, this proposition is applicable to Theorem 7.1, as if a matrix has simple eigenvalues then its eigenvectors are spanning.

Proposition 7.5 highlights the requirement of selecting a set of inputs that break the symmetry structure of $\mathcal{D}$. Determining sets are a useful construct to describe this process.

Definition 7.6. A subset S of the vertices of a digraph $\mathcal{D}$ is called a *determining set* if whenever $g, h \in \text{Aut}(\mathcal{D})$ so that $g(s) = h(s)$ for all $s \in S$, then $g = h$. The determining number of a digraph $\mathcal{D}$, denoted $\det(\mathcal{D})$, is the smallest integer r so that $\mathcal{D}$ has a determining set of size r.

Another term for a determining set S is the *fixing set* due to the fact that no non-trivial automorphism of $\mathcal{D}$ fixes all members in S. Formally, a set $S \subseteq V(\mathcal{D})$ is a determining set if and only if the stabilizing set $\text{stab}(S)$ of S, defined as $\text{stab}(S) := \{g \in \text{Aut}(\mathcal{D}) | \sigma(v) = v, \forall \sigma \in S\}$, only contains the trivial automorphism. Directly from Proposition 7.5 and the definition of determining sets we have the following corollary.

Corollary 7.7. *A necessary condition for controllability of the pair* $(A(\mathcal{D}), B(S))$ *is that* S *is a determining set. Hence,* $|S| \geq \det(\mathcal{D})$.

The automorphism group of a composite digraph is intimately linked to the automorphisms of its prime factors. This link translates through to the determining set of the composite digraph summarized in the following:

Theorem 7.8 ([11]). *Let* $\mathcal{D} = \mathcal{D}_1^{k_1} \square \cdots \square \mathcal{D}_m^{k_m}$ *be the prime factor decomposition for a connected digraph* $\mathcal{D}$. *Then* $\det(\mathcal{D}) = \max\{\det(\mathcal{D}_i^{k_i})\}$.

We now have the required ground work to state a consequence of the digraph automorphism structure of the digraph pertaining to Theorem 7.1.

Theorem 7.9. *Under the assumptions of Theorem 7.1, consider the controllable pairs* $(A(\mathcal{D}_1), B(S_1))$ *and* $(A(\mathcal{D}_2), B(S_2))$ *where* $|S_1| = \det(\mathcal{D}_1)$ *and* $|S_2| = 1$. *Then* $S = S_1 \otimes S_2$ *is the smallest set such that* $(A(\mathcal{D}_1 \square \mathcal{D}_2), B(S))$ *is controllable.*

Proof. As all eigenvalues of $A(\mathcal{D}_1 \square \mathcal{D}_2)$ are simple, from Proposition 7.4 the prime factors of $\mathcal{D}$, and subsequently $\mathcal{D}_1$ and $\mathcal{D}_2$, are relatively prime. Thus from Theorem 7.8, $\det(\mathcal{D}) = \max(\det(\mathcal{D}_1), \det(\mathcal{D}_2))$. Further, as $(A(\mathcal{D}_2), B(S_2))$ is controllable then $1 = |S_2| \geq \det(\mathcal{D}_2) \geq 1$, so $\det(\mathcal{D}) = \det(\mathcal{D}_1)$. Now $B(S_1) \otimes B(S_2) = B(S)$ for some $S \subseteq V(\mathcal{D}_1 \square \mathcal{D}_2)$ and $|S| = |S_1|\,|S_2| = |S_1|$. As $|S| = \det(\mathcal{D})$, the pair $(A(\mathcal{D}_1 \square \mathcal{D}_2), B(S))$ is controllable with the smallest number of inputs. □

We now revisit Example 7.2 with Theorem 7.9 in mind.

Example 7.10. Further examination of Example 7.2, the $\text{Aut}(\mathcal{D}_2) = \{id, \sigma, \tau, \sigma\tau\}$ where id is the identity permutation, $\sigma(1, 2, 3, 4) = (1, 4, 3, 2)$ and $\tau(1, 2, 3, 4) = (3, 2, 1, 4)$. Hence, $\det(\mathcal{D}_2) = 2 = |S_2|$ and $|S_1| = 1$. Applying Theorem 7.9, S is the smallest controllable input set.

7.5 Layered Control

The following theorem details our second method for extending control of the factors to the composite digraph. This control scheme involves repeating the form of control matrix B_1 to every $\mathcal{D}_1$ layer of $\mathcal{D}$, motivating the name *layered control*. As the Kronecker product exhibits permutation equivalency, these results are equivalent to extending the control matrix B_2 to every $\mathcal{D}_2$ layer of $\mathcal{D}$.

Theorem 7.11. *Consider $A(\cdot) \in \boldsymbol{A}_{\oplus}$ and digraphs $\mathcal{D}_1$ and $\mathcal{D}_2$ in Fig. 7.1 with n_1 and n_2 nodes, respectively, where the matrices $A(\mathcal{D}_1)$ and $A(\mathcal{D}_2)$ are diagonalizable. The pair $(A(\mathcal{D}_1), B_1)$ is controllable if and only if $(A(\mathcal{D}), B)$ is controllable, where $\mathcal{D} = \mathcal{D}_1 \square \mathcal{D}_2$ and $B = B_1 \otimes I_{n_2}$.*

Proof. (*Only If*) Assume $(A(\mathcal{D}_1), B_1)$ is controllable; then by the PBH test there exists some left eigenvector u such that $u^T B_1 = 0$. From Sect. 6.2.1, $u \otimes v$ is a left eigenvector of $(A(\mathcal{D}), B)$, where v is some left eigenvector of $(A(\mathcal{D}_2), B_2)$, and $(u \otimes v)^T B = (u^T \otimes v^T)(B_1 \otimes I_{n_2}) = u^T B_1 \otimes v^T = 0$. Therefore, $(A(\mathcal{D}), B)$ is uncontrollable.

(*If*) Assume $(A(\mathcal{D}), B)$ is uncontrollable then there exists some left eigenvector w such that $w^T B = 0$. From Sect. 6.2.1, w is a linear combination of the set of eigenvectors of the form $u_i \otimes v_j$, where u_i and v_j are left eigenvectors of $(A(\mathcal{D}_1), B_1)$ and $(A(\mathcal{D}_2), B_2)$, respectively. Thus, $w = \sum \alpha_{ij}(u_i \otimes v_j)$, where the scalars $\alpha_{ij} \in \mathbb{R}$ are nonzero, and

$$\begin{aligned} 0 = w^T B &= \sum \alpha_{ij}(u_i \otimes v_j)^T (B_1 \otimes I_{n_2}) \\ &= \sum \alpha_{ij}(u_i^T \otimes v_j^T)(B_1 \otimes I_{n_2}) \\ &= \sum \alpha_{ij} u_i^T B_1 \otimes v_j^T, \end{aligned}$$

or equivalently, $\sum \left[\alpha_{ij} u_i^T B_1\right]_k v_j^T = 0$, for $k = 1, \ldots, n_1$. This occurs only if $\left[\alpha_{ij} u_i^T B_1\right]_k = 0$ for all k and i corresponding to some α_{ij}, i.e., $u_i^T B_1 = 0$, since the set of eigenvectors v_j are linearly independent. Therefore, $(A(\mathcal{D}_2), B_2)$ is uncontrollable. □

The assumption of diagonalizability is, for example, satisfied for invertible, real symmetric, and irreducible matrices, as well as matrices with simple eigenvalues. Further, if $\mathcal{D}$ is undirected or strongly connected, then $A(\mathcal{D})$ is a real symmetric or irreducible matrix, respectively—satisfying the assumption. This assumption is removed in the extended journal version of this chapter [15].

An illustrative example of Theorem 7.11 follows.

Example 7.12. Consider digraphs $\mathcal{D}_1$ and $\mathcal{D}_2$ in Fig. 7.1. For $S_1 = \{1'\}$ and $S_2 = \{1, 2\}$, the pairs $(-\mathscr{L}(\mathcal{D}_1), B_3(S_1))$ and $(-\mathscr{L}(\mathcal{D}_2), B_4(S_2))$ are controllable, and $\mathscr{L}(\mathcal{D}_1)$ and $\mathscr{L}(\mathcal{D}_2)$ are diagonalizable. The nodes corresponding to sets S_1 and S_2 are half shaded in Fig. 7.1. Now, $B_3(S_1) \otimes I = B_{12}(S_a)$

where $S_a = \{(1',1),(1',2),(1',3),(1',4)\}$, denoted by the lower half shaded nodes in Fig. 7.1. Similarly, $I \otimes B_4(S_2) = B_{12}(S_b)$ and $S_b = \{(1',1),(1',2),(2',1),(2',2),(3',1),(3',2)\}$, denoted by the upper half shaded nodes in Fig. 7.1. Therefore from Theorem 7.11, pairs $(-\mathscr{L}(\mathcal{D}), B_{12}(S_a))$ and $(-\mathscr{L}(\mathcal{D}), B_{12}(S_b))$ are controllable.

Theorem 7.11 provides a useful tool of combining families of digraphs with known controllability, such as the Laplacian of the path and cycle graphs, with digraphs where controllability is hard to establish, such as random digraphs, or digraphs that are difficult to control, such as the complete graph.

Further, Theorem 7.11 can be combined with Theorem 7.1 to produce controllable digraphs. A composite digraph $\mathcal{D}$ can be decomposed into $\mathcal{D}_a \square \mathcal{D}_b$ where $\mathcal{D}_a$ is the largest factor digraph of $\mathcal{D}$ such that $A(\mathcal{D}_a)$, where $A(\cdot) \in \boldsymbol{A}_{\oplus}$, has simple eigenvalues and $\mathcal{D}_b$ has order n_b. Hence, $\mathcal{D}_a$ can be decomposed into its primes factors $\mathcal{D}_1 \square \ldots \square \mathcal{D}_k$. Assuming controllable matrix pairs $(A(\mathcal{D}_i), B_i)$ for $i = 1, \ldots, k$ can be found, then by Theorem 7.1, $(A(\mathcal{D}_a), B_1 \otimes \cdots \otimes B_k)$ is controllable. By Theorem 7.11, assuming $A(\mathcal{D}_b)$ is diagonalizable, then $\big(A(\mathcal{D}), B_1 \otimes \cdots \otimes B_k \otimes I_{n_b}\big)$ is controllable. This technique is used in the following example to establish controllability of the grid $\mathcal{P}_2 \square \mathcal{P}_4 \square \mathcal{P}_5$.

Example 7.13. Denote the path graphs of length two, four, and five path graph as $\mathcal{P}_2, \mathcal{P}_4$, and $\mathcal{P}_5$, respectively. Since all path graphs are controllable from either end node, designating one of the ends as the first node, the pairs $(-\mathscr{L}(\mathcal{P}_4), B_4(S))$ and $(-\mathscr{L}(\mathcal{P}_5), B_5(S))$ are controllable for $S = \{1\}$. Noting that $\mathscr{L}(\mathcal{P}_4 \square \mathcal{P}_5)$ has distinct eigenvalues using the technique as Example 7.2, from Theorem 7.1, the pair $(-\mathscr{L}(\mathcal{P}_4 \square \mathcal{P}_5), B_{20}(S))$ is controllable. Further applying Theorem 7.11, and noting all graphs involved are undirected which satisfies the diagonalizable assumption, then $(-\mathscr{L}(\mathcal{P}_2 \square \mathcal{P}_4 \square \mathcal{P}_5), B_{40}(S'))$ for $S' = \{(1,1),(2,1)\}$ is controllable as $I_2 \otimes B_{20}(S) = B_{40}(S')$. Here we have a 40-node grid controllable from two nodes.

Interestingly, $\mathscr{L}(\mathcal{P}_2 \square \mathcal{P}_4 \square \mathcal{P}_5)$ has repeated eigenvalues and so, from the PBH test, at least two input nodes are required to form a controllable set. Hence, the set S' found in Example 7.13 is the smallest input set.

7.5.1 Layered Output Feedback

An attraction of composite networks is that they exhibit repeated layers of the factors. Theorem 7.11 takes advantage of this by extending the controllable inputs in one factor layer to many. The same can be done to the observable outputs. The next proposition shows that the control signal can similarly be designed for a factor and extended to the composite network with the effect of generating distributed output feedback stabilization.

Proposition 7.14. *Consider $A(\cdot) \in \boldsymbol{A}_\oplus$ and the n_1 node and n_2 node digraphs $\mathcal{D}_1$ and $\mathcal{D}_2$ in Fig. 7.1, where the matrix $A(\mathcal{D}_2)$ is semistable. If the dynamics $(A(\mathcal{D}_1), B_1, C_1)$ is stabilizable with output feedback $u_a = Ky_a$ for inputs u_a and outputs y_b, then the dynamics $(A(\mathcal{D}_1 \Box \mathcal{D}_2), B_1 \otimes I_{n_2}, C_1 \otimes I_{n_2})$ is stabilizable with the output feedback $u = (K \otimes I_{n_2})y$ for inputs u and outputs y. Further, the control law can be realized with local layer feedback across the layers of $\mathcal{D}_1$.*

Proof. For convenience, we present the equivalent result in terms of the layers of $\mathcal{G}_2$. As K is a stabilizing feedback gain for the system described by the matrices $(A(\mathcal{D}_2), B_2, C_2)$ then $A(\mathcal{D}_2) + B_2KC_2$ is stable. Consider the dynamics of the system $(A(\mathcal{D}_1 \Box \mathcal{D}_2), I_{n_1} \otimes B_2, I_{n_1} \otimes C_2)$ with output feedback $u = (I_{n_1} \otimes K)y$. Then,

$$\begin{aligned}\dot{x}(t) &= \left(A(\mathcal{D}_1 \Box \mathcal{D}_2) + (I_{n_1} \otimes B_2)\,(I_{n_1} \otimes K)\,(I_{n_1} \otimes C_2)\right) x(t)\\ &= \left(A(\mathcal{D}_1) \otimes I_{n_2} + I_{n_1} \otimes A(\mathcal{D}_2) + I_{n_1} \otimes B_2KC_2\right) x(t)\\ &= \left(A(\mathcal{D}_1) \oplus (A(\mathcal{D}_2) + B_2KC_2)\right) x(t).\end{aligned}$$

As the Cartesian sum of semistable and stable matrices is stable then $I_{n_1} \otimes K$ is stabilizing, since each eigenvalue of the composite matrix lies in the left half plane. Further as $I_{n_1} \otimes K$ is block diagonal the feedback loop can be broken into the inputs and outputs of each layer of $\mathcal{D}_2$. Specifically distributing the inputs and outputs, we have $u = \left[u_1^T, \ldots, u_{n_1}^T\right]^T$ and $y = \left[y_1^T, \ldots, y_{n_1}^T\right]^T$, where u_i and y_i are the inputs and outputs of the ith layer of $\mathcal{D}_2$, respectively. Hence, the feedback can be written as $u_i = Ky_i$ for $i = 1, \ldots, n_1$, i.e., local layer feedback. □

Proposition 7.14 describes a setup where we have a stabilizing distributed feedback on each factor layer requiring only local feedback on the sensors and actuators placed on that layer. The following is an example illustrating this layered output feedback stabilization.

Example 7.15. Define the matrix representation

$$[A(\mathcal{D})]_{ij} = \begin{cases} w_{ij} & \text{for } i \neq j \\ \frac{1}{2}w_{ii} - \sum_{i \neq j} w_{ij} & \text{otherwise.} \end{cases}$$

An equivalent form is $A(\mathcal{D}) := -\mathscr{L}(\mathcal{D}) + \frac{1}{2}\mathscr{D}_s$, and so $A(\cdot) \in \boldsymbol{A}_\oplus$. For the digraphs $\mathcal{D}_1$ and $\mathcal{D}_2$ described in Fig. 7.1, $A(\mathcal{D}_1)$ is unstable and $A(\mathcal{D}_2)$ is semistable. Consider the dynamics of the system described by the matrices $(A(\mathcal{D}_1), B(S_1), C(S_2))$ where $S_1 = \{1'\}$ and $S_2 = \{2'\}$ then the output feedback $u = ky$ is stabilizing for $k < -\frac{1}{2}$. From Proposition 7.14, the output feedback $k \otimes I_4$ is stabilizing for the composite system, which is realized by the distributed feedback $u_{(1',i)} = ky_{(2',i)}$ for $i = 1, \ldots, 4$, where $u_{(1',i)}$ is the input applied to node $(1', i)$ and $y_{(2',i)}$ is the output measured from node $(2', i)$.

7.6 Filtering on Social Product Networks

A Cartesian network structure is not uncommon in social networks due to the layered structure of the society. For example, consider an interacting network of nuclear families in a neighborhood. Interactions among families often involve like-gendered parents interacting with like-gendered parents and similarly like-aged children with like-aged children. This network can be realized through a Cartesian product $\mathcal{D}_1 \Box \mathcal{D}_2$ of the inter-family interaction digraph $\mathcal{D}_1$ and the family members interaction digraph $\mathcal{D}_2$.

It is often unrealistic or expensive to make a census of the full population of a social group. An alternative is to sample the network, and subsequently estimate its state dynamics through an *opinion dynamics filter*. However, a requirement for designing such an estimator is observability.

For our example, the underlying model has been adopted from [12], and is the discrete form of the continuous dynamics $(A(\mathcal{D}), C(S))$, where $A(\cdot) = -\mathcal{L}(\cdot) \in \boldsymbol{A}_{\oplus}$, $x(t)$ is the n agents' opinions and $\mathcal{D} = \mathcal{D}_1 \Box \mathcal{D}_2$ is the underlying influence network and S is the set of sampled agents.

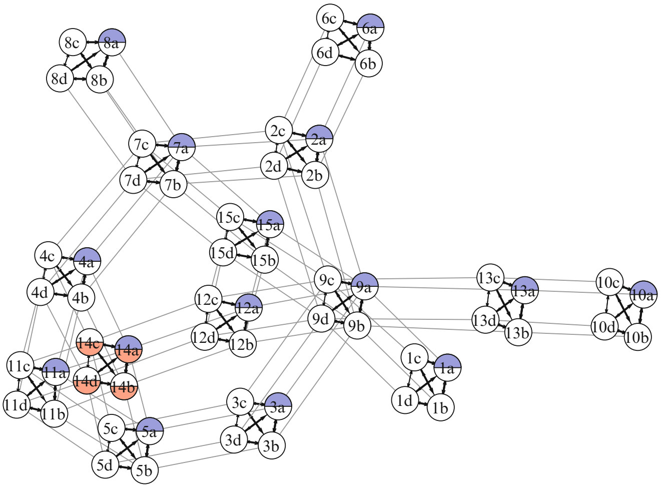
We consider the inter-family interaction digraph $\mathcal{D}_1$ based on the famous Florentine family digraph [13], with each node representing one of the fifteen families. The layers of $\mathcal{D}_1$ are denoted in Fig. 7.2 by the grey undirected, unweighted edges. The family member interaction digraph $\mathcal{D}_2$ with nodes a, b, c, and d correspond to the interaction network in a nuclear family amongst a father, mother, older child, and younger child, respectively, with adjacency matrix $\mathscr{A}(\mathcal{D}_2)$ appearing in Fig. 7.2. The composite digraph $\mathcal{D}_1 \Box \mathcal{D}_2$ corresponds to the resulting 60 members' interaction digraph, and is depicted in Fig. 7.2.

Assuming that all the members of one demographic in the social network can be observed, e.g., all the mothers, then Theorem 7.11, applied to $\mathcal{D}_2$, provides the necessary observability condition for the filter design.

For example, the dynamics are observable, under one demographic, for $C_{60}(S) = I \otimes C_4(S_2)$, where $S_2 = \{a\}$ and $S = S_F := \{1a, 2a, \ldots, 15a\}$, the set of fathers of the society. This information would be attractive to an advertiser, as the opinions of all 60 members can be divulged by surveying the fathers. If instead an advertiser was interested in a good family to survey, then Theorem 7.11 can be applied to $\mathcal{D}_1$, leading to an observability matrix $C_{60}(S) = C_{15}(S_1) \otimes I$, where $S_1 = \{14\}$ and $S = S_{14} := \{14a, 14b, 14c, 14d\}$, i.e., every member of family 14 is observed directly.

Alternatively, as $A(\mathcal{D}_1 \Box \mathcal{D}_2)$ has simple eigenvalues, using Theorem 7.1, the dynamics with the observation matrix $C_{60}(S) = C_{15}(S_1) \otimes C_4(S_2)$ is observable, where $S = S_1 \times S_2 = S_{F14} := \{14a\}$. Therefore, in this scenario, surveying the father of family 14 would provide the opinion of all members.

A discrete Kalman filter was applied to the described social dynamics, with all fathers in the social network observed, i.e., $S = S_F$. A sample opinion state estimate over time is provided in Fig. 7.3, supporting the observability of the pair $(A(\mathcal{D}_1 \Box \mathcal{D}_2), C_{60}(S))$.



$$\mathscr{A}(\mathcal{D}_2) = \begin{bmatrix} 0 & 0.4 & 0.1 & 0.1 \\ 0.3 & 0 & 0.2 & 0.2 \\ 0.4 & 0.3 & 0 & 0.1 \\ 0.35 & 0.45 & 0.1 & 0 \end{bmatrix}$$

Fig. 7.2 Composite digraph $\mathcal{D}_1 \square \mathcal{D}_2$. The layers of $\mathcal{D}_1$ are displayed with *grey* edges and have edge weight 1. The layers of $\mathcal{D}_2$ are *black* with weights divulged through the definition of $\mathscr{A}(\mathcal{D}_2)$. The shading on the nodes pertains to the control inputs, S_F (*upper half shaded nodes*), S_{14} (*lower half shaded nodes*), and S_{F14} (*fully shaded nodes*), described in Example 7.6

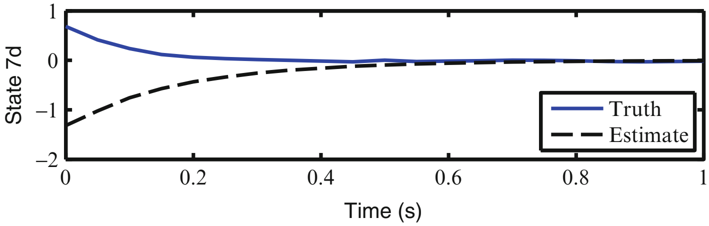


Fig. 7.3 The true and estimated state of a sample member of the society, namely the youngest child of family 7 (node 7d), over time for the discrete Kalman filter pertaining to Example 7.6

7.7 Remarks

This chapter presents an analysis of the controllability and observability of dynamics over composite networks formed by the graph Cartesian product of its factors. Using the tools of graph theory, group theory, and Kronecker algebra, we explored the composition of the controllable input sets of the factor networks to form a controllable control set of the composite network. Future work of particular interest involves extending these results to other types of graph products such as the direct product.

References

1. M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton: Princeton University Press, 2010.
2. C. Godsil, "Controllability on networks," 2009. [Online]. Available: http://quoll.uwaterloo.ca/mine/Talks/control.pdf
3. C. Godsil, "Quantum physics and graph spectra," 2009. [Online]. Available: http://quoll.uwaterloo.ca/mine/Talks/bristol.pdf
4. H. Tanner, "On the controllability of nearest neighbor interconnections," in *43rd IEEE Conference on Decision and Control*, 2004, pp. 2467–2472.
5. M. Nabi-Abdolyousefi and M. Mesbahi, "On the controllability properties of circulant networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 12, pp. 3179–3184, 2013.
6. G. Parlangeli and G. Notarstefano, "Observability and reachability of grid graphs via reduction and symmetries," in *Proc. 50th IEEE Conference on Decision and Control*, 2011, pp. 5923–5928.
7. M. Nabi-Abdolyousefi and M. Mesbahi, "System theory over random consensus networks: controllability and optimality properties," in *Proc. 50th IEEE Conference on Decision and Control*, 2011, pp. 2323–2328.
8. S. Zhang, M. Camlibel, and M. Cao, "Controllability of diffusively-coupled multi-agent systems with general and distance regular coupling topologies," in *Proc. 50th IEEE Conference on Decision and Control*, 2011, pp. 759–764.
9. Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *Nature*, vol. 473, no. 7346, pp. 167–73, May 2011.
10. T. Kailath, *Linear Systems*. Upper Saddle River: Prentice Hall, 1979.
11. D. L. Boutin, "The determining number of a Cartesian product," *Journal of Graph Theory*, vol. 61, no. 2, pp. 77–87, Jun. 2009.
12. N. E. Friedkin and E. C. Johnson, "Influence networks and opinion change," *Advances in Group Processes*, vol. 16, no. 1, pp. 1–29, 1999.
13. J. F. Padgett, "Marriage and elite structure in renaissance Florence, 1282–1500," in *Paper delivered to the Social Science History Association*, 1994.
14. A. Chapman, M. Nabi-Abdolyousefi, and M. Mesbahi, "Identification and infiltration in consensus-type networks," in *1st IFAC Workshop on Estimation and Control of Networked Systems*, 2009, pp. 84–89.
15. A. Chapman, M. Nabi-Abdolyousefi, and M. Mesbahi, "Controllability and observability of networks-of-networks via Cartesian products," *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2668–2679, 2014.

Part IV
Structural Controllability

Chapter 8
Strong Structural Controllability of Networked Dynamics

Abstract This chapter examines strong structural controllability of linear-time-invariant networked systems. We provide necessary and sufficient conditions for strong structural controllability involving constrained matchings over the bipartite graph representation of the network. An $\mathcal{O}(n^2)$ algorithm to validate if a set of inputs leads to a strongly structurally controllable network and to find such an input set is proposed. The problem of finding such a set with minimal cardinality is shown to be NP-complete. Minimal cardinality results for strong and weak structural controllability are compared.

8.1 Introduction

Of increasing importance is the manipulation and monitoring of dynamic networks. The cornerstone of effective control and observation of networked systems is the appreciation of the interplay between system performance and network structure.

Fruitful investigation into this interaction has occurred in the area of controllability. In controllability, the importance of structure has appeared through investigation of the roles of symmetries in the network [3, 6]. There has been a strong focus on consensus-based networks with controllability established for many special families of graphs including circulants, grids, distance regular and Cartesian products [2–4].

Research in weak and strong structural controllability (s-controllability) has had a rich history since its inception [7, 8], and by its very nature exposes the role of structure in network control. Structural controllability establishes generic (weak) and complete (strong) controllability of a network based solely on the direct coupling between nodes appearing as a distinct pattern of zeros in the network dynamics. This is irrespective of the magnitude of these couplings. Weak and strong s-controllability also provide lower and upper bounds, respectively, on the cardinality of a minimum input set for controllability. Figure 8.1 illustrates conceptually this approach to controllability.

One attraction of s-controllability is that, independent of small variations in the coupling strength, controllability can be guaranteed. This provides network controllability robust to parameter uncertainty and floating point errors. Further, unlike controllability, methods to establish s-controllability are typically numerically

A. Chapman, *Semi-Autonomous Networks*, Springer Theses,
DOI 10.1007/978-3-319-15010-9_8

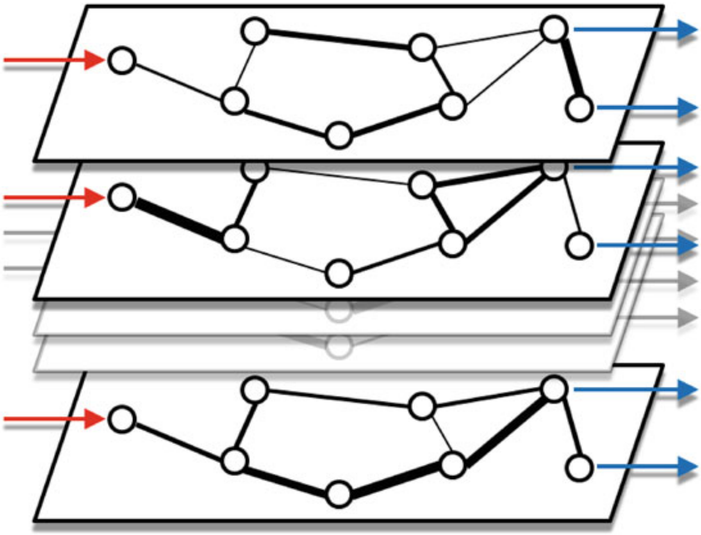


Fig. 8.1 An illustration of the s-controllability concept. Weak and strong s-controllability establish generic and absolute controllability and observability over all realizations of a network

stable. Weak and strong s-controllability also provide lower and upper bounds, respectively, on the cardinality of a minimum input set for controllability.

Recently, a result by Liu et al. has appeared linking weak s-controllability to matchings in a bipartite graph representation of the network [9]. This provides an attractive and efficient way to form generically controllable input sets. Further, unlike controllability, methods to establish s-controllability are typically numerically robust.

Although it is atypical that a weakly s-controllable network has a coupling strength realization rendering it uncontrollable, there are systems that are unconsciously designed with such coupling. An example with such a coupling is unweighted undirected graphs such as the aforementioned consensus-based networks. Such homogeneity generates symmetry in the network which typically renders such a system uncontrollable. For these cases, strong s-controllability presents itself as a useful alternative to weak s-controllability.

The question addressed in the chapter is whether a similar matching-based method can be found to efficiently check and find completely controllable input sets. Steps in this direction were undertaken by Reinschke et al. who established an $\mathcal{O}(n^3)$ method to check an input set is strongly s-controllable [10]. Reinschke et al. conjectured a graph-theoretic method involving spanning cycles to establish strong s-controllability which was recently refined and proven by Jarczyk et al. [11]. We provide necessary and sufficient constrained matching conditions for strong s-controllability in terms of constrained matching conditions. This result has implications for both checking and finding strongly s-controllable input sets.

Our work on this topic entails establishing computationally efficient methods to check and find inputs sets that lead to strong s-controllability. In this direction, an adaptation of the algorithm of Reinschke et al. is provided to check an input set is strongly s-controllable in $\mathcal{O}(n^2)$. It is established that finding a minimum cardinality input set is NP-complete. A greedy $\mathcal{O}(n^2)$ version of our algorithm is supplied to provide a strongly s-controllable input set which has been shown through simulation to perform well.

The organization of the chapter is as follows. We commence with the introduction of pattern matrices and bipartite matching. We describe the linear-time-invariant model that we are examining and present some of the existing results in strong s-controllability. The problem of finding a strongly s-controllable input set is investigated. Finally, we propose an algorithm to validate strong s-controllability and find a strongly s-controllable input set.

8.2 Pattern Matrices

A *pattern matrix* $\boldsymbol{A}$ (or Boolean structure matrix) is a matrix with each element either a zero or a star, denoted throughout by a ×. A numerical matrix A is called a *realization* of the pattern $\boldsymbol{A}$ if A can be obtained by assigning nonzero numerical values to the star entries of $\boldsymbol{A}$, in short $A \in \boldsymbol{A}$. A square pattern matrix has a unique digraph $\mathcal{D}$. The adjacency matrix $\mathscr{A}(\mathcal{D})$ of $\mathcal{D}$ can be formed by replacing the star entries of $\boldsymbol{A}$ with ones. In turn, the pattern matrix $\boldsymbol{A}(\mathcal{D})$ of a digraph $\mathcal{D}$ can be formed by replacing the nonzero entries of $\mathscr{A}(\mathcal{D})$ with stars.

An example of a pattern matrix of a digraph is presented in Example 8.1.

Example 8.1. Consider the digraph $\mathcal{D}$ in Fig. 8.2. The corresponding pattern matrix is

$$\boldsymbol{A}(\mathcal{D}) = \begin{bmatrix} \times & 0 & \times \\ \times & 0 & \times \\ 0 & \times & 0 \end{bmatrix},$$

with ×-elements denoting stars.

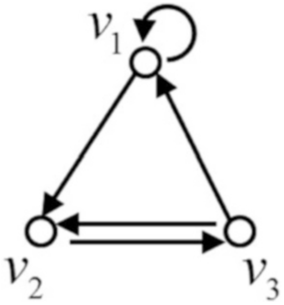


Fig. 8.2 Digraph $\mathcal{D}$

8.3 Model

Commonly, for a linear-time-invariant networked system of single-integrator agents, the digraph $\mathcal{D}$ characterizes the zero structure of the state matrix, and input and output points to the network appear in the input and output matrices [1]. In other terms, the digraph $\mathcal{G}$ and input node set S define pattern matrices and a specific realization of these pattern matrices define the dynamics. Specifically, the state matrices are realizations of the $n \times n$ pattern matrix $\boldsymbol{A}(\mathcal{D})$. The input node set $S = \{i_1, i_2, \ldots, i_m\}$, where $m \leq n$, define the $n \times m$ pattern matrix $\boldsymbol{B}_n(S)$ formed by setting the nonzero entries of $[e_{i_1}, e_{i_2}, \ldots, e_{i_m}] \in \mathbb{R}^{n\times m}$ to stars. Similarly, the output matrix forms the pattern matrix $\boldsymbol{B}_n(S) = \boldsymbol{C}_n(S)^T$. If it is clear from the context, we remove the subscript n for brevity.

In this chapter, we are exploring structural controllability (and observability) of such a system of the form

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t), \end{aligned} \tag{8.1}$$

where $A \in \boldsymbol{A}(\mathcal{D})$, $B \in \boldsymbol{B}(S_1)$ and $C \in \boldsymbol{C}(S_2)$. For succinctness, if $\mathcal{D}$ and S are clear, then $\boldsymbol{A}$,$\boldsymbol{B}$, and $\boldsymbol{C}$ will replace $\boldsymbol{A}(\mathcal{D})$, $\boldsymbol{B}(S_1)$, and $\boldsymbol{C}(S_2)$, respectively. Henceforth, we shall focus on structural controllability of the pair $(\boldsymbol{A}, \boldsymbol{B})$ noting that results can be similarly applied to observability of the pair $\left(\boldsymbol{A}, \boldsymbol{B}^T\right)$ through duality.

8.4 Structural Controllability

Controllability, and its dual system property observability, is often a necessary condition for the application of control system tools, such as stabilization and optimal control. There are many techniques for establishing controllability of linear systems. One such approach for establishing controllability of a pair (A, B) in (8.1) is through the full column rank of the matrix $\left[\, A_\lambda \;\; B \,\right]$ for every eigenvalue λ of A, where $A_\lambda = A - \lambda I$ [12]. Due to the duality between controllability and observability, the pair (A, B) is controllable if and only if the pair (A, C) is observable, where $B = C^T$, and so a similar approach for establishing observability can be performed.

Unfortunately, the above approach, like other controllability criteria, has computational drawbacks, falling victim to rounding errors, and requiring an accurate knowledge of the system dynamics [13]. A more numerically stable approach, which relaxes the modeling precision requirements on the system, is the establishment of s-controllability and observability.

Lin [7] defined a pair $(\boldsymbol{A}, \boldsymbol{B})$ as weak s-controllable if it admits some controllable numerical realization (A, B). From a generic analysis of this setup [14], it follows that weak s-controllability implies that *almost all* numerical realizations of $(\boldsymbol{A}, \boldsymbol{B})$ are controllable. Mayeda and Yamada [8] adapted this by defining the pair $(\boldsymbol{A}, \boldsymbol{B})$ as strongly s-controllable system if *all* numerical realizations (A, B) are controllable. We state that for a given $\boldsymbol{A}$, the input set S is weakly (strongly) s-controllable if the pair $(\boldsymbol{A}, \boldsymbol{B}(S))$ is weakly (strongly) s-controllable.

It is clear that in general a smaller input set S is required to meet the conditions for weak s-controllability as compared to strong s-controllability. This benefit comes with the loss of guaranteed controllability that strong s-controllability provides over all realizations. It is this reduced network attachment cost (small $|S|$) for a high chance of controllability, compared to a conservative certainty of controllability that makes weak and strong s-controllability ideal for reasoning about infiltration and security. More specifically, an infiltrator attempting to control an unknown weighted network will try to establish a cheap weak input set while, for security, a larger strong input set investment is worthwhile for the guarantee, under fluctuations in edge weights that controllability will be maintained.

Extending the aforementioned rank condition for controllability, the pattern matrix pair $(\boldsymbol{A}, \boldsymbol{B})$ in Model (8.1) is weakly (strongly) s-controllable if and only if $\left[\, \boldsymbol{A}_\lambda \;\; \boldsymbol{B} \,\right]$ is full rank for some (all) realizations of $\boldsymbol{A}$ and $\boldsymbol{B}$, and $\lambda \in \mathbb{C}$. In other words, s-controllability can be established through a pattern matrix rank condition. In turn pattern matrix rank conditions can be extracted by examining matchings on an equivalent bipartite graph representation of $(\boldsymbol{A}, \boldsymbol{B})$—a numerically robust combinatorial property. This connection was recently explored by Liu et al. [5] for weak s-controllability. This approach can be extended to strong s-controllability of the pattern matrix pair $(\boldsymbol{A}, \boldsymbol{B})$. The following Theorem 8.2 formally states this extension.

Theorem 8.2 ([10]). *The pair $(\boldsymbol{A}, \boldsymbol{B})$ is strongly s-controllable if and only if the matrix $\left[\, \boldsymbol{A}_\lambda \;\; \boldsymbol{B} \,\right]$ is full rank for all realizations of $\boldsymbol{A}$ and $\boldsymbol{B}$, and $\lambda \in \mathbb{C}$.*

It is with this in mind that we now discuss relevant background on bipartite graphs and matching.

8.4.1 Bipartite Graphs and Matching

A different graph representation of the network, and in fact any $n \times m$ matrix, can be formed using a bipartite graph representation of the interactions. A bipartite graph $\mathcal{H} = \left(V^+, V^-, E\right)$ is an undirected graph on independent node sets V^+ and V^-, where the edge set E connects nodes in V^+ to V^-. A bipartite graph representation of a matrix $M \in \mathbb{R}^{p \times q}$ can be formed by setting $V^+ = \{1, \ldots, q\}$ and $V^- = \{1, \ldots, p\}$ and having an edge $\{i, j\} \in E$ if and only if $[M]_{ij} \neq 0$. The bipartite graph of a pattern matrix of an equivalent $A(\mathcal{D})$ can be formed in this way.

Let $\mathcal{H} = \left(V^+, V^-, E\right)$ be a bipartite graph. A set of r edges $\left\{\left(i_1^+, i_1^-\right), \ldots, \left(i_t^+, i_r^-\right)\right\}$ in $\mathcal{H}$ is said to be a t*-matching* (between $I^+ = \left\{i_1^+, \ldots, i_r^+\right\} \subseteq V^+$ and

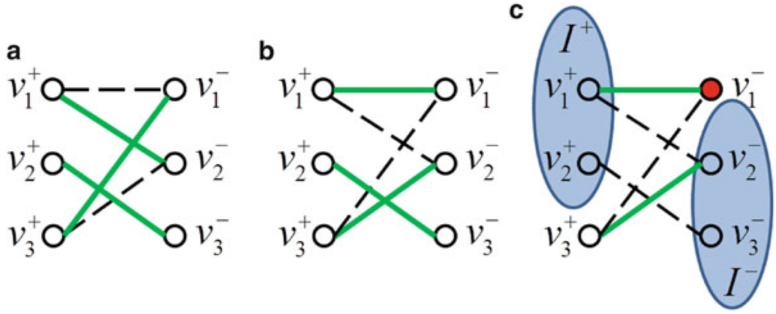


Fig. 8.3 (**a**, **b**) Unconstrained 3-matchings and (**c**) constrained 2-matching with corresponding sets I^+, I^- and unmatched node v_1 (*solid red*) on the bipartite graph $\mathcal{H}$ appearing in Example 8.3

$I^- = \{i_1^-, \ldots, i_r^-\} \subseteq V^-$) if $i_1^+, \ldots, i_r^+$ are distinct and $i_1^-, \ldots, i_r^-$ are distinct. Such a t-matching is said to be a *constrained t-matching* (or uniquely restricted matching) if it is the only t-matching in $\mathcal{H}$ between I^+ and I^-. Those nodes in $\mathcal{H}$ that are in I^- are referred to as *matched*, and those that are not in I^- are called *unmatched*. A matching is *T-less* if it contains no edges of the form (j^+, j^-), where j is in the set T. A T-less matching where $T = \{1, \ldots, \min\{|V^+|, |V^-|\}\}$, is referred to as a *self-less* matching. A (constrained) (T-less) t_1-matching in $\mathcal{H}$ is *maximum* if there is no (constrained) (T-less) t_2-matching in $\mathcal{H}$ with $t_2 > t_1$.

We say that a pattern matrix $\boldsymbol{A}$ has a (constrained) t-matching if the associated bipartite graph $\mathcal{H}$ has a (constrained) t-matching. An example of a pattern matrix of a graph and associated maximum t-matchings is presented in the following example.

Example 8.3. Consider again the graph $\mathcal{D}$ in Fig. 8.3a with the associated bipartite graph $\mathcal{H}$ in Fig. 8.3b. The pattern matrix $\boldsymbol{A}(\mathcal{D})$ appearing in Example 8.3 has two maximum 3-matchings, $\{(v_1^+, v_1^-), (v_2^+, v_3^-), (v_3^+, v_2^-)\}$ and $\{(v_1^+, v_2^-), (v_2^+, v_3^-), (v_3^+, v_1^-)\}$ and four maximum constrained 2-matchings $\{(v_1^+, v_1^-), (v_2^+, v_3^-)\}$, $\{(v_1^+, v_2^-), (v_2^+, v_3^-)\}$, $\{(v_2^+, v_3^-), (v_3^+, v_1^-)\}$ and $\{(v_2^+, v_3^-), (v_3^+, v_2^-)\}$. A selection of these is shown in Fig. 8.3.

There is an intimate relationship between the matrix rank of realizations of $\boldsymbol{A}$ and the (constrained) t-matchings of $\boldsymbol{A}$. The following theorem summarizes some of these links and will be subsequently used in this chapter.

Theorem 8.4 ([15]). *Let $\boldsymbol{A}$ be an $m \times n$ pattern matrix, and let t be a nonnegative integer,* $1 \leq t \leq \min\{m, n\}$.

(1) If $\boldsymbol{A}$ has a t-matching, then there exists a matrix $A \in \boldsymbol{A}$ with rank$(A) \geq t$.
(2) If $\boldsymbol{A}$ has a constrained t-matching, then every matrix $A \in \boldsymbol{A}$ has rank$(A) \geq t$.

Specifically, conditions (1) and (2) provide a mechanism to establish weak s-controllability and strong s-controllability, respectively. For an in-depth study of the pattern matrix rank and matching relationship we refer the reader to Hershkowitz and Schneider [16] and Golumbic et al. [17].

Reinschke et al. [10] provided, without proof, an algebraic result equivalent to Theorem 8.2. The following is a definition necessary for this result.

Definition 8.5. A structured pair $(\boldsymbol{A}, \boldsymbol{B})$ is said to be in form III if there exist two permutation matrices P_1 and P_2 such that

$$P_1 \, [\boldsymbol{A}, \boldsymbol{B}] \, P_2 = \begin{bmatrix} \circledast & \cdots & \circledast & \times & 0 & \cdots & 0 \\ \vdots & & \vdots & \ddots & \times & \ddots & \vdots \\ & & & & \ddots & \ddots & 0 \\ \circledast & \cdots & \circledast & \cdots & \cdots & \circledast & \times \end{bmatrix},$$

where the $\times$-elements denote the location of star elements and the $\circledast$-elements denote the location of either zero or star elements.

Before stating Theorem 8.6, we define $\boldsymbol{A}_\times$ as the pattern matrix formed by placing stars along the diagonal of $\boldsymbol{A}$. The aforementioned theorem follows.

Theorem 8.6 ([10]). *The structured pair $(\boldsymbol{A}, \boldsymbol{B})$ is strongly s-controllable if and only if*

1. *the pair $(\boldsymbol{A}, \boldsymbol{B})$ is of form III,*
2. *and the matrix $(\boldsymbol{A}_\times, \boldsymbol{B})$ can be transformed into form III in such a way that the $\times$-elements do not correspond to diagonal elements that were stars in $\boldsymbol{A}$.*

8.5 Testing Inputs for Strong S-Controllability

The connection of Theorem 8.6 is through the following constrained matching property.

Theorem 8.7 ([17]). *Let $\mathcal{H} = \left(V^+, V^-, E\right)$ be a bipartite graph. A t-matching M is constrained if and only if we can order the nodes of $V^+ = \left\{v_1^+, \dots, v_n^+\right\}$, and $V^- = \left\{v_1^-, \dots, v_m^-\right\}$, such that $\left(v_i^+, v_i^-\right) \in M$, $1 \le i \le t$, and $\left(v_i^+, v_j^-\right) \notin E$ for $1 \le j < i \le t$.*

The following theorem presents the promised equivalent constrained matching result to Theorem 8.6.

Theorem 8.8. *Let S be an input set with cardinality $m \le n$. The pair $(\boldsymbol{A}, \boldsymbol{B}(S))$ is strongly s-controllable if and only if $\boldsymbol{A}(S|\cdot)$ has a constrained $(n-m)$-matching and $\boldsymbol{A}_\times(S|\cdot)$ has a constrained V_s-less $(n-m)$-matching.*

Proof. From Theorem 8.7, a pair $(\boldsymbol{A}, \boldsymbol{B}(S))$ is of Form III (the first condition in Theorem 8.6) if and only if $\boldsymbol{A}(S|\cdot)$ has a constrained $(n-m)$-matching. The second condition in Theorem 8.6 is equivalent to the existence of a constrained V_s-less $(n-m)$-matching in $\boldsymbol{A}_\times(S|\cdot)$. This is apparent by isolating the associated constrained matchings codified in the nonzero diagonal of $P_1 \left[\boldsymbol{A}, \boldsymbol{B}\right] P_2$. The elements on this diagonal will be of the form $\left\{a_{i_1^+ j_1^-}, \dots a_{i_{n-m}^+} a_{j_{n-m}^-}, b_{1k_1}, \dots, b_{mk_m}\right\}$, where $S = \{k_1, \dots, k_m\}$ and correspond to a constrained $(n-m)$-matching $\left\{i_1^+, j_1^-\right\}, \dots, \left\{i_{n-m}^+, j_{n-m}^-\right\}$. Similarly, for $P_1 \left[\boldsymbol{A}_\times, \boldsymbol{B}\right] P_2$ the elements corresponding to V_s appear below the diagonal and so do not appear in the constrained $(n-m)$ matching, making it V_s-less. $\square$

Consequently, a check that an input set S is strongly s-controllable reduces to the problem of finding constrained $(n-m)$-matchings. The conditions in Theorem 8.8 require a validation that there exists a constrained n-matching. A maximum bipartite matching can be found deterministically in $\mathcal{O}(\sqrt{n}\,|E|)$ [18], and testing whether a given bipartite matching is constrained can be checked in $\mathcal{O}\left(n + |E|\right)$ [17].

Reinschke et al. [10] provided a $\mathcal{O}(n^3)$ algorithm to check the conditions in Theorem 8.6. We have used a similar approach in Algorithm 9 but performed the check in $\mathcal{O}(n^2)$, reducing computation by tracking the column sums through the vectors σ and $\sigma_\circledast$. Due to its similarity to [10] we present the algorithm here without proof. This algorithm is further developed in the following section, and so if only validation of strong s-controllability is required then the process can be terminated at the lines marked with asterisks, in which case the set S is not strongly s-controllable. For the case where S is strongly s-controllable, this will be validated once the algorithm has run its course.

8.5.1 Minimum Cardinality Input Sets

In this section we explore the smallest strongly s-controllable input set for a special class of networks. First, a general result.

Proposition 8.9. *There exists no graph that is strongly s-controllable from all single inputs sets $S = \{i\}$, for $i = 1, \dots, n$.*

Proof. Assume otherwise, by Theorem 8.8, $\boldsymbol{A}_\times(\{i\}\,|\cdot)$ has a constrained V_s-less $(n-1)$-matching for every $i \in \{1, \dots, n\}$. Using the property that an $m \times n$ pattern matrix $\boldsymbol{M}$ is full rank if and only if it contains a constrained $\min(m, n)$-matching [16, Theorem 3.9], then every realization $A \in \boldsymbol{A}_\times$ has a rank $n-1$ submatrix when an arbitrary row is removed. Therefore, every such A has rank n and so $\boldsymbol{A}_\times$ has a constrained n-matching from the same property. By Theorem 8.7, there exists permutation matrices P_1 and P_2 such that $P_1 \boldsymbol{A}_\times P_2$ is lower triangular with nonzero diagonal, i.e., the elements corresponding to V_s must lie below the diagonal. As there must be exactly one term of the form a_{ii} in every row and column of $P_1 \boldsymbol{A}_\times P_2$, then the diagonal of $P_1 \boldsymbol{A}_\times P_2$ are the elements $a_{11}, \dots, a_{nn}$ of $\boldsymbol{A}_\times$. Therefore V_s is empty otherwise one of its elements would lie on the diagonal. Similar to the argument for $\boldsymbol{A}_\times$, the pattern matrix $\boldsymbol{A}$ must also be triangularizable with nonzero diagonal elements corresponding to $a_{11}, \dots, a_{nn}$, but if V_s is empty, then $a_{ii} = 0$ for all i and contradiction the fact that $\boldsymbol{A}$ has nonzero diagonal elements. $\square$

Proposition 8.9 is in stark contrast to the weakly s-controllable where there are many networks which exhibit single input controllability from an arbitrary node. One such class of networks that falls in this category for weak s-controllability is the family of self-damped connected networks. These are networks where every node's state directly damps itself, i.e., for every node i, $\dot{x}_i = \alpha_i x_i + \sum_{j \neq i} \alpha_j x_j$, where $\alpha_i \neq 0$. The popular consensus (Laplacian) dynamics fall into this class for connected undirected graphs.

The following two propositions pertain to self-damped undirected networks. Proposition 8.10 illustrates the rarity of single strongly s-controllable inputs.

Proposition 8.10. *The only connected self-damped undirected network strongly s-controllable from a single input is the path graph, controllable from either end node.*

Proof. If $\boldsymbol{A}$ is self-damped and $S = \{i\}$, then $(\boldsymbol{A}, \boldsymbol{B}(S))$ is strongly s-controllable if and only if there exists a permutation matrix P such that $P\boldsymbol{A}P^T$ is unreduced upper-Hessenberg[1] and $P\boldsymbol{B}(S) = \boldsymbol{B}(S)$ [19, Theorem 2.4]. The first condition is equivalent to the graph bandwidth[2] of realizations of $\boldsymbol{A}$ being 2. The only undirected graph with bandwidth 2 is the path graph. In bandwidth form (i.e., with a bandwidth labeling) $P\boldsymbol{B}(S) = \boldsymbol{B}(S)$ if and only if i is either end node. □

On the other extreme, the following proposition indicates that there is only one graph s-controllable from all but one node.

Proposition 8.11. *The only connected self-damped undirected network strongly s-controllable requiring $n-1$ inputs to be strongly s-controllable is the complete graph.*

Proof. If $n-1$ inputs are required, by Theorem 8.8, the largest constrained matching in $\boldsymbol{A}$ is a 1-matching. If the network is not a complete graph, then there exists some edge $\{i, j\} \notin E$, where because it is self-damped, $i \neq j$. As the network is connected then there exists some edge $\{i, p\} \in E$ with $i \neq p$, similarly there exists some edge $\{j, q\} \in E$ with $j \neq q$. Consequently, there is a constrained V_s-less 2-matching $\left\{\left\{v_i^+, v_p^-\right\}, \left\{v_q^+, v_j^-\right\}\right\}$. This satisfies Theorem 8.8, for $m = 2$, i.e., the network is strongly s-controllable from $n-2$ inputs. Further, it can be shown that the largest constrained matching in $\boldsymbol{A}$ for a complete graph is a V_s-less 1-matching. From Theorem 8.8, the proposition follows. □

Due to the importance of connected self-damped undirected networks, Algorithm 9 was exercised on every self-damped undirected network on 2–10 nodes. By testing every permutation of inputs the cardinality of the smallest strongly s-controllable inputs $|S|$ were found. The results are featured in Fig. 8.4, where $n_D := |S|/n$.

[1] A matrix is unreduced upper-Hessenberg if all entries on the first superdiagonal nonzero and all entries above this diagonal are zero.

[2] The bandwidth of a graph is the minimum $\max\{|i-j| \mid \{i, j\} \in E\}$ over all labeling of the nodes.

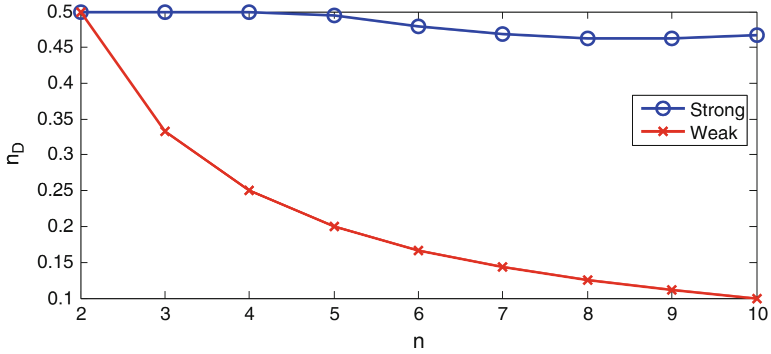


Fig. 8.4 The variable n_D for weak and strong s-controllability for all self-damped undirected graphs for 2–10 nodes

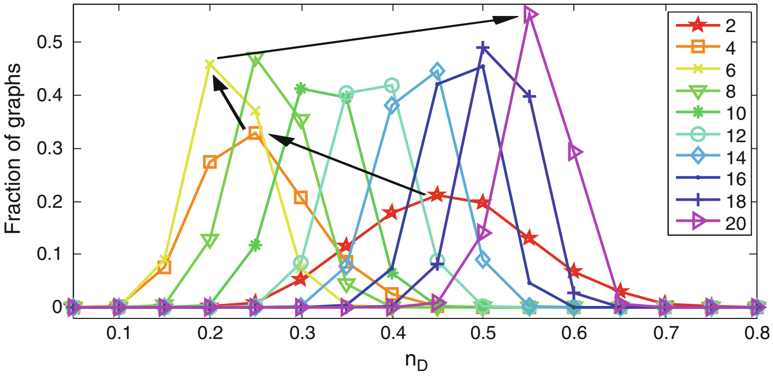


Fig. 8.5 Directed Erdős–Rényi random graphs on 20 nodes. Each $\langle k \rangle$ value denoted in the legend was sampled 1,200 times. The *black arrows* indicate the direction of increasing $\langle k \rangle$ values

We note on average that approximately half the nodes are required to be strongly s-controllable.

To investigate the distribution of n_D across more general graphs, we examine the family of directed Erdős–Rényi random graphs.[3] Algorithm 9 was run on 20-node Erdős–Rényi graphs testing all input permutations, for samplings of $\langle k \rangle$ from 2 to 20 with each $\langle k \rangle$ sampled 1,200 times. Figure 8.5 summarizes these results.

We observe a critical threshold phenomenon as $\langle k \rangle$ increases. Specifically for $\langle k \rangle \leq 6$, n_D decreases with increasing $\langle k \rangle$ while for $\langle k \rangle \geq 6$, n_D increases with increasing $\langle k \rangle$. Such a thresholding phenomenon is not uncommon in random networks [20]. This particular threshold can be attributed to the fact $k_c = 2 \log n \approx 5.99$ is a sharp threshold for the connectedness[4] of Erdős–Rényi random

[3] Directed Erdős–Rényi random graphs are randomly generated graphs with an edge $(i, j) \in E$ independently existing with probability p [20]. The mean degree is defined as $\langle k \rangle = 2np$.

[4] A directed graph is connected if its underlying undirected graph is connected.

networks [20], i.e., for $k < k_c$ ($k > k_c$), the network is almost surely disconnected (connected). Thus for disconnected networks, as connectivity increases, the network becomes controllable from less inputs. Further, for connected graphs as the number of edges increases n_D increases as it becomes more challenging to establish a constrained matching. From a design perspective, $k = k_c$ presents an ideal directed Erdős–Rényi random networks for on average the minimum required number of inputs for strong s-controllability.

An attraction of weak and strong s-controllability is that they provide lower and upper bounds, respectively, on the minimum number of inputs required for general controllability. Consequently, it is a fruitful exercise to compare n_D for weak and strong s-controllability on the two families of graphs presented in this section.

As there always exists an n-matching involving the self-damped edges of the network, by Theorem 9.1, the network is weakly s-controllable from any arbitrary single node. Figure 8.4 compares the average n_D for weak and strong s-controllability. Though the weak s-controllability result implies that almost all graphs are controllable from a single node, we note that, from the strong s-controllability bound, on average the worst case on some graphs can require far more inputs.

Liu et al. [5] examined connected directed Erdős–Rényi random networks finding as n tends to infinity that $n_D \approx e^{-\langle k \rangle/2}$. Figure 8.6 compares the sampled average n_D for weak and strong s-controllability. We observe for small values of $\langle k \rangle$ that the weak and strong bounds are close but as $\langle k \rangle$ increases, n_D for weak s-controllability tends to 0 and n_D for strong s-controllability appears to tend towards 1. This is not surprising and fundamental to the difference between weak and strong s-controllability, in that weak s-controllability requires the existence of t-matchings while strong s-controllability requires constrained t-matchings. The addition of edges in a bipartite graph, for example as $\langle k \rangle$ increases, promotes t-matchings while discourages constrained t-matchings, separating the bounds.

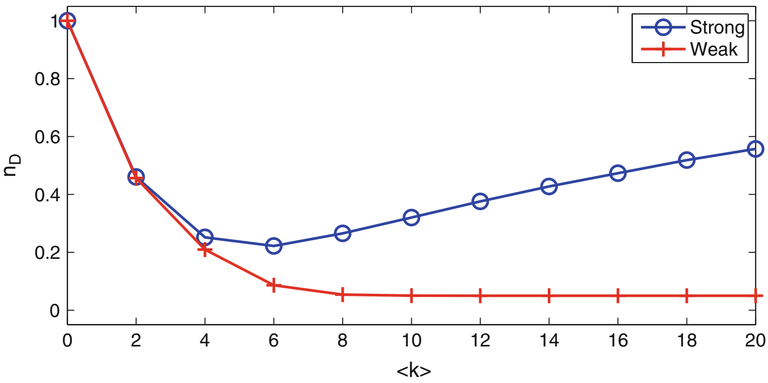


Fig. 8.6 The variable n_D for weak and strong s-controllability for a sampling of directed Erdős–Rényi random networks on 20 nodes. Each $\langle k \rangle$ value for strong s-controllability was sampled 1,200 times

8.6 Finding Strongly S-Controllable Inputs

Validating that a given set of inputs is controllable is computationally distinct from searching for a minimum strongly s-controllable input set. This section focuses on this search problem.

The following theorem provides a bipartite based approach to find strongly s-controllable inputs.

Theorem 8.12. *Given a state matrix $\boldsymbol{A}$ with the constrained $(n-m_1)$-matching with unmatched nodes S_1^+ and S_1^-. Let $\boldsymbol{A}_\times$ contain a constrained V_s-less $(n-m_2)$-matching with unmatched nodes S_2^+ and S_2^-. Then the inputs associated with $S_1^- \bigcup S_2^-$, namely $S_1 \bigcup S_2$, is a strongly s-controllable input set with cardinality $m = |S_1 \bigcup S_2| \leq \min(m_1 + m_2, n)$.*

Proof. First, from Proposition 8.9, both $\boldsymbol{A}$ and $\boldsymbol{A}_\times$ cannot have constrained n-matchings, where the matching on $\boldsymbol{A}_\times$ is V_s-less. Hence, $S_1^- \bigcup S_2^-$ is nonempty. From the first condition of the theorem, $\boldsymbol{A}(S_1|\cdot)$ has a constrained $(n-m_1)$-matching. As constrained matchings are hereditary, $\boldsymbol{A}\ (S_1 \bigcup S_2|\cdot)$ has a constrained $(n-m)$-matching. Similarly, from the second condition of the theorem, $\boldsymbol{A}_\times\ (S_2|\cdot)$ has a V_s-less $(n-m_2)$-matching and consequently, $\boldsymbol{A}_\times\ (S_1 \bigcup S_2|\cdot)$ has a constrained V_s-less $(n-m)$-matching. Thus, from Theorem 8.8 the theorem follows. □

The smallest cardinality strongly s-controllable input set is the smallest cardinality set $S_1 \bigcup S_2$ that satisfies Theorem 8.12.

For the case where $\boldsymbol{A}$ has all diagonal elements nonzero, i.e., the network is self-damped, and the case where $\boldsymbol{A}$ has all diagonal elements zero, i.e., no node is self-damped, only one of the conditions in Theorem 8.12 needs to be validated. This result is summarized in the following corollary.

Corollary 8.13. *Given a state matrix $\boldsymbol{A}$ with diagonal elements all nonzero or all zero. Consider the maximum constrained self-less $(n-m)$-matching of $\boldsymbol{A}_\times$, with unmatched nodes S^+ and S^-. Then the m inputs associated with S^- is a minimum cardinality strongly s-controllable input set.*

Proof. When $\boldsymbol{A}$ has all diagonal elements nonzero, the pattern matrices $\boldsymbol{A}$ and $\boldsymbol{A}_\times$ are equal and share a maximum constrained self-less $(n-m)$-matching. When $\boldsymbol{A}$ has only zero diagonal elements then it can be formed from $\boldsymbol{A}_\times$ by removing its diagonal elements. A constrained t-matching is invariant to the removal of edges from a bipartite graph that are not members of the matching. It follows that $\boldsymbol{A}$ and $\boldsymbol{A}_\times$ share the same constrained self-less $(n-m)$-matching. From Theorem 8.12, the corollary follows. □

From Theorem 8.12 and Corollary 8.13, in the best case the problem of finding the minimum cardinality strongly s-controllable input set is equivalent to finding a maximum constrained matching. This is computationally hard; Golumbic et al. showed that in fact finding a maximum constrained matching in bipartite graphs is

NP-complete [17]. Further, Mishra has recently shown that even approximating a maximum constrained matching is hard demonstrating a bipartite graph that cannot be approximated with a factor of $\frac{1}{2\sqrt[3]{9}}n^{\frac{1}{3}-\epsilon}$ for any $\epsilon > 0$, in polynomial time [21].

We provide a greedy Algorithm 9 to produce an input set that validates each condition in Theorem 8.12. The algorithm involves adding control inputs at points in the algorithm that would traditionally terminate the algorithm returning an "uncontrollable" result. Applying the algorithm involves first inputting the pair $(\boldsymbol{A}, \boldsymbol{B}(S_1))$ where S_1 and T are empty. The algorithm will return an input set S_2 which validates the first condition in Theorem 8.12. Subsequently, applying the algorithm to the pair $(\boldsymbol{A}_{\times}, \boldsymbol{B}(S_2))$ and $T = V_s$ provides a set S_3 which in addition to S_2 validates the second condition in Theorem 8.12. Consequently, a strongly s-controllable pair is $(\boldsymbol{A}, \boldsymbol{B}(S_2 \bigcup S_3))$. Due to the challenges of establishing theoretic guarantee on the cardinality of maximum constrained matching approximation, the following section provides Monte Carlo support for the Algorithm 9's performance.

8.6.1 Algorithm's Performance

This section compares the cardinality of the strongly s-controllable input set $\left|S_{\text{alg}}\right|$ found by Algorithm 9 to the cardinality of the optimal strongly s-controllable inputs set $\left|S_{\text{opt}}\right|$, found through an exhaustive check of all input permutations.

For each of the self-damped undirected graphs up to 10 nodes, the ratio $R = \left|S_{\text{alg}}\right| / \left|S_{\text{opt}}\right|$ was calculated. All such graphs were within a factor of 2 of the optimal cardinality. Similarly, R was calculated for each Erdős–Rényi random networks on $\langle k \rangle = \{2, 6, 10, 14, 18\}$, and summarized in Table 8.1. All such graphs were within a factor of 2.3 of the optimal cardinality. The poorest performance occurred about the threshold value $\langle k \rangle = 6$.

Table 8.1 Fraction of $\langle k \rangle$ Erdős–Rényi random networks on 20-nodes that exhibit an R value less than 1.0, 1.3, 1.7, 2.0 and 2.3, respectively

		R				
		1.0	1.3	1.7	2.0	2.3
$\langle k \rangle$	2	0.54	0.99	1.00	1.00	1.00
	6	0.18	0.61	0.90	$1 - 40\epsilon$	1.00
	10	0.26	0.90	$1 - 30\epsilon$	1.00	1.00
	14	0.34	0.97	1.00	1.00	1.00
	18	0.39	$1 - 8\epsilon$	1.00	1.00	1.00

Here, $\epsilon = 0.0001$

Protocol 9 checks if given $\boldsymbol{A}$, $\boldsymbol{B}(S_1)$ and the set T, if $(\boldsymbol{A}, \boldsymbol{B}(S_1))$ is of form III. If not, inputs S_2 are provided to guarantee $(\boldsymbol{A}, \boldsymbol{B}(S_1 \bigcup S_2))$ is of form III

```
Initialize:
$i = n, j = n + m$
Replace the elements in $\boldsymbol{A}$ corresponding to members of $T$ with the element $\circledast$
Create the row vector $\sigma$ equal to the number of nonzeros, and $\circledast$'s in each column of $[\boldsymbol{A}, \boldsymbol{B}]$
Create the row vector $\sigma_{\circledast}$ equal to the number of $\circledast$'s in each column of $[\boldsymbol{A}, \boldsymbol{B}]$
Create the empty list $S$
Create the $n \times 1$ column vector $p$ with $p(i) = i$
while $i > 0$ do
    Find the minimum positive value $\nu$ with column index $j_s$ of the $1 \times j$ submatrix of $\sigma$ such
    that $\sigma(k) = 1$ and $\sigma_{\circledast}(k) = 1$
    if No such value exists then
        No such matching - Not strongly s-controllable.*
        Add $1, \ldots, i$ to $S$
        Break
    end
    if $\sigma_{\circledast}(j_s) \neq 1$ then
        Find the first nonzero row $i_s$ of the column vector $j_s$ of the $i \times j$ submatrix of $[\boldsymbol{A}, \boldsymbol{B}]$
    else
        Find the first $\circledast$ row $i_s$ of the column vector $j_s$ of the $i \times j$ submatrix of $[\boldsymbol{A}, \boldsymbol{B}]$
    end
    if $i_s \neq i$ then
        Permute row $i_s$ and $i$ of the matrix $[\boldsymbol{A}, \boldsymbol{B}]$, and the column vector $p$
    end
    if $j_s \neq j$ then
        Permute the column $j_s$ and $j$ of the matrix $[\boldsymbol{A}, \boldsymbol{B}]$, and the row vectors $\sigma$ and $\sigma_{\otimes}$
    end
    if $\nu = 1$ then
        $j = j - 1$
    else
        No such matching - Not strongly s-controllable.*
        Add $i$ to $S$
    end
    foreach $k = 1, \ldots, j$ do
        if $[\boldsymbol{A}, \boldsymbol{B}]_{ik}$ is $\circledast$ then
            $\sigma_{\circledast}(k) = \sigma_{\circledast}(k) - 1$
            $\sigma(k) = \sigma(k) - 1$
        else if $[\boldsymbol{A}, \boldsymbol{B}]_{ik}$ is nonzero then
            $\sigma(k) = \sigma(k) - 1$
        end
    end
    $i = i - 1$
end
if $S$ is empty then
    Found such a matching.
else
    Construct the set $S_2 = \{p(S(i)) | i = 1, \ldots, |S|\}$
end
```

8.7 Remarks

This chapter presents an analysis of strong structural controllability in networked systems. We provided an equivalent constrained matching condition for strong structural controllability. A polynomial time algorithm to validate these conditions and to form strong structural controllable input sets was presented. We proceeded to show that the search for a minimum cardinality strong structural controllability input set is NP-complete; finding a factor approximation is also shown to be difficult. Insights were provided into the spread of minimum controllable input sets using weak and strong structural controllability bounds. Future work of particular interest involves establishing conditions when weak and strong structural controllability share similar size minimum cardinality input sets.

References

1. M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton: Princeton University Press, 2010.
2. M. Nabi-Abdolyousefi and M. Mesbahi, "On the controllability properties of circulant networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 12, pp. 3179–3184, 2013.
3. G. Parlangeli and G. Notarstefano, "Observability and reachability of grid graphs via reduction and symmetries," in *Proc. 50th IEEE Conference on Decision and Control*, 2011, pp. 5923–5928.
4. S. Zhang, M. Camlibel, and M. Cao, "Controllability of diffusively-coupled multi-agent systems with general and distance regular coupling topologies," in *Proc. 50th IEEE Conference on Decision and Control*, 2011, pp. 759–764.
5. Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *Nature*, vol. 473, no. 7346, pp. 167–73, May 2011.
6. A. Rahmani and M. Mesbahi, "Pulling the strings on agreement: anchoring, controllability, and graph automorphisms," in *Proc. American Control Conference*, 2007, pp. 2738–2743.
7. C.-T. Lin, "Structural Controllability," *IEEE Transactions on Automatic Control*, vol. AC-19, no. 3, pp. 201–208, Aug. 1974.
8. H. Mayeda and T. Yamada, "Strong structural controllability," *SIAM Journal on Control and Optimization*, vol. 17, no. 1, pp. 123–138, 1979.
9. Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks (Supplementary)," *Nature*, vol. 473, no. 7346, pp. 167–73, May 2011.
10. K. J. Reinschke, F. Svaricek, and H.-D. Wend, "On strong structural controllability of linear systems," in *Proc. 31st IEEE Conference on Decision and Control*, no. 1, 1992, pp. 203–206.
11. J. C. Jarczyk, F. Svaricek, and B. Alt, "Strong structural controllability of linear systems revisited," in *Proc. 50th IEEE Conference on Decision and Control*, Dec. 2011, pp. 1213–1218.
12. C.-T. Chen, *Linear Systems Theory and Design*. New York: Oxford University Press, 1999.
13. C. Paige, "Properties of numerical algorithms related to computing controllability," *IEEE Transactions on Automatic Control*, vol. 26, no. 1, pp. 130–138, Feb. 1981.
14. R. Shields and J. Pearson, "Structural controllability of multiinput linear systems," *IEEE Transactions on Automatic Control*, vol. 21, no. 2, pp. 203–212, Apr. 1976.
15. R. Cantó, M. Boix, and B. Ricarte, "On the minimal rank completion problem for pattern matrices," *WSEAS Transactions on Mathematics*, vol. 3, no. 3, pp. 711–716, 2004.

16. D. Hershkowitz and H. Schneider, “Ranks of zero patterns and sign patterns,” *Linear and Multilinear Algebra*, vol. 34, no. 90, pp. 3–19, 1993.
17. M. Golumbic, T. Hirst, and M. Lewenstein, “Uniquely restricted matchings,” *Algorithmica*, vol. 31, no. 2, pp. 139–154, 2001.
18. S. Micali and V. V. Vazirani, “An $O(\sqrt{|V|}|E|)$ algorithm for finding maximum matching in general graphs,” in *21st Annual Symposium on Foundations of Computer Science*, Oct. 1980, pp. 17–27.
19. D. D. Olesky, M. Tsatsomeros, and P. van den Driessche, “Qualitative controllability and uncontrollability by a single entry,” *Linear Algebra and its Applications*, vol. 187, pp. 183–194, 1993.
20. B. Bollobás, *Random Graphs*. New York: Cambridge University Press, 2001.
21. S. Mishra, “On the maximum uniquely restricted matching for bipartite graphs,” *Electronic Notes in Discrete Mathematics*, vol. 37, pp. 345–350, Aug. 2011.

Chapter 9
Security and Infiltration of Networks: A Structural Controllability and Observability Perspective

Abstract This chapter examines the role of structural controllability (s-controllability) in the design of secure linear-time-invariant networked systems. We reason about secure network design in the face of two attack vectors: a “Disrupt” attack where the infiltrator’s objective is to perturb the network to render it unusable, and a “Highjack and eavesdrop” attack to actively control and probe the network. For the former attack, strong s-controllable input sets are chosen to control the network to provide robustness to these attacks. Weak s-controllable input sets are selected by infiltrators for the “Highjack and eavesdrop” attack so as to generically guarantee a successful attack.

9.1 Introduction

Network security and its complement, network infiltration, are of paramount importance in design and analysis of complex dynamic networks. Dynamic networks are core to the natural world, such as biological, chemical, and social networks, and our technological world with networks such as the Internet, power grids, and robotic networks. In recent years, there has been a myriad of research in the area of network systems [6–8]. Of increasing importance is securing networks from outside manipulation and monitoring. The complementary problem is identifying effective means to infiltrate, influence, or identify networks. At the cornerstone of these problems is the interplay between system performance and network structure.

Most modern day technological networks rely on access security to protect the network. An alternative method to generate a secure network is intrusion detection[1] coupled with either inter-agent security through each agent’s dynamics or intra-agent security via the network topology. The former includes implementation of disturbance rejection or agent disabling techniques, e.g., noise canceling systems and power grid “brown outs”. The latter involves global or local network design to

[1] Techniques for intrusion or fault detection on consensus-type networks include those based on reachability analysis [3], and the more popular unknown-input observers [1, 4, 5].

A. Chapman, *Semi-Autonomous Networks*, Springer Theses,
DOI 10.1007/978-3-319-15010-9_9

improve security, e.g., TCP network re-routing. This topology-centric approach for security is the main focus of the present work. Specifically, the following two attack vectors are the focus of this chapter:

- Disrupt: An infiltrator perturbs the coupling strength between nodes in the network to disrupt the user's ability to control and/or monitor the network.
- Highjack and eavesdrop: An infiltrator with knowledge of the inter-agent dependencies, i.e., who effects whom, but not the coupling strength between nodes, attaches to a subset of "infiltrated" nodes in the graph. The infiltrator then applies signals and/or monitors the state of these nodes to control and/or identify the network.

The current work is part of a more general effort that aims to identify fundamental bounds on the security of coordination algorithms for dynamic systems when infiltrated by an adversary. As such, our work is related to a number of other research works such as those in computer network security [13], disease control [14, 15], and predator/prey swarming [2]. The chapter presents the problem of network security from both the infiltrator's perspective complementing work on infiltration detection such as [3].

Structural controllability (s-controllability) provides controllability results which only require knowledge of interconnections and not the associated coupling strengths, establishing weak and strong s-controllability removes the accurate modeling requirement of controllability. A such, s-controllability is an attractive tool for infiltrators to establish low cardinality s-controllable input sets to highjack and eavesdrop on the network without accurate knowledge of the network parameters. Typically, infiltrating more nodes in the network is difficult and exposes an infiltrator to a higher risk of detection. Thus, weak s-controllability provides a more cost-effective measure than strong s-controllability and for this reason will be used as the main tool in this chapter for analysis of the "Highjack and eavesdrop" attack scenario.

Although it is atypical that a weakly s-controllable network has a coupling strength realization rendering it uncontrollable, there are systems that are inadvertently designed or maliciously altered with such a coupling. Examples with such a coupling are unweighted undirected graphs such as the aforementioned consensus-based networks and the "Disrupt" attack scenario. Such homogeneity generates symmetry in the network which typically renders a system uncontrollable. For these cases, strong s-controllability is a useful alternative to weak s-controllability, and will be used in this chapter as a method to secure networks against "Disrupt" attacks.

In this chapter s-controllability conditions are exploited to reason about a secure design against "Disrupt" and "Highjack and eavesdrop" attacks.

9.2 Weak Structural Controllability: A Cautious Lower Bound

As previously mentioned, weak s-controllability is a generic condition on controllability. If the family of systems in Model (8.1) is weak s-controllable, a given realization will be controllable with high probability. For non-homogeneous networks with independent edge weights, if an input set satisfies weak s-controllability, it will, in general, also satisfy general controllability. For homogenous networks, such as unweighted networks, symmetries inherent in the network tend to break the controllability of weak inputs. For example, from a single input node the undirected complete graph is weak s-controllable but its unweighted realization requires $n - 1$ nodes to control completely.

From an infiltrator's perspective, if the network has unknown weights, weak s-controllability presents a useful method to isolate, with high probability, controllable or observable network attachment points. The necessity of controllability and observability for effective network identification and control makes weak s-controllability ideal for infiltrator "Highjack and Eavesdrop" scenarios. From a different perspective, a cautious lower bound suitable for network security is additional protection of weak input attachment nodes to prevent such an attack. To this end, the following theorem provides a matching method to find and check for weak input sets [9].

Theorem 9.1. *Given a pattern matrix $\boldsymbol{A}(\mathcal{D})$:*

(1) (check) The pair $(\boldsymbol{A}, \boldsymbol{B}(S))$ is weakly s-controllable from a nonempty m-input set S if and only if $\boldsymbol{A}(S|\cdot)$ has an $(n-m)$-matching and S is input accessible.

(2) (find) Given a maximum matching with unmatched nodes S': If S' is empty $S = \{i\}$ for any $i = 1, \ldots, n$, otherwise $S = S'$. If T is the rooted set for those nodes in $\mathcal{G}$ (input) inaccessible to S, then $(\boldsymbol{A}, \boldsymbol{B}(S \bigcup T))$ is weakly s-controllable.

The strength of this result is that the structure of the graph can be directly linked to a controllable input set. Further, this graph feature described in Theorem 9.1(1) can be efficiently checked via the method of maximum matching in bipartite graphs which can be calculated deterministically in $\mathcal{O}\left(\sqrt{n}\,|E|\right)$ time [12] and probabilistically in $\mathcal{O}(n^{2.376})$ time [16], coupled with a depth first search to determine input accessibility in $\mathcal{O}(|E|)$ time [17]. Theorem 9.1 is demonstrated in the following example.

Example 9.2. Referring to Example 8.1, for $\boldsymbol{A}(S|\cdot)$ the set $S = \{i\}$ is input accessible and has a 2-matching for $i = 1, 2$, and 3, namely $\left\{\left(v_2^+, v_3^-\right), \left(v_3^+, v_2^-\right)\right\}$, $\left\{\left(v_1^+, v_1^-\right), \left(v_2^+, v_3^-\right)\right\}$ and $\left\{\left(v_1^+, v_2^-\right), \left(v_3^+, v_1^-\right)\right\}$, respectively. These matchings appear in Fig. 9.1. As each of these input sets satisfies the condition in Theorem 9.1, the pairs $(\boldsymbol{A}, \boldsymbol{B}(\{1\}))$, $(\boldsymbol{A}, \boldsymbol{B}(\{2\}))$, and $(\boldsymbol{A}, \boldsymbol{B}(\{3\}))$ are weakly s-controllable.

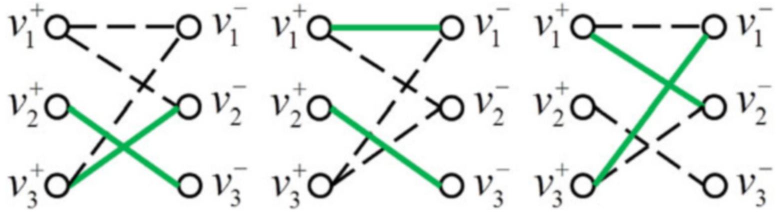


Fig. 9.1 Three 2-matchings relevant to Example 9.2 satisfying the matching condition in Theorem 9.1 on the bipartite graph $\mathcal{H}$ appearing in Example 8.1

The following example examines a "Highjack and Eavesdrop" scenario for an unknown realization of Model (8.1). Theorem 9.1 is applied to acquire the most vulnerable weakly controllable attack vectors and reason about the relative node security in the network.

Example 9.3 (Highjack and Eavesdrop). Consider the graph $\mathcal{D}$ in Fig. 9.2a and network dynamics corresponding to a realization of $(\boldsymbol{A}(\mathcal{D}), \boldsymbol{B}(S), \boldsymbol{C}(S))$ in Model (8.1). A weakly controllable inputs set S presents a particularly attractive attack vector for an infiltrator. The node set S almost always provides a controllable input and observable output sets to effectively control (highjack) the network and/or monitor (eavesdrop on) the network through system identification.

Applying Theorem 9.1 to all possible input sets the smallest cardinality attack vectors involve the successful infiltration of two nodes. Specifically, the nine smallest attack vectors are of the form $\{j, 7\}$ and $\{i, 8\}$, where $j \in \{1, 3, 4, 5\}$ and $i \in \{1, 2, 3, 5, 6, 7\}$. All larger attack vectors contain one of these pairs. In aggregate there are 821 attack vectors, the 345 attacks involving at most four nodes are distributed as $\mathcal{A} = \{38, 36, 38, 37, 38, 36, 60, 62\}$ where the number of attacks involving node i is $\mathcal{A}(i)$.

From a network design perspective the most to least vulnerable node sets are $\{8\}, \{7\}, \{1, 3, 5\}, \{4\}, \{2, 6\}$, providing a priority ordering for security. Figure 9.2b indicates this ordering. Further, if nodes 7 and 8 are completely secured against attachment, then there will be no input set that will render the system controllable or observable.

9.3 Strong Structural Controllability: Guaranteed Security

An attractive feature of s-controllability is the provided controllability guarantees in the face of perturbations. Specifically, as long as the interconnections in the graph remain intact and no new ones are added controllability will be maintained. One can consider this a type of *controllability robustness*—an often elusive feature of system dynamics, with progress made in the area of controllability of interval matrices [18]. For arbitrary edge weight perturbations the benefits of weak s-controllability are

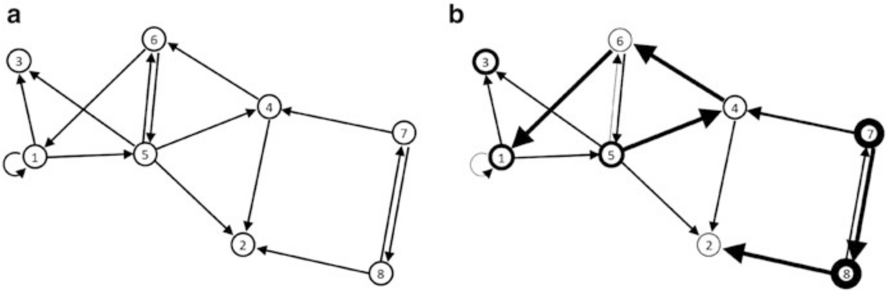


Fig. 9.2 (**a**) Graph $\mathcal{D}$ and (**b**) resultant node and edge weights importance pertaining to Examples 9.3 and 9.6

similar to that of strong s-controllability. In a malicious infiltrator scenario where edge weights may be perturbed to actively disrupt the network then edge weights rendering a weak input set uncontrollable may be intentionally selected. For this "Disrupt" scenario, an input set that is strong is preferred as, up to the removal of edges, the input set remains controllable, guaranteeing security.

Noting the similarities in the definitions for weak and strong s-controllability, it is not surprising that structural features that are equivalent to weak s-controllability share similarities to structural features that are equivalent to strong s-controllability. It is with this in mind that we explore the role of t-matchings in strong s-controllability.

Validating that a given set of inputs is controllable is computationally distinct from searching for a minimum strongly s-controllable input set. This section focuses on this search problem. The following theorem provides a bipartite matching approach to find strongly s-controllable inputs.

Theorem 9.4. *Given a pattern matrix $\boldsymbol{A}(\mathcal{G})$.*

(1) (check) The pair $(\boldsymbol{A}, \boldsymbol{B}(S))$ is strongly s-controllable from a nonempty m-input set S if and only if $\boldsymbol{A}(S|\cdot)$ has a constrained $(n-m)$-matching and $\boldsymbol{A}_{\times}(S|\cdot)$ has a constrained V_s-less $(n-m)$-matching

(2) (find) Given a constrained $(n-m_1)$-matching on $\boldsymbol{A}$ with unmatched nodes S_1 and a constrained V_s-less $(n-m_2)$-matching on $\boldsymbol{A}_{\times}$ with unmatched nodes S_2, then $(\boldsymbol{A}, \boldsymbol{B}(S_1 \bigcup S_2))$ is strongly s-controllable.

Fundamentally, the "check" condition in Theorem 9.4 requires a validation that there exists a maximum n-matching in $\boldsymbol{A}$ $(S|\cdot)$ and in $\boldsymbol{A}_{\times}$ $(S|\cdot)$ that is constrained. A maximum bipartite matching can be found deterministically in $\mathcal{O}(\sqrt{n}\,|E|)$ time [12], and testing whether a given bipartite matching is constrained can be checked in $\mathcal{O}\,(n+|E|)$ time [11]. Reinschke et al. [10] provided a $\mathcal{O}(n^3)$ algorithm to check the conditions in Theorem 9.4. We have presented a similar approach performing the check in $\mathcal{O}(n^2)$ [19].

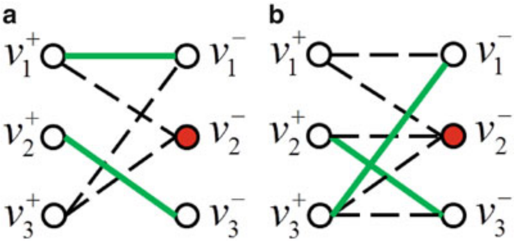


Fig. 9.3 (**a**) Constrained 2-matching on $\boldsymbol{A}(\{2\}|\cdot)$ and (**b**) V_s-less constrained 2-matching on $\boldsymbol{A}_\times(\{2\}|\cdot)$ relevant to Example 9.5 satisfying the conditions of Theorem 9.4 with input set $S = \{2\}$ on the bipartite graph $\mathcal{H}$ appearing in Example 8.1

Consequently, a check that an input set S is strongly s-controllable reduces to a problem of finding constrained $(n - m)$-matchings. This is demonstrated in the following example.

Example 9.5. Returning to Example 8.1, for $i = 1$ and 2 (but not 3), $\boldsymbol{A}(S|\cdot)$ with input set $S = \{i\}$ has a constrained 2-matching, satisfying the first condition in Theorem 9.4(1). Examining the matrix

$$\boldsymbol{A}_\times = \begin{bmatrix} \times & 0 & \times \\ \times & \times & \times \\ 0 & \times & \times \end{bmatrix},$$

and setting $V_s = \{1\}$, the pattern matrix $\boldsymbol{A}_\times$ has three constrained V_s-less 2-matchings, $\{(v_1^+, v_2^-), (v_2^+, v_3^-)\}$, $\{(v_2^+, v_3^-), (v_3^+, v_1^-)\}$ and $\{(v_2^+, v_3^-), (v_3^+, v_2^-)\}$. Thus for $i = 1$ and 2 (but not 3), the input set $S = \{i\}$ with $\boldsymbol{A}_\times(S|\cdot)$ has a constrained V_s-less 2-matchings satisfying the second condition in Theorem 9.4(1). Therefore, the pairs $(\boldsymbol{A}, \boldsymbol{B}(\{1\}))$ and $(\boldsymbol{A}, \boldsymbol{B}(\{2\}))$ are strongly s-controllable. The matchings associated with input set $\{2\}$ appear in Fig. 9.3.

We now apply Theorem 9.4 to a "Disrupt" scenario where an infiltrator perturbs edge weights, potentially removing them so as to reduce the effectiveness of the control input into network running a realization of Model (8.1). Examining the effect of edge failures on strong s-controllability of the input set one can identify the critical edges in the network and identify the most significant security vulnerabilities.

Example 9.6 (Disrupt). Consider the graph $\mathcal{D}$ in Fig. 9.2a and network dynamics corresponding to a realization of $(\boldsymbol{A}(\mathcal{D}), \boldsymbol{B}(S), \boldsymbol{C}(S))$ in Model (8.1), where $S = \{3, 5, 7\}$. Applying Theorem 9.4 reveals that the system is strongly s-controllable and observable. If an infiltrator's objective is to disrupt the effective control and monitoring of the network, a viable strategy is to perturb the network's interconnection strengths, i.e., its edge weights. As long as no interconnections are broken, input set S can still control and monitor the network. But, in the event that edges can fail, it is important to establish the most critical.

Applying Theorem 9.4 to every combination of edge failures there are 241,216 successful attack combinations that cause the loss of strong s-controllability. The most critical edges, whose individual failure removes strong s-controllability, are edges in $E_1 = \{4 \to 6, 5 \to 4, 6 \to 1, 7 \to 8, 8 \to 2\}$ and $E_2 = \{7 \to 4\}$. All other successful combinations involve edges in $E_1 \bigcup E_2$. It is important to note that unlike Example 9.3, where if S_1 is weak and $S_1 \subseteq S_2$ then S_2 is weak, the same condition does not hold for edge failure attack vectors. For example, the removal of E_2 renders the system no longer strong s-controllable but also removing $1 \to 1$ returns strong s-controllability.

Examining all 1,208 successful attacks involving three edges, there were 106, 89, 67, and 60 successful attacks involving each of the nodes in E_1, E_2, E_3, and E_4, respectively, where $E_3 = E \backslash \left\{ \bigcup_{i=1}^{3} E_i \right\}$ and $E_4 = \{1 \to 1, 5 \to 6\}$. Consequently, the network edge sets $E_1, \ldots, E_4$ present a preferential ordering of the distribution of security resources. Figure 9.2b depicts this ordering.

9.4 Remarks

This chapter presents an analysis of the security of networked system topologies using weak and strong s-controllability. Focusing on “Disrupt” and “Highjack and eavesdrop” attack scenarios, we propose controllability metrics to identify vulnerable nodes and critical edges of a network. This was accomplished through a computationally efficient matching condition on weak and strong s-controllability. Future work of particular interest involves establishing conditions for output weak and strong s-controllability and their implications for network security.

References

1. S. Sundaram and C. N. Hadjicostis, “Distributed function calculation via linear iterations in the presence of malicious agents - part I : attacking the network,” in *Proc. American Control Conference*, 2008, pp. 1350–1355.
2. R. Olfati-Saber, “Flocking for multi-agent dynamic systems: algorithms and theory,” *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.
3. A. Fagiolini, G. Valenti, L. Pallottino, G. Dini, and A. Bicchi, “Decentralized intrusion detection for secure cooperative multi-agent systems,” in *Proc. 46th IEEE Conference on Decision and Control*, 2007, pp. 1553–1558.
4. F. Pasqualetti, A. Bicchi, and F. Bullo, “Distributed intrusion detection for secure consensus computations,” in *Proc. 46th IEEE Conference on Decision and Control*, 2007, pp. 5594–5599.
5. F. Pasqualetti, A. Bicchi, and F. Bullo, “On the security of linear consensus networks,” in *Proc. 48th IEEE Conference on Decision and Control*, 2009, pp. 4894–4901.
6. S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. Hwang, “Complex networks: structure and dynamics,” *Physics Reports*, vol. 424, no. 4–5, pp. 175–308, 2006.

7. N. Ganguly, A. Deutsch, and A. Mukherjee, *Dynamics on and of Complex Networks: Applications to Biology, Computer Science, and the Social Sciences*. Boston: Birkhauser, 2009.
8. L. Kocarev and G. Vattay, *Complex Dynamics in Communication Networks*. Berlin: Springer, 2005.
9. Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *Nature*, vol. 473, no. 7346, pp. 167–73, May 2011.
10. K. J. Reinschke, F. Svaricek, and H.-D. Wend, "On strong structural controllability of linear systems," in *Proc. 31st IEEE Conference on Decision and Control*, no. 1, 1992, pp. 203–206.
11. M. Golumbic, T. Hirst, and M. Lewenstein, "Uniquely restricted matchings," *Algorithmica*, vol. 31, no. 2, pp. 139–154, 2001.
12. S. Micali and V. V. Vazirani, "An $O(\sqrt{|V|}|E|)$ algorithm for finding maximum matching in general graphs," in *21st Annual Symposium on Foundations of Computer Science*, Oct. 1980, pp. 17–27.
13. A. Gueye and J. C. Walrand, "Security in networks: A game-theoretic approach," in *Proc. 47th IEEE Conference on Decision and Control*, 2008, pp. 829–834.
14. M. Bloem, T. Alpcan, and T. Basar, "Optimal and robust epidemic response for multiple networks," in *Proc. 46th IEEE Conference on Decision and Control*, 2007, pp. 5074–5079.
15. Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos, "Epidemic spreading in real networks: an eigenvalue viewpoint," in *Proc. 22nd International Symposium on Reliable Distributed Systems*, 2003, pp. 25–34.
16. M. Mucha and P. Sankowski, "Maximum matching via Gaussian elimination," in *45th IEEE Symp. Foundations of Computer Science*, 2004, pp. 248–255.
17. S. Even, *Graph Algorithms*. New York: Cambridge University Press, 2011.
18. K. Wang and A. Michel, "Necessary and sufficient conditions for the controllability and observability of a class of linear, time-invariant systems with interval plants," *IEEE Transactions on Automatic Control*, vol. 39, no. 7, pp. 1443–1447, 1994.
19. A. Chapman and M. Mesbahi, "Strong structural controllability of networked dynamics," in *Proc. American Control Conference*, 2013, pp. 6141–6146.

Final Remarks

Chapter 10
Conclusion and Future Work

Abstract This chapter provides some final remarks and explores future research directions in the area of semi-autonomous networks.

10.1 Concluding Remarks

The objective of this dissertation has been to propose system-theoretic approaches to examine the notion of semi-autonomy in networked systems. Core to this process is the examination of the network's structure and how its properties manifest itself in the system dynamics. The dissertation has endeavored to adapt and extend existing works as well as to make new and innovative forays into protocols, design tools, and modeling methods.

A common thread in this research of networked systems is the appearance of well-known graph measures. For example, in our analysis the variance damping and mean tracking measures can be formulated as the effective resistance of nodes; the highest closeness centrality node is also the most influential node under the mean tracking measure for trees; and structural controllability conditions are equivalent to establishing matchings over bipartite representations of the network. Further, this work refines some of the existing connections to graph measures such as the connection between symmetry and uncontrollability. Specifically, the number of controlled nodes must be at least the determining number of a graph. These connections allow one to draw from the rich area of graph theory for design and reasoning over networked systems.

Consensus is considered one of the fundamental protocols of networked systems. One of the many attractions of the consensus protocol is its fluidic origin and intuition, convergence to agreement, and its clear encoding of the network within the state matrix. These elements were explored in other protocols and dynamic representations. Examples include the fluidic-motivated advection dynamics, which can be considered the conservative dual of consensus over digraphs, and the achievement of agreement over alternate subspaces, using nonlinear forms of consensus. Like the consensus protocol, these protocols also have network-encoded state matrices. In fact, key to the protocol's analysis is that the state matrices, encoded as strongly connected digraphs, are irreducible M-matrices. Interestingly, because of this encoded feature, properties such as stability and controllability can

A. Chapman, *Semi-Autonomous Networks*, Springer Theses,
DOI 10.1007/978-3-319-15010-9_10

be formulated from the network interconnections alone, with little knowledge of the specifics of the protocol. The dissertation explored two such cases: the interval matrix representation of the dynamics, and the pattern matrix representation. Establishing the underlying workings of these generalized protocols provides a myriad of new applications of network dynamic theory as well as addressing the question: *How pronounced is the role of the network*?

The fruitful examination of the network draws from the distributed structure of the protocols and their use of the network itself as the medium of information propagation between agents. Consequently, this dissertation has shown that redesigning the network topology can deter or encourage the flow of certain agents' information. In contrast with centralized network design approaches, this work's protocols have endeavored to maintain the networked system's distributed nature via local agent edge redesign. This work further justifies local edge rewiring to improve various metrics using game-theoretic analysis and by exercising a monotonicity principle. In turn, exploiting the network information flow, local edge reweighting was possible by employing distributed online learning.

Finally, this research demonstrated that network composition and decomposition can maintain the form of the information propagation structure using a Cartesian product formulation and analysis. The result is an efficient way to analyze layered controllability and the ability to perform local layered distributed output feedback stabilization. This perspective emphasizes the valuable connections between the local network information and the corresponding system theoretic performance.

10.2 Future Directions

Much of the research within this dissertation can be considered in its infancy with many potential directions for future work. Each chapter was concluded with a brief mention of future research. The following sections provide a broader perspective of ideas and extensions for the analysis and design of semi-autonomous systems.

10.2.1 Network Design: Striking a Balance

As touched upon in the concluding remarks, one of the powerful aspects of networked systems is the potential to vary the medium of information propagation—the network. This was the basis for the graph redesign problem over the mean tracking measure $J_\mu(\mathcal{G}, \mathcal{R}) \underset{t_f \to \infty}{=} J_\mu(\mathcal{G}, \mathcal{R}, t_f)$, for instance. A natural question is: whether one does better than a static graph for a given set of dynamics? To shed some light on this query consider the mean tracking measure for finite time,

$$\begin{aligned}
J_\mu\left(\mathcal{G},\mathcal{R},t_f\right) &= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\lambda_i\left(-A\right)}\left(1-e^{-2\lambda_i(-A)t_f}\right)\\
&= \frac{1}{n}\left(\sum_{i=1}^{n}\frac{1}{\lambda_i\left(-A\right)}-\sum_{i=1}^{n}\frac{1}{\lambda_i\left(-A\right)}e^{-2\lambda_i(-A)t_f}\right)\\
&= \frac{1}{n}\mathbf{tr}\left(-A^{-1}\right)-\frac{1}{n}\sum_{i=1}^{n}\frac{1}{\lambda_i\left(-A\right)}e^{-2\lambda_i(-A)t_f}\\
&= J_\mu(\mathcal{G},\mathcal{R})-\frac{1}{n}\sum_{i=1}^{n}\frac{1}{\lambda_i\left(-A\right)}e^{-2\lambda_i(-A)t_f}.
\end{aligned}$$

Assuming the number of edges in $\mathcal{G}$ and $\mathcal{R}$ is constant, as for the edge swapping protocols, then $\mathbf{tr}\left(-A\right)$ is constant and so $\lambda_i\left(-A\right)$ is constant. If good mean tracking is required, then one would want to solve the optimization problem

$$\begin{aligned}
\min_{\lambda_1,\lambda_2,\ldots,\lambda_n} \quad & J_\mu\left(\mathcal{G},\mathcal{R},t_f\right)=\textstyle\sum_{i=1}^{n}\frac{1}{\lambda_i}\left(1-e^{-2\lambda_i t_f}\right)\\
s.t. \quad & \textstyle\sum_{i=1}^{n}\lambda_i=c\\
& \lambda_i>0 \qquad\qquad \text{for } i=1,\ldots,n.
\end{aligned}$$

Note that $f_i(\lambda)=\frac{1}{\lambda_i}\left(1-e^{-2\lambda_i t_f}\right)$ is a strictly decreasing convex function over the constraint set, for all $t_f\geq 0$. Consequently, with a little work and noting that the cost function and constraints are symmetric in λ_i's, the optimal solution is $\lambda_1=\lambda_2=\cdots=\lambda_n$. To achieve repeated eigenvalues typically a compact graph with many symmetries is required. Unfortunately, the smallest eigenvalue of $A(\mathcal{G},\mathcal{R})$ is only shifted slightly from the Laplacian zero eigenvalue and much smaller than the other eigenvalues. Consequently, for small t_f, $\sum_{i=1}^{n}\frac{1}{\lambda_i}\left(1-e^{-2\lambda_i t_f}\right)\approx\frac{1}{\lambda_1}\left(1-e^{-2\lambda_1 t_f}\right)$. Hence, graphs with small λ_1 which are typically elongated graphs are favorable for small t_f.

Figure 10.1 provides an example of this observation comparing $J_\mu\left(\mathcal{G},\mathcal{R},t_f\right)$ over different t_f values for a ten-node path and star graph influenced from an end and center node, respectively. Under 3.5 s the path graph is preferred after which the star graph has the smallest $J_\mu\left(\mathcal{G},\mathcal{R},t_f\right)$. A natural solution would be to implement a *time dependent graph*. As t_f increases the optimal edge-count-preserving graph would vary from an elongated graph with slow information propagation to a compact graph with fast information propagation. Another option is to dynamically vary the edge weights of a static edge graph for the same purpose.

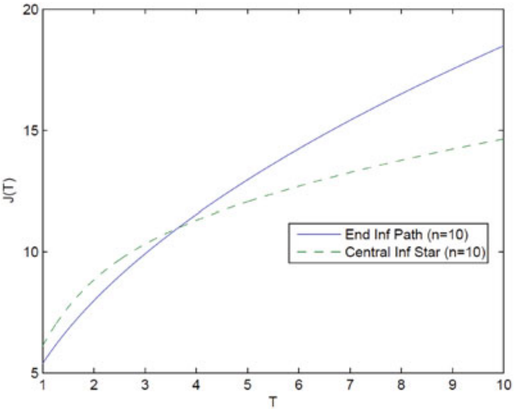


Fig. 10.1 Comparison of $J_\mu\left(\mathcal{G},\mathcal{R},t_f\right)$ for varying t_f for an end node influenced path graph and a centrally influenced star graph on ten nodes

Another challenge in network design is the network's indiscriminate transmission of information. Hence, optimizing for one metric often degrades another. An example is the relationship, derived in Chap. 3, between the mean tracking measure $J_\mu\left(\mathcal{G},\mathcal{R}\right)$ and variance damping measure $J_\sigma\left(\mathcal{G},\mathcal{R}\right)$, namely

$$J_\mu\left(\mathcal{G},\mathcal{R}\right) = J_\sigma\left(\mathcal{G},\mathcal{R}\right) + \frac{1}{n}\sum_{v_i\notin\pi(\mathcal{E}_R)} E_{\text{eff}}\left(v_i\right).$$

The equality bodes poorly for security applications where good variance damping (low $J_\sigma\left(\mathcal{G},\mathcal{R}\right)$) but poor mean tracking (high $J_\mu\left(\mathcal{G},\mathcal{R}\right)$) is required. The online learning method, introduced in Chap. 3, can be used to adapt the network design to address the current state and requirements of the network. One could envisage learning the current noise or mean deviation in the network and weight each metric accordingly then redesigning as required.

An extreme situation of conflicting network design is when both friendly and unfriendly foreign agents attach to the network, and good mean tracking is favorable for some agents and unfavorable for others. A promising direction is a game-theoretic approach, as in Chap. 3, to isolate foreign network agents while grouping native agents about friendly influence point in the network.

10.2.2 Networks-of-Networks

The dissertation's brief excursion into Cartesian products presents only one of the graphical methods for composing and decomposing graphs. Traditionally, a graph product of two graphs $\mathcal{D}_1 = (V_1, E_1)$ and $\mathcal{D}_2 = (V_2, E_2)$ forms a new graph with vertex set $V_1 \times V_2$, where an edge in the product exists between two nodes (v, w) and (v', w') if properties $v = v'$, $(v, v') \in E_1$ or $(v, v') \notin E_1$ are met (and similarly for w and w'). Under these requirements, there are 256 different graph

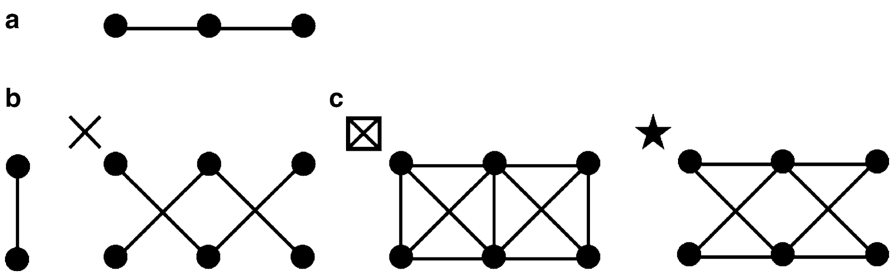


Fig. 10.2 Factor digraphs $\mathcal{D}_1$ and $\mathcal{D}_2$ and composite digraph $\mathcal{D}_1 \circledast \mathcal{D}_2$, where $\circledast \in \{\times, \boxtimes, \star\}$, i.e., the direct, strong and star product, respectively. (**a**) $\mathcal{D}_1$. (**b**) $\mathcal{D}_2$. (**c**) $\mathcal{D} = \mathcal{D}_1 \circledast \mathcal{D}_2$

products formed from combination of these choices. Even more graph products can be formed if more exotic definitions of graphs products are adopted. Examples of some products are displayed in Fig. 10.2. Graph products have been found to exhibit many favorable and predictable composition properties such as expander graphs, extensive symmetry groups, and composable spectrums. Further, many networks naturally take the form of graph products or can be efficiently approximated as such.

The main tool for the Cartesian product controllability analysis was the eigenvector-based PBH test and the fact that eigenvectors can be composed through the Kronecker product. In fact, many graph products can be represented as sums of Kronecker products of the adjacency matrix of $\mathcal{D}_1$ and $\mathcal{D}_2$, implying a similar controllability analysis technique. Some specific examples of graph products and their Kronecker representations are:

- Direct Product: Denoted as $\mathcal{D}_1 \times \mathcal{D}_2$ and defined with edges when $((v, v') \in E_1$ and $(w, w') \in E_2)$. Further,

$$\mathscr{A}(\mathcal{D}_1 \times \mathcal{D}_2) = \mathscr{A}(\mathcal{D}_1) \otimes \mathscr{A}(\mathcal{D}_2).$$

- Star Product: Denotes as $\mathcal{D}_1 \star \mathcal{D}_2$ are defined with edges when $((v, v') \in E_1$ and $(w, w') \in E_2)$ or $((v, v') \in E_1$ and $w = w')$. Further,

$$\mathscr{A}(\mathcal{D}_1 \star \mathcal{D}_2) = I \otimes \mathscr{A}(\mathcal{D}_2) + \mathscr{A}(\mathcal{D}_1) \otimes \mathscr{A}(\mathcal{D}_2).$$

- Strong Product: Denoted as $\mathcal{D}_1 \boxtimes \mathcal{D}_2$ and defined with edges when $((v, v') \in E_1$ and $(w, w') \in E_2)$ or $((v, v') \in E_1$ and $w = w')$ or $(v = v'$ and $(w, w') \in E_2)$. Further,

$$\mathscr{A}(\mathcal{D}_1 \boxtimes \mathcal{D}_2) = \mathscr{A}(\mathcal{D}_1) \otimes I + I \otimes \mathscr{A}(\mathcal{D}_2) + \mathscr{A}(\mathcal{D}_1) \otimes \mathscr{A}(\mathcal{D}_2).$$

- Lexicographic Product: Denotes as $\mathcal{D}_1 \bullet \mathcal{D}_2$ and defined with edges when $((v, v') \in E_1)$ or $(v = v'$ and $(w, w') \in E_2)$. Further,

$$\mathscr{A}(\mathcal{D}_1 \boxtimes \mathcal{D}_2) = \mathscr{A}(\mathcal{D}_1) \otimes \mathbf{1}\mathbf{1}^T + I \otimes \mathscr{A}(\mathcal{D}_2).$$

- Rooted Product with root i: Denotes as $\mathcal{D}_1 \circ \mathcal{D}_2$ and defined with edges when $(w = w_i$ and $(v, v') \in E_1)$ or $(v = v'$ and $(w, w') \in E_2)$. Further,

$$\mathscr{A}(\mathcal{D}_1 \boxtimes \mathcal{D}_2) = \mathscr{A}(\mathcal{D}_1) \otimes e_i e_i^T + I \otimes \mathscr{A}(\mathcal{D}_2).$$

One could also conceive of *dynamic graph products* with different dynamic interactions within layers $\mathcal{G}_1$ as compared to within layers of $\mathcal{G}_2$. In other words, the edge dynamics within a layer are distinct from those edges between layers. For example, consider a Cartesian product graph where agents can measure relative positions across their layer $\mathcal{G}_1$ and relative velocities between layers, i.e., across the layer $\mathcal{G}_2$. A physical interpretation would be that edges within $\mathcal{G}_1$ layers are springs and edges within $\mathcal{G}_2$ layers are dampers. The subsequent dynamics for an agent ij' would be

$$\begin{bmatrix} \dot{x}_{ij'} \\ \ddot{x}_{ij'} \end{bmatrix} = \begin{bmatrix} -\sum_{(i,k)\in E_1} (x_{ij'} - x_{kj'}) \\ -\sum_{(j',k')\in E_2} (\dot{x}_{ij'} - \dot{x}_{ik'}) \end{bmatrix};$$

using Kronecker products this is equivalent over all the states to

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} \mathscr{A}(\mathcal{D}_1) \otimes I & 0 \\ 0 & I \otimes \mathscr{A}(\mathcal{D}_2) \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix}.$$

Similar to the Cartesian product examination these static and dynamics graph product variants can be analyzed for trajectory, spectrum, and controllability composition properties as well as other system theoretic composition properties. The vision is for a library of graph products with associated features to mirror that of the traditional compositional technique such as cascade, parallel, and feedback forms.

10.2.3 Degree of Controllability

A shortcoming of the classification of a system as controllable or not is that there is no indication of the *degree of controllability* or uncontrollability.

10.2.3.1 Controllability Gramian

One measure of the effective controllability is the smallest eigenvalue of the controllability gramian P, which is the solution to the Lyapunov equation

$$A^T P + PA = -BB^T.$$

Consider, for instance, the controllability gramian of the layered control $(A(\mathcal{D}_1 \square \mathcal{D}_2), I \otimes B_2)$ in Sect. 7.5. Assume $A(\mathcal{D}_1)$ and $A(\mathcal{D}_2)$ are stable and symmetric, and V and Λ are the unitary eigenvector and eigenvalue matrices of $A(\mathcal{D}_1)$, respectively. Then,

$$\begin{aligned} A(\mathcal{D}_1 \square \mathcal{D}_2)^T P + PA(\mathcal{D}_1 \square \mathcal{D}_2) &= -(I \otimes B_2)(I \otimes B_2)^T \\ \left(V^T \Lambda V \otimes I + I \otimes A(\mathcal{D}_2)\right) P + P\left(V^T \Lambda V \otimes I + I \otimes A(\mathcal{D}_2)\right) &= -\left(I \otimes B_2 B_2^T\right) \\ (\Lambda \otimes I + I \otimes A(\mathcal{D}_2))\, \bar{P} + \bar{P}\,(\Lambda \otimes I + I \otimes A(\mathcal{D}_2)) &= -\left(I \otimes B_2 B_2^T\right), \end{aligned}$$

where $\bar{P} = (V \otimes I) P (V^T \otimes I)$. Decomposing $\bar{P}$ into a block diagonal matrix with diagonal blocks $\bar{P}_1, \ldots, \bar{P}_{n_1}$ of size $n_2 \times n_2$, the ith diagonal block of the equation is

$$(\lambda_i(A(\mathcal{D}_1))I + A(\mathcal{D}_2))\, \bar{P}_i + \bar{P}_i\, (\lambda_i(A(\mathcal{D}_1))I + A(\mathcal{D}_2)) = -B_2 B_2^T,$$

and so $\bar{P}_i$ is the controllability gramian of $(\lambda_i(A(\mathcal{D}_1))I + A(\mathcal{D}_2), B_2)$. Hence, as the eigenvalues of $\bar{P}$ are the same as P (as $V \otimes I$ is unitary), then the smallest eigenvalue of P is smaller than $(\lambda_n(A(\mathcal{D}_1))I + A(\mathcal{D}_2), B_2)$, i.e., the composite system is less controllable than the factor system.

This type of analysis has potential application to other graph products, providing degree of controllability guarantees for composite graphs.

10.2.3.2 Controllability Index

It is generally difficult to preserve the graph structure in the calculation of the controllability gramian. An alternative measure of the degree of controllability is the controllability index which is the smallest positive integer μ for which the controllability matrix

$$\mathcal{C} = \left[\, B \;\, AB \;\, A^2B \;\, \ldots \;\, A^{\mu-1}B \,\right],$$

is full rank. This is referred to as μ-controllability. A physical interpretation of the controllability index is the smallest number of integrators required to arbitrarily control the dynamics. The attraction of this measure is that the graph structure is maintained in the controllability matrix block columns, and the powers of A can have a graph related interpretation. For example, if A is the adjacency matrix, then powers of $\left[A^k\right]_{ij}$ are the number of paths from node i to node j of length k. A consequence is that μ must be at least the diameter of the graph and more specifically can be no less than the smallest graph distance between native and control nodes in the graph.

10.2.3.3 Extended Controllability Matrix

A variation of the controllability matrix is the extended controllability matrix

$$\bar{\mathcal{C}} = \begin{bmatrix} B & I & 0 & 0 & 0 & 0 & 0 \\ 0 & -A & B & I & 0 & 0 & 0 \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ 0 & 0 & 0 & -A & B & I & 0 \\ 0 & 0 & 0 & 0 & 0 & -A & B \end{bmatrix}$$

of dimension $n^2 \times n\,(n+m-1)$. A feature of this matrix is that the pair (A, B) is controllable if and only if $\bar{\mathcal{C}}$ has rank n^2. An appeal of this formulation is for structural controllability where pattern rank conditions can be checked on the pattern form of $\bar{\mathcal{C}}$. This form would require only a single check for strong structural controllability rather than the two conditions required in Chap. 8.

Strong and weak μ-controllability can be established in a similar way by truncating the first $2(n-\mu)$ blocks column and $n-\mu$ block rows from $\bar{\mathcal{C}}$.

10.2.3.4 Output Controllability

Often, not all the states in the network need to be controlled. A measure of the degree of uncontrollability of the system can be gauged through output controllability, which is a weaker controllability condition and isolates the node states $S \subset V$ which are controllable. A system $(A, B, C(S))$ is output controllable if the output controllability matrix

$$\mathcal{C}_o = C\left[\,B \;\; AB \;\; A^2B \;\ldots\; A^{n-1}B\,\right]$$

is full rank. Equivalently, the system is full rank if the output extended controllability matrix

$$\bar{\mathcal{C}}_o = \begin{bmatrix} B & I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -A & B & I & 0 & 0 & 0 & 0 \\ \vdots & & \ddots & \ddots & \ddots & & \vdots & \vdots \\ 0 & 0 & 0 & -A & B & I & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & -A & B & I \\ 0 & 0 & \cdots & & \cdots & 0 & 0 & -C \end{bmatrix}$$

of dimension $(n^2 + r) \times n\,(n + m - 1)$ is full rank. Similar output controllability index matrices can be constructed using the same matrix truncation techniques as in the previous sections. Matrix $\mathcal{C}_o$ is similarly more appropriate for establishing traditional controllability while $\bar{\mathcal{C}}_o$ could be used to check for output controllability.

The rank of different controllability representations provide many avenues to explore degrees of controllability of a system. These variations promise a more fine-grained form of controllability than the boolean controllability case.

Appendix

Single Anchor State Measures

The single anchor influence scheme adopts a naive approach for influencing the network, justified by the lack of a prior knowledge of the network structure by the attached node. In this case, the attached node merely attempts to steer the system to a common state u_c.[1] This section is devoted to characterizing how effective such an influence scheme for specific graphs and attachment points.

Specifically, we derive fundamental bounds for four metrics of influence. One metric is the controllability gramian $P(\mathcal{G}, \mathcal{R}^i)$, discussed in Sect. 3.4. The other three metrics are derived from the infinite time horizon cost from Sect. 3.3,

$$J(\mathcal{G}, \mathcal{R}^i, \tilde{x}(0)) = -\tilde{x}(0)^T A(\mathcal{G}, \mathcal{R}^i)^{-1} \tilde{x}(0).$$

The first of these metrics is mean tracking measure or average performance cost, and can be derived as

$$\mathbb{E}_{\|\tilde{x}(0)\|=1} J(\mathcal{G}, \mathcal{R}^i, \tilde{x}(0)) = J_\mu(\mathcal{G}, \mathcal{R}^i).$$

The other metrics are referred to as the minimum and maximum performance costs in terms of the eigenvalues of the Dirichlet matrix as[2]

$$\begin{aligned} J^{\min}\left(\mathcal{G}, \mathcal{R}^i\right) &= \inf_{\|\tilde{x}(0)\|=1} J\left(\mathcal{G}, \mathcal{R}^i, \tilde{x}(0)\right) \\ &= \lambda_1\left(-A\left(\mathcal{G}, \mathcal{R}^i\right)^{-1}\right) \\ &= 1/\lambda_n\left(-A\left(\mathcal{G}, \mathcal{R}^i\right)\right), \end{aligned}$$

[1] For example, in order to realign a formation or change its speed.

[2] Considered as intrusion or management costs.

A. Chapman, *Semi-Autonomous Networks*, Springer Theses,
DOI 10.1007/978-3-319-15010-9

and

$$\begin{aligned} J^{\max}\left(\mathcal{G},\mathcal{R}^i\right) &= \sup_{\|\tilde{x}(0)\|=1} J\left(\mathcal{G},\mathcal{R}^i,\tilde{x}(0)\right) \\ &= \lambda_n\left(-A\left(\mathcal{G},\mathcal{R}^i\right)^{-1}\right) \\ &= 1/\lambda_1\left(-A\left(\mathcal{G},\mathcal{R}^i\right)\right). \end{aligned}$$

The critical graphs that bound the performance costs $J^{\min}$, $J^{\max}$, and J_μ for a single input anchor influence scheme over all graphs are the n-node complete graph $\mathcal{K}$, the path graph $\mathcal{P}$, and the star graph $\mathcal{S}$. The derivation of the four metrics and bounds relating to these graphs follow.

Proposition A.1. *For the n-node path graph $\mathcal{P}$, with an influencing node attached to an end node $v_1 \in \mathcal{P}$, the minimum, maximum, and average performance costs are, respectively,*

$$J^{\min}\left(\mathcal{P},\mathcal{R}^1\right) = \frac{1}{2}\left(1+\cos\frac{2\pi}{2n+1}\right)^{-1},$$

$$J^{\max}\left(\mathcal{P},\mathcal{R}^1\right) = \frac{1}{2}\left(1+\cos\frac{2\pi n}{2n+1}\right)^{-1},$$

and

$$J_\mu(\mathcal{P},\mathcal{R}^1) = \frac{1}{2}(n+1).$$

Proof. The Dirichlet matrix for an influencing agent attached to v_1 an end node of $\mathcal{P}$ is

$$-A\left(\mathcal{P},\mathcal{R}^1\right) = \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 2 & -1 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix},$$

with

$$\lambda_i\left(-A\left(\mathcal{P},\mathcal{R}^1\right)\right) = 2\left(1+\cos\frac{2\pi i}{2n+1}\right),$$

for $i = 1, \ldots, n$, with the corresponding eigenvector with its jth entry $\sin(\frac{2\pi ij}{2n+1})$; see [2]. By symmetry, the scenario where the other end of the path is influenced is identical. □

Proposition A.2. *For the n-node path graph $\mathcal{P}$, with influence node attached to an end node $v_1 \in \mathcal{P}$, the controllability gramian for (3.2), defined element-wise for each entry (p, q), is*

$$\left[P\left(\mathcal{P}, \mathcal{R}^1\right)\right]_{pq} = \frac{8}{(1+2n)^2} \sum_{w=1}^{n} \sum_{z=1}^{n} \frac{\sin pw\beta \, \sin qz\beta \, \sin w\beta \, \sin z\beta}{2 + \cos w\beta + \cos z\beta},$$

where $\beta = 2\pi/(2n+1)$.

Proof. The proof follows from verifying that the matrix satisfies the Lyapunov equation. □

Proposition A.3. *For the n-node star graph $\mathcal{S}$, with the influence node attached to the central node $v_1 \in \mathcal{S}$, the minimum, maximum, and average performance costs are, respectively,*

$$J^{\min}\left(\mathcal{S}, \mathcal{R}^1\right) = 2\left(n + 1 + \sqrt{n^2 + 2n - 3}\right)^{-1},$$

$$J^{\max}\left(\mathcal{S}, \mathcal{R}^1\right) = 2\left(n + 1 - \sqrt{n^2 + 2n - 3}\right)^{-1},$$

and

$$J_\mu\left(\mathcal{S}, \mathcal{R}^1\right) = 2 - \frac{1}{n}.$$

Proof. The Dirichlet matrix for an influencing agent attached to the central node of $\mathcal{S}$, v_1, is

$$-A\left(\mathcal{S}, \mathcal{R}^1\right) = \begin{bmatrix} n & -\mathbf{1}^T \\ -\mathbf{1} & I \end{bmatrix}.$$

Examining the identity $-A\left(\mathcal{S}, \mathcal{R}^1\right) v = \lambda v$, we note that there are $n-2$ eigenvectors of the form $v = [0 \; \alpha]^T$, where $\alpha \in \mathbb{R}^{n-1}$ and $\sum \alpha_i = 0$ corresponding to $\lambda = 1$, and two eigenvectors of the form $v = [1 - \lambda \quad \mathbf{1}^T]^T$, where

$$\lambda = \frac{1}{2}\left(1 + n \pm \sqrt{n^2 + 2n - 3}\right).$$

□

Proposition A.4. *For the n-node star graph $\mathcal{S}$, with influencing node attached to the central node $v_1 \in \mathcal{S}$, the controllability gramian of (3.2) is*

$$P\left(\mathcal{S},\mathcal{R}^1\right) = \frac{1}{2\left(n+1\right)}\left(\mathbf{1}\mathbf{1}^T + e_1 e_1^T\right),$$

with $\lambda_j\left(P\left(\mathcal{S},\mathcal{R}^1\right)\right) = 0$, *for* $j = 1,\ldots,n-2$ *and*

$$\lambda_j\left(P\left(\mathcal{S},\mathcal{R}^1\right)\right) = \frac{1}{4}\left(1 \pm \frac{\sqrt{n^2-2n+5}}{n+1}\right),$$

for $j = n-1, n$, *respectively.*

Proof. The proof follows from verifying that the matrix satisfies the Lyapunov equation with $B = e_1$. The gramian and its corresponding eigenvalues are identical as for the complete graph with an attached node. □

Proposition A.5. *For the n-node complete graph $\mathcal{K}$, with an influencing node attached to any node $v_i \in \mathcal{K}$, the minimum, maximum, and average performance costs are, respectively,*

$$J^{\min}\left(\mathcal{K},\mathcal{R}^1\right) = 2\left(n+1+\sqrt{n^2+2n-3}\right)^{-1},$$

$$J^{\max}\left(\mathcal{K},\mathcal{R}^1\right) = 2\left(n+1-\sqrt{n^2+2n-3}\right)^{-1},$$

and

$$J_\mu\left(\mathcal{K},\mathcal{R}^1\right) = \left(1+\frac{2}{n}-\frac{2}{n^2}\right).$$

Proof. The Dirichlet matrix for an influencing agent attached to v_1, an arbitrary node of $\mathcal{K}$, is

$$-A\left(\mathcal{K},\mathcal{R}^1\right) = \begin{bmatrix} n & -\mathbf{1}^T \\ -\mathbf{1} & -\mathbf{1}\mathbf{1}^T + nI \end{bmatrix}.$$

Examining the identity $-A\left(\mathcal{K},\mathcal{R}^1\right)v = \lambda v$, there are $n-2$ eigenvectors of the form $v = [0\ \alpha]^T$, where $\alpha \in \mathbb{R}^{n-1}$ and $\sum \alpha_i = 0$ corresponding to $\lambda = n$ and two eigenvalues of the form $v = [1-\lambda \quad \mathbf{1}^T]^T$, where

$$\lambda = \frac{1}{2}\left(1+n \pm \sqrt{n^2+2n-3}\right).$$

Due to the symmetry of $\mathcal{K}$, the cost incurred by the attached node to steer the network is independent of where it attaches. □

We now proceed to apply these critical graph bounds to arbitrary graphs.

Proposition A.6. *For an n-node connected graph $\mathcal{G}$, the minimum and maximum performance costs of attaching to a node $v_i \in \mathcal{G}$ is bounded as*

$$\inf_{(\mathcal{G},i)} J^{\min}\left(\mathcal{G},\mathcal{R}^i\right) = 2\left(1+n+\sqrt{n^2+2n-3}\right)^{-1},$$

and

$$\sup_{(\mathcal{G},i)} J^{\max}\left(\mathcal{G},\mathcal{R}^i\right) = \frac{1}{2}\left(1+\cos\frac{2\pi n}{2n+1}\right)^{-1}.$$

Proof. Let $\mathcal{G}$ be an arbitrary n-node graph with its complement graph $\hat{\mathcal{G}}$, noting that $L(\mathcal{G})+L(\hat{\mathcal{G}})=L(\mathcal{K})$, where $\mathcal{K}$ is the n-node complete graph. Since

$$L(\mathcal{G})+L(\hat{\mathcal{G}})+e_ie_i^T = L(\mathcal{K})+e_ie_i^T,$$

it follows that

$$\lambda_n(L(\mathcal{G})+e_ie_i^T) \le \lambda_n(L(\mathcal{K})+e_ie_i^T).$$

Therefore,

$$\begin{aligned} J^{\min}\left(\mathcal{K},\mathcal{R}^i\right) &= (\lambda_n(L(\mathcal{K})+e_ie_i^T))^{-1} \\ &\le (\lambda_n(L(\mathcal{G})+e_ie_i^T))^{-1} \\ &= J^{\min}\left(\mathcal{G},\mathcal{R}^i\right). \end{aligned}$$

Now consider a spanning tree $\mathcal{T}$ of the graph $\mathcal{G}$. Attaching an influencing node to $\mathcal{T}$ and examining the corresponding smallest Dirichlet eigenvalue, we obtain

$$\lambda_1(L(\mathcal{T})+e_ie_i^T) \le \lambda_1(L(\mathcal{G})+e_ie_i^T).$$

Next, construct the new tree $\tilde{\mathcal{T}}$ by mirroring $\mathcal{T}$ about the influencing node and treating it as a *native* node in this new graph which has $2n+1$ nodes. From Lemma 6 of [3], it follows that

$$\lambda_2(\tilde{\mathcal{T}}) \le \lambda_1(L(\mathcal{T})+e_ie_i^T).$$

In the meantime, as the path graph is the tree with the least second smallest eigenvalue over all n-node connected graphs [1], it follows that $\lambda_2(\tilde{\mathcal{P}}) \leq \lambda_2(\tilde{\mathcal{T}})$ where $\tilde{\mathcal{P}}$ is a path of order $2n+1$. From Proposition A.1, for a path graph $\mathcal{P}$ of order n, with the influence node attached to $v_1 \in \mathcal{P}$, it is known that $\lambda_2(\tilde{\mathcal{P}}) = \lambda_1(-A\left(\mathcal{P}, \mathcal{R}^1\right))$; see also [1]. Combining these bounds, we arrive at the inequality

$$\lambda_2(\tilde{\mathcal{P}}) \leq \lambda_1(L(\mathcal{T}) + e_i e_i^T),$$

and thereby,

$$\begin{aligned} J^{\min}\left(\mathcal{G}, \mathcal{R}^i\right) &= \frac{1}{\lambda_1(L(\mathcal{G}) + e_i e_i^T)} \\ &\leq \frac{1}{\lambda_1\left(-A\left(\mathcal{P}, \mathcal{R}^1\right)\right)} \\ &= J^{\min}\left(\mathcal{P}, \mathcal{R}^1\right). \end{aligned}$$

Closed form solutions for $J^{\min}\left(\mathcal{K}, \mathcal{R}^i\right)$ and $J^{\min}\left(\mathcal{P}, \mathcal{R}^i\right)$ are found in Propositions A.1 and A.5, thus completing the proof. □

Proposition A.7. *For an n-node m-edge graph $\mathcal{G}$ with largest node degree $d_{\max}(\mathcal{G})$ and smallest node degree $d_{\min}(\mathcal{G})$, the minimum and maximum performance costs of attaching to any node $v_i \in \mathcal{G}$ are bounded as*

$$\frac{1}{2d_{\max}(\mathcal{G}) + 1} \leq J^{\min}\left(\mathcal{G}, \mathcal{R}^i\right),$$

and

$$J^{\max}\left(\mathcal{G}, \mathcal{R}^i\right) \leq \frac{1}{2}\left(n^2 + 3n - 2m - 2\right).$$

Proof. Let the vertices in $\mathcal{G}$ be labeled such that the diagonal entries of the matrix $2\Delta(\mathcal{G}) + e_i e_i^T$ are ordered in a non-decreasing order; thus, $\lambda_j\left(\left(-A\left(\mathcal{G}, \mathcal{R}^i\right)\right)\right) \leq 2d_j(\mathcal{G})$ for all j. However, since

$$\lambda_n\left(-A\left(\mathcal{G}, \mathcal{R}^i\right)\right) \leq \lambda_n\left(-A\left(\mathcal{G}, \mathcal{R}^n\right)\right) \leq 2d_{\max}(\mathcal{G}) + 1,$$

it follows that

$$\left(2d_{\max}(\mathcal{G}) + 1\right)^{-1} \leq \lambda_n\left(-A\left(\mathcal{G}, \mathcal{R}^i\right)\right)^{-1} = J^{\min}\left(\mathcal{G}, \mathcal{R}^i\right).$$

Consider next a spanning tree $\mathcal{T}$ of $\mathcal{G}$. Then $0 \preceq \mathscr{L}(\mathcal{G}) - \mathscr{L}(\mathcal{T}) = \mathscr{L}(\mathcal{G}) + e_i e_i^T - (\mathscr{L}(\mathcal{T}) + e_i e_i^T) = -A(\mathcal{G}, \mathcal{R}^i) - (-A(\mathcal{T}, \mathcal{R}^i))$ and thereby $\lambda_j\left(-A\left(\mathcal{T}, \mathcal{R}^i\right)\right) \leq \lambda_j\left(-A\left(\mathcal{G}, \mathcal{R}^i\right)\right)$, for $j = 1, \ldots, n$. Consequently the mean tracking measure $J_\mu\left(\mathcal{G}, \mathcal{R}^i\right)$ has the property that

$$\begin{aligned} nJ_\mu(\mathcal{G}, \mathcal{R}^i) &\leq \sum_{j=1}^{n} \lambda_j\left(-A\left(\mathcal{T}, \mathcal{R}^i\right)\right)^{-1} \\ &\leq \sup_{\mathcal{T} \subseteq \mathcal{G}} nJ_\mu\left(\mathcal{T}, \mathcal{R}^i\right) \\ &= \sup_{\mathcal{T} \subseteq \mathcal{G}} \sum_{j=1}^{n} d\left(v_i, v_j\right) + n \\ &= \frac{1}{2}\left(n^2 + 3n - 2m - 2\right), \end{aligned}$$

where the last inequality is from examining spanning trees of all n-nodes m-edges graphs, which in turn provides the final equality $\sum_{j=1}^{n} d\left(v_i, v_j\right) \leq \frac{1}{2}(n-1)(n+2) - m$. Applying this bound we also have

$$\begin{aligned} J^{\max}\left(\mathcal{G}, \mathcal{R}^i\right) &= \lambda_1\left(-A\left(\mathcal{G}, \mathcal{R}^i\right)\right)^{-1} \\ &\leq \sum_{j=1}^{n} \lambda_j\left(-A\left(\mathcal{G}, \mathcal{R}^i\right)\right)^{-1} \\ &= nJ_\mu(\mathcal{G}, \mathcal{R}^i) \\ &\leq \frac{1}{2}\left(n^2 + 3n - 2m - 2\right), \end{aligned}$$

thus completing the proof. □

References

1. M. Petrovic and I. Gutman, “The path is the tree with smallest greatest Laplacian eigenvalue,” *Kragujevac Journal of Mathematics*, vol. 24, pp. 67–70, 2002.
2. W.-C. Yueh, “Eigenvalues of several tridiagonal matrices,” *Applied Mathematics E-Notes*, vol. 5, pp. 66–74, 2005.
3. T. Biyikoglu and J. Leydold, “Algebraic connectivity and degree sequences of trees,” *Linear Algebra and its Applications*, vol. 430, no. 2–3, pp. 811–817, 2009.

About the Author

Airlie Chapman received the Ph.D. degree from the William E. Boeing Aeronautics and Astronautics Department at the University of Washington, Seattle, in 2013 and was simultaneously awarded the M.S. degree in mathematics. She received the B.S. degree in aeronautical (space) engineering and the M.S. degree in engineering research from the University of Sydney, Australia, in 2006 and 2008, respectively. She is currently a postdoctoral fellow at the University of Washington, Seattle.

Dr. Chapman was awarded the College of Engineering Dean's Fellowship at the University of Washington and is a two-time recipient of the Amelia Earhart Fellowship. Her research interests are networked dynamic systems and graph theory with applications to robotics and aerospace systems.

A. Chapman, *Semi-Autonomous Networks*, Springer Theses,
DOI 10.1007/978-3-319-15010-9

Parts of this thesis that have been published in the following conference, journal, and book articles

- Airlie Chapman and Mehran Mesbahi (2015) UAV Swarms: Models and Effective Interfaces. In *Handbook of Unmanned Aerial Vehicles*, Springer.
- Airlie Chapman, Marzieh Nabi-Abdolyousefi and Mehran Mesbahi (2014) Controllability and observability of networks-of-networks via Cartesian products, 2668–2679. In *IEEE Transactions on Automatic Control* 59 (10).
- Airlie Chapman, Eric Schoof and Mehran Mesbahi (2013) Distributed Online Topology Design for Disturbance Rejection, 817–821. In *Proc. 52nd IEEE Conference on Decision and Control.*
- Airlie Chapman and Mehran Mesbahi (2013) Security and Infiltration of Networks: A Structural Controllability and Observability Perspective, 143–160. In *Control of Cyber-Physical Systems: Lecture Notes on Control and Information Sciences* 449, Springer
- Airlie Chapman and Mehran Mesbahi (2013) Semi-Autonomous Consensus: Network Measures and Adaptive Trees, 19–31. In *IEEE Transactions on Automatic Control* 58 (1).
- Airlie Chapman and Mehran Mesbahi (2013) Strong Structural Controllability of Networked Dynamics, 6141–6146. In *Proc. American Control Conference.*
- Airlie Chapman, Eric Schoof and Mehran Mesbahi (2012) Advection on Networks with an Application to Decentralized Load Balancing, 2680–2681. In *Proc. IEEE Conference on Intelligent Robots and Systems.*
- Airlie Chapman and Mehran Mesbahi (2012) Cartesian products of Z-Matrix networks: Factorization and interval analysis. In *Proc. 20th International Symposium on Mathematical Theory of Networks and Systems.*
- Airlie Chapman, Marzieh Nabi-Abdolyousefi and Mehran Mesbahi (2012) Controllability and Observability of Cartesian Product Networks, 80–85. In *Proc. 51st IEEE Conference on Decision and Control.*
- Airlie Chapman and Mehran Mesbahi (2012) Stability Analysis of Nonlinear Networks via M-matrix Theory: Beyond Linear Consensus, 6626–6631. In *Proc. American Control Conference.*
- Airlie Chapman and Mehran Mesbahi (2012) System Theoretic Aspects of Influenced Consensus: Single Input Case, 1505–1511. In *IEEE Transactions on Automatic Control* 57 (6).
- Airlie Chapman and Mehran Mesbahi (2011) Advection on Graphs, 1461–1466. In *Proc. 50th IEEE Conference on Decision and Control.*
- Airlie Chapman, Ran Dai and Mehran Mesbahi (2011) Network Topology Design for UAV Flocking with Wind Gusts. In *Proc. AIAA Guidance, Navigation and Control Conference.*
- Airlie Chapman and Mehran Mesbahi (2011) UAV Flocking with Wind Gusts: Adaptive Topology and Model Reduction, 1045–1050. In *Proc. American Control Conference.*

- Airlie Chapman, Eric Schoof and Mehran Mesbahi (2010) Semi-Autonomous Networks: Theory and Decentralized Protocols, 1958–1963. In *Proc. IEEE International Conference on Robotics and Automation*.
- Airlie Chapman and Mehran Mesbahi (2010) Semi-Autonomous Networks: Network Resilience and Adaptive Trees, 7473–7478. In *Proc. 49th IEEE Conference on Decision and Control*.
- Airlie Chapman, Marzieh Nabi-Abdolyousefi and Mehran Mesbahi (2009) Identification and Infiltration in Consensus-type Networks, 84–89. In *1st IFAC Workshop on Estimation and Control of Networked Systems*.

Index

A. Chapman, *Semi-Autonomous Networks*, Springer Theses,
DOI 10.1007/978-3-319-15010-9